Decomposable Intelligence on Cloud-Edge IoT Framework for Live Video Analytics

Yi Zhang, *Student Member, IEEE,* Jiun-Hao Liu, Chih-Yu Wang, *Member, IEEE,* and Hung-Yu Wei, *Senior Member, IEEE*

***Abstract*—With the rapid development of deep learning technology, the modern internet-of-things (IoT) cameras have very high demands on communication, computing, and memory resources so as to achieve low latency and high accuracy live video analytics. Thanks to mobile edge computing (MEC), intelligent offloading to the MEC nodes can bring a lot of benefits, especially when the decomposable pipeline is adopted in the cloud-edge architecture. In this paper, we provide Decomposable Intelligence on a Cloud-Edge IoT (DICE-IoT) framework to support joint latency- and accuracy- aware live video analytic services. Specifically, the intelligent framework enables the pipeline sharing mechanism to reduce MEC resource usage. A Nash bargaining is proposed to incentivize cooperative computing provision between the MEC and the cloud, and a Generalized Benders Decomposition (GBD) based approach is utilized to optimize the social welfare. The results show that the proposed DICE-IoT framework can achieve a win-win-win solution to the IoT device, the MEC, and the cloud stratums.**

***Index Terms*—Live video analytics, mobile edge computing, joint resource allocation, Nash bargaining**

I. Introduction

Real-time video analytics in the city-wide area is a critical function in public safety applications, such as violence detection, traffic monitoring, self-driving, VR/AR, etc [1]. Such a function is expected to be realized by the modern Internet of Things (IoT) based surveillance system. In recent years, by leveraging powerful deep learning (DL) technology, the detection accuracy of computer vision and video analytics has been dramatically improved, which makes live video analytics becomes possible at the software level. Nevertheless, it is still challenging to implement a DL-based real-time video analytic service due to computing resources, bandwidth, and latency constraints. In conventional IoT systems, the live video sequences are captured from surveillance cameras and delivered to the remote cloud for video analytics, which has very large bandwidth demands [2]. In addition, performing the DL model requires large GPU and memory resources, which are not available at the IoT devices and thus need the assistance of the cloud. However, high latency is inevitable due to the network congestion and long-distance transmissions from the end device to the cloud. Traditional cloud-based solutions, even with the support of 5G, is not sufficient for the live video analytic services.

Different from some improved solutions to the cloud computing, i.e., heterogeneous cloud [3] or cloudlet [4], mobile edge computing (MEC) distributes a substantial number of capabilities closer to the IoT devices, i.e., communication, computing, storage, and control [5, 6]. For live video analytics, intelligent offloading to the MEC nodes [7] can alleviate network congestion, reduce latency, and lower operating costs. However, a complete DL model may be impossible to be loaded in a MEC node equipped with limited GPU memory. Fortunately, the decomposable pipeline brings new opportunities by leveraging cloud-edge architecture. The *layer-level pipeline composition* partitions the entire DL model into multiple DL sub-models so that they could be deployed in MEC nodes and remote cloud separately [8]. As a result, an optimal partition of a particular DL model could be observed offline, which yields less total inference delays comparing with MEC-only and cloud-only schemes. Besides, an inference pipeline for video analytics usually consists of multiple detection models and therefore a well-designed *model-level pipeline composition* may bring benefits [9]. For example, the best composition of a video monitoring pipeline in some cases could be to apply YOLOv3 [10] at the MEC node for object detection and further employ FaceNet [11] at the remote cloud for face recognition. In addition, recent studies expose that there exists a resource-quality tradeoff when selecting a combination of knobs for video analytics, which contains video resolution, frame sampling rate, and specific analytic model [12, 13]. That is to say, a suitable configuration of various knobs can relief resource consumption subject to the desired accuracy requirement.

A rich set of resource allocation solutions [8, 12, 14–16] have been proposed to deploy live video analytics in MEC environments. However, most of the current solutions may not be practical enough to the real-world service deployments due to several aspects: *1)* Both communication and computing resource allocations are urgent issues to meet the high quality of service (QoS) demands. For simplicity, some work [8] assumes that the link bandwidth has been reserved, which greatly reduces the value and difficulty of the problem. *2)* Both computing and memory resources limit the capacity of analytic services, especially when advanced DL technology is applied. But most of the works ignore the importance of memories and only target at computing resources in terms of CPU core [12], processor speed [14, 15], even a more simplified

Corresponding author: Hung-Yu Wei.

Yi Zhang is with the Graduate Institute of Communication Engineering, National Taiwan University, Taiwan (email: yzhang.cn@outlook.com)

Jiun-Hao Liu is with the Department of Electrical Engineering, National Taiwan University, Taiwan (email: b02901112@ntu.edu.tw)

Chih-Yu Wang is with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan (email: cywang@citi.sinica.edu.tw)

Hung-Yu Wei is with the Graduate Institute of Communication Engineering and Department of Electrical Engineering, National Taiwan University, Taiwan (email: hywei@cc.ee.ntu.edu.tw)

formulation as resource capacity [16]. *3)* Low latency and high accuracy video analytics, which are pressing demands for a modern IoT-based surveillance system, have never been jointly considered. **Therefore, it is necessary to address a joint resource allocation problem with multi-type resources subject to both latency and accuracy requirements for live video analytics.**

In this paper, we propose **D**ecomposable **I**ntelligence on a **C**loud-**E**dge **IoT** (DICE-IoT) framework to provide joint latency- and accuracy- aware live video analytics to the IoT-based surveillance cameras called webcams. In the proposed intelligent framework, the MEC stratum consisting of several MEC nodes is controlled by a MEC orchestrator, while the cloud stratum with an unlimited resource budget is managed by a cloud controller. We consider multi-type resources when performing a specific decomposable pipeline, including link bandwidth, GPU utilization, and memory. Due to the limited computing capacities and memory resources of MEC nodes, we share a similar idea of [16] that allows pipeline sharing for multiple webcams. As the MEC orchestrator, the objective is to maximize the expected utility of the MEC stratum with optimal bandwidth allocation and computing provision, where the latter consists of user association and configuration selection. It is worth noting that the proposed decomposable intelligence is not limited to live video analytic applications. By modifying a part of variables or constraints, e.g., without considering video resolution, it still can be applied to general cloud-edge IoT framework with joint latency and accuracy concerns. The original contributions of this paper are summarized as follows:

- Recall that it is necessary to address a joint resource allocation problem with multi-type resources subject to both latency and accuracy requirements for live video analytics. In this paper, we present a practical cloud-edge IoT framework to intelligently provide decomposable live video analytics with joint latency and accuracy awareness. Specifically, multi-type resources (i.e., link bandwidth, GPU utilization, and memory) are considered in the joint resource allocation of network and computing demands. A pipeline sharing mechanism is enabled for GPU memory usage reduction.
- It is expected that MEC resources will be deployed by the network operator, which will be different from the cloud operator. Conflicts of interests may arise in the cloud-edge computing provision. We propose a Nash bargaining with price negotiation between the MEC orchestrator and the cloud controller to incentivize the cloud stratum for cooperative cloud-edge computing provision. Unlike the most of the related works, which applies greedy or heuristic approaches to deal with mixed-integer nonlinear programming problem (MINLP), we utilize the reformulation linearization technique (RLT) to resolve the difficulty of the joint resource allocation problem and then provide Generalized Benders Decomposition (GBD) based approach for optimal social welfare calculation.
- We take a decomposable inference pipeline consisting of object detection and face recognition as an application example to evaluate the proposed DICE-IoT framework. We not only set up a practical experimental testbed for parameter measurement but also conduct extensive numerical simulations. The results show that the proposed intelligent framework achieves a win-win-win solution to the IoT device, the MEC, and the cloud stratums thanks to the proposed pipeline sharing mechanism and the cooperative computing provision. The proposed GBD-based approach can greatly reduce the execution time compared with the greedy approach.

The rest of the paper is organized as follows. We summarize related works in Section II. The proposed DICE-IoT framework is presented in Section III, and then a joint resource allocation problem is raised in Section IV to provide video analytic services subject to low latency and high accuracy requirements. Furthermore, we provide a Nash bargaining in Section V to incentivize cooperative cloud-edge framework and then propose a GBD-based social welfare calculation in Section VI. Section VII shows our evaluation results and, finally, Section VIII concludes this work.

II. Related work

When integrating the advanced MEC technologies into live video analytics, decomposable pipeline [1, 8, 9, 17] has great potential to exploit the advantages of the edge computing environments. The distributed intelligent video surveillance (DIVS) system [1] shares a similar idea of [8] that enables parallel training, model synchronization, and workload balancing through distributed DL sub-models deployed on the MEC nodes. Four representative prediction pipelines in the model-level composition are provided in [9], including image processing, video monitoring, social media prediction, and TensorFlow cascade. Besides, a proactive pipeline optimizer is deployed to satisfy the end-to-end latency and a reactive controller is provided to monitor the per-model configuration at runtime. Besides, frequent reconfiguration [17] of model-level pipelines composition is required because of dynamic topology changes caused by the user's mobility, i.e., users move between edges, as well as the switching of active applications.

QoS-based resource allocation is the main challenge when deploying a live video analytics system. So far, a lot of resource allocation solutions [8, 12, 14–16, 18] have been proposed to satisfy the pressing demands of live video analytics. In order to maximize the overall QoS of multi-access point (AP) wireless camera networks, a joint deployment and association [18] are provided to determine suitable locations of APs and then the camera-AP association as well as the video stream transmission rate. For the application of Automated License Plate Recognition (ALPR), the offloading task selection and bandwidth allocation [14] are proposed for each single edge node to minimize the latency of video analytics. In a single camera system, a cooperative video processing scheme [15] is proposed to offload video chunks to multiple edge-capable groups nearby. In order to minimize the average video coding rate, group formulation and video–group matching are addressed sequentially using greedy and low-complex heuristic algorithms, respectively.

To overcome the insufficient resources equipped at mobile devices and push the workload of mobile deep learning ap-

TABLE I
A COMPARISION OF RELATED WORK IN THE LITERATURE

Reference	Architecture	Resource allocation	Required metrics	Pipeline sharing	Proposed methods
[8]	Edge + Cloud	**C**	**D**	×	▷ Event-triggered online schedule algorithm
[12]	Edge	**C**	**D**, **Q**	×	▷ Greedy approximation with high-value query
[14]	Edge + Cloud	**N**, **C**	**D**	×	▷ Continuous relaxation & exhaustive search (**optimal**)
[15]	Edge	**C**	**D**, **Q**	×	▷ Greedy group formation & Video-group matching
[16]	Edge + Cloud	**N**, **C**	**Q**	○	▷ Multiple-choice multi-dimensional knapsack problem (MMK) & greedy-based heuristic approach
[18]	×	**N**	**D**	×	▷ *1)* Branch-and-bound (**optimal**) *2)* Iterative heuristics
Our work	Edge + Cloud	**N**, **C**, **M**	**D**, **Q**	○	▷ Nash Bargaining & GBD-based approach (**optimal**)

Hints: with = ○ w/o = × **N** = Network **C** = Computing **M** = Memory
D = Delay **Q** = Accuracy **optimal** = Optimal solution

plications to the near-end edge instead of the remote cloud, a layer-level pipeline composition [8] is presented to partition AlexNet so as to yield shorter total delay for edge inference tasks. However, it assumes that the link bandwidth has been reserved and each mobile task has its private pipeline. By contrast, our proposed DICE-IoT system not only considers the bandwidth allocation problem but also allows pipeline sharing for multiple IoT cameras if possible. In VideoStorm architecture [12], a greedy but efficient profiler is employed to pick a handful of knob configurations. To improve near-future performance, query lag is predicted at the scheduler and two objectives, maximizing the sum of utilities and maximizing minimum utility, are formulated, where the utility is defined as the weighted sum of quality improvement and lag reduction. Note that the worker in this system acts as a MEC node and the remote cloud is not considered. Furthermore, they only consider single resource type, i.e., CPU resource, while our proposed DICE-IoT system aims at a more complicated joint resource allocation problem.

The VideoEdge framework [16], which is state-of-the-art work, provides a joint resource allocation of network and computing demands for video analytics using a heuristic approach. However, the latency of the analytic task is not considered; the computing resource is simply formulated as resource capacity; the network links between cameras and the local private clusters are not discussed. More specifically, a query planning is applied to select the best knob configuration for a query and a component placement is used to determine the locations of pipeline components according to the available resource capacities. Furthermore, component merging is considered to eliminate redundant components due to insufficient network or computing resources. Unlike our proposed pipeline sharing mechanism, only those common components from multiple queries of the same camera are allowed to merge. Besides, the VideoEdge framework controls only the frame resolution while our proposed framework supports a list of decomposable pipelines by considering different video resolution, frame sampling rate, and analytic models. Compared with the heuristic approach of the VideoEdge framework, our proposed framework provides an optimal solution for bandwidth allocation and computing provision. Finally, a comparison of related work in the literature is listed in Table I. It shows that most of the current solutions have more or less weaknesses in joint resource allocation, required metrics, and the proposed methods while our work provides an optimal solution in a cloud-edge architecture by compensating those weaknesses.

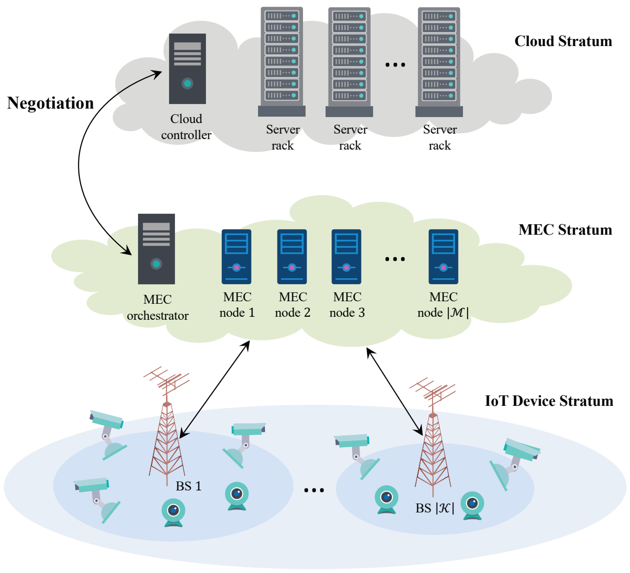


Fig. 1. The proposed DICE-IoT architecture for live video analytics.

III. SYSTEM MODEL

The hierarchical DICE-IoT architecture is presented in Fig. 1. We consider a *MEC stratum*, which consists a number of MEC nodes $\mathcal{M} = \{1, \cdots, |\mathcal{M}|\}$ controlled by a *MEC orchestrator*. Each MEC node $j \in \mathcal{M}$ is equipped with limited memory and computing resources. A *cloud stratum* at the far end owns a large number of server racks, which are independently managed by a *cloud controller*. Generally, we assume that the cloud stratum has an unlimited resource budget. Under the proposed architecture, there are IoT-based surveillance cameras $\mathcal{N} = \{1, \cdots, |\mathcal{N}|\}$, called *webcams*, fixed in the *IoT device stratum* that captures live video sequences and then upload to the MEC stratum for real-time video analytics, e.g., object detection and face recognition.

In the MEC stratum, those analytic results could be furthered used for control and management, e.g., security alarm. Specifically, each webcam $i \in \mathcal{N}$ has its own latency requirement L_i^{req} and quality requirement Q_i^{req} in terms of detection accuracy. Unlike conventional cloud computing architecture, the proposed system leverages the power of the MEC stratum for latency reduction, i.e., communication delay and computing delay, so as to satisfy the latency requirement and meanwhile achieve high detection accuracy. However, due to the limited computing capacities and memory resources of MEC nodes, it is further encouraged to **negotiate** with the cloud controller for cooperative computing provisioning deployment.

A. Network and Communication Model

We assume that the webcams are within the range of some base stations (BSs) $\mathcal{K} = \{1, \cdots, |\mathcal{K}|\}$ so that they can upload the video streams through the wireless channels. The backhaul network is modeled as a mesh network connecting BSs and MEC nodes and the cloud by dedicated high-speed wired fiber with fixed capacities. That is, each BS is physically connected to some MEC nodes and all MEC nodes have their dedicated backhaul links to the cloud. In this way, the near-end network can be represented by a directed graph $G(\mathcal{V}, \mathcal{L})$ where $\mathcal{V}$ includes the set of MEC nodes $\mathcal{M}$, the set of BSs $\mathcal{K}$, and the set of webcams $\mathcal{N}$. The $\mathcal{L}$ comprised of wired $\mathcal{L}^{wd}$ and wireless links $\mathcal{L}^{wl}$.

1) Wired link: The achievable backhaul capacity of the wired link $l_{kj} \in \mathcal{L}^{wd}$ between BS $k \in \mathcal{K}$ is denoted by r_{kj}. The achievable backhaul capacity between MEC node j and the cloud is denoted by $r_{j,cld}$. Each wired link is equally shared for video stream transmissions between both ends.

2) Wireless link: We assume that each webcam i is connected to its nearest BS k, denoted by $l_{ik} \in \mathcal{L}^{wl}$. The BS k has total W_k (in MHz) radio spectrum for wireless transmissions. Note that the interference management and power allocation are not the main goal of this paper, we assume that there is not inter-BS interference and fixed transmission power is adopted. According to the Shannon bound, the spectrum efficiency of wireless link l_{ik} is expressed as

$$\gamma_{ik} = \log_2\Big(1 + \frac{\rho_i g_{ik}}{\sigma_k^2}\Big), \tag{1}$$

where ρ_i is the transmission power of webcam i, g_{ik} is the channel gain between webcam i and BS k, and σ_k^2 is the power spectrum density of additive white Gaussian noise at BS k. Specifically, we do not consider the small-scale fading when modeling the SNR in this paper, which leads a stochastic optimization problem [19].

B. Computing Model

We know that video analytics could be processed by classical computer vision methods as well as deep neural networks (DNN). In the proposed DICE-IoT system, we assume that different decomposable inference pipelines could be performed at the cloud-edge framework to analyze the video streams uploaded by the webcams by either layer-level or model-level pipeline compositions. Here we introduce an available configuration set $\Phi = \{\phi\}$, where each specific decomposable pipeline ϕ is regarded as a *configuration*. For configuration ϕ, we denote its video resolution as $rs^\phi = w^\phi \times h^\phi$ and frame sampling rate as fr^ϕ (in fps), where w^ϕ and h^ϕ are input frame width and height to configuration ϕ, respectively. We introduce a binary indicator φ^ϕ, that is, $\varphi^\phi = 1$ if the video analytic task is partially/fully offloaded to the remote cloud. Otherwise, the full video analytic task is processed in the MEC stratum. Furthermore, we define a complete configuration set $\Phi^U(rs^\phi, fr^\phi, \varphi^\phi)$, which consists a full list of available configurations in the proposed system. Therefore, the isolated configuration set of the MEC stratum without any interaction with the cloud stratum is

TABLE II
IMPORTANT PARAMETERS OF CONFIGURATION ϕ

Notation	Definition
fr^ϕ	Frame sampling rate
rs^ϕ	Video resolution $rs^\phi = w^\phi \times h^\phi$
w^ϕ, h^ϕ	Frame {width, height}
φ^ϕ	Partial offloading indicator to the cloud
b^ϕ, b^ϕ_{cld}	Per frame data size {uploaded from a webcam to a MEC node, output from a MEC node to the cloud}
t^ϕ_j, t^ϕ_{cld}	Per frame processing time at {MEC node j, the cloud}
p^ϕ_j, p^ϕ_{cld}	Per frame power consumption at {MEC node j, the cloud}
u^ϕ_j, u^ϕ_{cld}	GPU utilization at {MEC node j, the cloud}
m^ϕ_j	GPU memory usage at MEC node j
q^ϕ	Detection accuracy

$$\Phi^{mec} = \{\phi : \varphi^\phi = 0, \forall \phi \in \Phi^U\}. \tag{2}$$

In the rest of the paper, the available configuration set Φ can be assigned to either Φ^U or Φ^{mec} according to different optimization problems.

The configuration selection impacts resource consumption, latency, and accuracy of the video analytics [13]. We list important parameters of configuration ϕ in Table II. The average value of those parameters could be measured with specific MEC node j from offline empirical analysis. Besides, we assume that the power consumption is proportional to the GPU utilization [20]. Therefore, the per frame power consumption at MEC node j and the cloud can be calculated by

$$\begin{cases} p_j^\phi = t_j^\phi u_j^\phi (p_j^{max} - p_j^{idle}) \\ p_{cld}^\phi = t_{cld}^\phi u_{cld}^\phi (p_{cld}^{max} - p_{cld}^{idle}) \end{cases}, \tag{3}$$

where p_j^{idle} is the power consumption in idle state of MEC node j and p_j^{max} is the maximum power consumption of MEC node j. Accordingly, p_{cld}^{idle} is the power consumption in the idle state of the cloud server and p_{cld}^{max} is the maximum power consumption of the cloud server.

IV. PROBLEM FORMULATION

To provide low latency and high accuracy video analytic services, adequate wireless bandwidth should be allocated to each webcam from its connecting BS and meanwhile, the

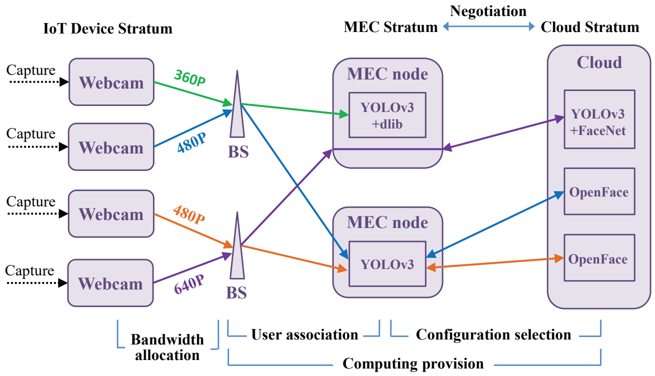


Fig. 2. Example of the proposed joint resource allocation problem.

MEC orchestrator determines the user association from the webcam to a specific MEC node, which is running a determined computing configuration to satisfy the service requirements. Specifically, we assume the MEC stratum is controlled by a single MEC orchestrator trading with the cloud controller for cooperative computing provisioning deployment. In this paper, we aim to maximize the expected utility of the MEC stratum by seeking the optimal **bandwidth allocation** and **computing provision** scheme. Fig. 2 illustrates an example adopting model-level pipeline composition, where the details will be described in Section VII.

A. Joint Resource Allocation

1) Bandwidth allocation: The bandwidth allocation is denoted as $\mathbf{a} = \{a_{ik}, \forall i, k\}$, where a_{ik} is a fraction of bandwidth allocated from BS k to webcam i. Therefore, the data rate of webcam i is

$$r_{ik} = a_{ik} W_k \gamma_{ik}, \quad \forall l_{ik} \in \mathcal{L}^{wl}. \tag{4}$$

Accordingly, we have the bandwidth allocation constraint

$$\text{C1} : \begin{cases} \text{I} : a_{ik} = 0, \quad \forall l_{ik} \notin \mathcal{L}^{wl} \\ \text{II} : a_{ik} \in [0,1], \quad \forall l_{ik} \in \mathcal{L}^{wl} \\ \text{III} : \sum\limits_{i \in \mathcal{N}} a_{ik} \leq 1, \quad \forall k \in \mathcal{K} \end{cases}. \tag{5}$$

2) Computing provision: According to the network links modeled by the graph $G(\mathcal{V}, \mathcal{L})$, the candidate MEC node set of webcam i is

$$\mathcal{M}_i^c = \{j : l_{ik} \in \mathcal{L}^{wl} \wedge l_{kj} \in \mathcal{L}^{wd}, \forall k \in \mathcal{K}, \forall j \in \mathcal{M}\}. \tag{6}$$

Note that webcam i has its own accuracy requirement and it is also equipped with some physical limitations: *i)* maximum frame width w_i^{max} and height h_i^{max}, and *ii)* maximum frame sampling rate fr_i^{max}. Therefore, the candidate configuration set of webcam i is

$$\begin{aligned} \Phi_i^c = \{\phi : w^\phi \leq w_i^{max} \wedge h^\phi \leq h_i^{max} \\ \wedge fr^\phi \leq fr_i^{max} \wedge q^\phi \geq Q_i^{req}, \forall \phi \in \Phi\}. \end{aligned} \tag{7}$$

The target of the computing provision is to determine the configurations of MEC nodes and user association between webcams and MEC nodes. We introduce binary variables $\mathbf{x} = \{x_{ij}^\phi, \forall i, j, \phi\}$ as the user association decision. The $x_{ij}^\phi = 1$ when webcam i is associated with MEC node j for video analytics using configuration ϕ; otherwise, $x_{ij}^\phi = 0$. We have the **user association** constraint

$$\text{C2} : \begin{cases} \text{I} : \sum\limits_{j \in \mathcal{M}_i^c} \sum\limits_{\phi \in \Phi_{ij}^c} x_{ij}^\phi \leq 1, \quad \forall i \\ \text{II} : \sum\limits_{j \in \mathcal{M} \setminus \mathcal{M}_i^c} \sum\limits_{\phi \in \Phi} x_{ij}^\phi = 0, \quad \forall i \\ \text{III} : \sum\limits_{j \in \mathcal{M}} \sum\limits_{\phi \in \Phi \setminus \Phi_i^c} x_{ij}^\phi = 0, \quad \forall i \end{cases}. \tag{8}$$

Note that a MEC node may need to process other applications and tasks, we assume that MEC node j has available GPU utilization u_j^{avl} and GPU memory m_j^{avl} for video analytics. In the proposed system, a MEC node is allowed to operate multiple configuration pipelines at the same time if it can afford sufficient computing capacities and resources. Besides, *the proposed framework allows pipeline sharing for GPU memory usage reduction*, that is, those pipelines with the same configuration at a MEC node could be shared by multiple webcams, as shown in Fig. 2. In this way, let $\mathbf{n} = \{n_j^\phi, \forall j, \phi\}$ represent the configuration selections of MEC nodes, where n_j^ϕ indicates that total n_j^ϕ pipelines with configuration ϕ are operated at MEC node j. As a webcam is served by at most one pipeline, the number of pipelines n_j^ϕ should not exceed the number of serving webcams $\sum_{i \in \mathcal{N}} x_{ij}^\phi$. Due to the limitation of available GPU utilization and memory, n_j^ϕ is bounded by $N_j^\phi = \min\{\lfloor u_j^{avl}/u_j^\phi \rfloor, \lfloor m_j^{avl}/m_j^\phi \rfloor, |\mathcal{N}|\}$, i.e., $n_j^\phi \in \{0, 1, \cdots, N_j^\phi\}$. From the above derivations, the **configuration selection** constraint is presented as

$$\text{C3} : \begin{cases} \text{I} : n_j^\phi \leq N_j^\phi, \quad \forall j, \phi \\ \text{II} : n_j^\phi \leq \sum_{i \in \mathcal{N}} x_{ij}^\phi, \quad \forall j, \phi \\ \text{III} : \sum\limits_{\phi \in \Phi} n_j^\phi u_j^\phi \leq u_j^{avl}, \quad \forall j \\ \text{IV} : \sum\limits_{\phi \in \Phi} n_j^\phi m_j^\phi \leq m_j^{avl}, \quad \forall j \end{cases}. \tag{9}$$

Note that multiple webcams share pipelines when selecting MEC node j with configuration ϕ, MEC node j needs to maintain a service queue for arrival video analytic tasks. Therefore, we define the following constraint to guarantee its stability of service:

$$\begin{aligned} \text{C4} : \quad & fr^\phi \cdot \sum_{i \in \mathcal{N}} x_{ij}^\phi \leq n_j^\phi / t_j^\phi \\ & \Rightarrow fr^\phi \cdot t_j^\phi \sum_{i \in \mathcal{N}} x_{ij}^\phi \leq n_j^\phi, \quad \forall j, \phi, \end{aligned} \tag{10}$$

that is, the total arrival rate to a pipeline should not be greater than its available service rate.

B. Latency Model

Since the final analytic results should be collected to the MEC stratum for further control and management, we consider one-way delay of uplink transmission and round-trip time (RTT) between the MEC stratum and the cloud stratum. Besides, we ignore the transmission delay from the cloud stratum to the MEC stratum because the final analytic results only contain small number of data. Therefore, if the decision

$x_{ij}^{\phi} = 1$, the per frame analytic latency of webcam i can be expressed as

$$L_{ij}^{\phi} = d_{ik}^{\phi,trans} + d_{kj}^{\phi,trans} + d_{j}^{\phi,proc} + \varphi^{\phi}(2d_{j,cld}^{prog} + d_{j,cld}^{\phi,trans} + d_{cld}^{\phi,proc}), \tag{11}$$

where the $d_{j}^{\phi,proc}$ and $d_{cld}^{\phi,proc}$ are per frame processing delay at MEC node j and the cloud, respectively. The $d_{ik}^{\phi,trans}$, $d_{kj}^{\phi,trans}$ and $d_{j,cld}^{\phi,trans}$ are per frame transmission delay from webcam i to BS k, from BS k to MEC node j, and from MEC node j to the cloud, respectively. Since the cloud stratum is remote to the edge stratum, we assume that there is a propagation delay $d_{j,cld}^{prog}$ when transmitting the packet through the backhaul medium from MEC node j to the cloud stratum.

To satisfy the latency requirement of webcam i, we have the following constraint

$$\text{C5}: x_{ij}^{\phi}(L_{i}^{req} - L_{ij}^{\phi}) \geq 0, \quad \forall i, j, \phi. \tag{12}$$

When allocating bandwidth fraction a_{ik} to webcam i, its transmission delay to BS k will be

$$d_{ik}^{\phi,trans} = \frac{b^{\phi}}{r_{ik}} = \frac{b^{\phi}}{a_{ik} W_{k} \gamma_{ik}}. \tag{13}$$

The transmission delay of the shared wired link l_{kj} can be measured by

$$d_{kj}^{\phi,trans} = \frac{\sum\limits_{l_{i'k} \in \mathcal{L}^{wl}} \sum\limits_{\phi \in \Phi} x_{i'j}^{\phi} b^{\phi}}{r_{kj}}. \tag{14}$$

Similarly, the transmission delay from MEC node j to the cloud is calculated by

$$d_{j,cld}^{\phi,trans} = \frac{\sum\limits_{i' \in \mathcal{N}} \sum\limits_{\phi \in \Phi} x_{i'j}^{\phi} b_{cld}^{\phi}}{r_{j,cld}}. \tag{15}$$

Recall that multiple webcams are allowed to share those pipelines with the same configuration at a MEC node. Considering the worst case that the MEC node receives video frames from multiple webcams almost at the same time, the processing delay, including queuing delay and service delay, at MEC node j could be represented by

$$d_{j}^{\phi,proc} = t_{j}^{\phi}\Big(\sum_{i' \in \mathcal{N}} x_{i'j}^{\phi} - n_{j}^{\phi}\Big) + t_{j}^{\phi}. \tag{16}$$

Thanks to sufficient storage and computing resources in the cloud stratum, we assume that the analytic task from each webcam can be processed in an individual pipeline at the cloud. Therefore, the processing delay at the cloud will be

$$d_{cld}^{\phi,proc} = t_{cld}^{\phi}, \tag{17}$$

which implies that there is no queuing delay.

Moreover, since the data rate of each link, i.e., wired link or wireless link, cannot exceed its transmission capacity, we have

$$\text{C6}: \sum_{j \in \mathcal{M}} \sum_{\phi \in \Phi} x_{ij}^{\phi} fr^{\phi} b^{\phi} \leq r_{ik}, \quad \forall l_{ik} \in \mathcal{L}^{wl}, \tag{18}$$

$$\text{C7}: \begin{cases} \text{I}: \sum\limits_{l_{ik} \in \mathcal{L}^{wl}} \sum\limits_{\phi \in \Phi} x_{ij}^{\phi} fr^{\phi} b^{\phi} \leq r_{kj}, & \forall l_{kj} \in \mathcal{L}^{wd} \\ \text{II}: \sum\limits_{i \in \mathcal{N}} \sum\limits_{\phi \in \Phi} x_{ij}^{\phi} fr^{\phi} b_{cld}^{\phi} \leq r_{j,cld}, & \forall j \end{cases}. \tag{19}$$

C. Utility Function

1) Cloud stratum: Once the computing provision is decided, the per-second operation cost of the cloud is

$$C_{cld} = \gamma_{cld} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \sum_{\phi \in \Phi} x_{ij}^{\phi} p_{cld}^{\phi} \cdot fr^{\phi}, \tag{20}$$

where γ_{cld} is a positive constant converting power consumption to cost at the cloud. Furthermore, we define the per-second effort of the cloud as the total GPU utilization

$$e_{cld} = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \sum_{\phi \in \Phi} x_{ij}^{\phi} t_{cld}^{\phi} u_{cld}^{\phi} \cdot fr^{\phi}. \tag{21}$$

To motivate the cloud to the optimal computing provisioning deployment, the MEC stratum pays π to compensate the per unit effort [21] afforded by the cloud. Therefore, the expected utility of the cloud will be

$$U_{cld} = R_{cld} - C_{cld}, \tag{22}$$

where $R_{cld} = \pi e_{cld}$ is the revenue.

2) MEC stratum: Accordingly, we can calculate the operation cost of the MEC stratum as follows:

$$C_{mec} = \gamma_{mec} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \sum_{\phi \in \Phi} x_{ij}^{\phi} p_{j}^{\phi} \cdot fr^{\phi}, \tag{23}$$

where γ_{mec} is a positive constant converting power consumption to cost at a MEC node. Since the practical operation behind the service is transparent to the IoT device, the service fee collected from a webcam is assumed to be proportional to its user satisfaction, i.e., detection accuracy with the latency and accuracy requirements satisfied. We have the revenue of the MEC stratum

$$R_{mec} = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \sum_{\phi \in \Phi} x_{ij}^{\phi} \cdot \mathcal{P}(q^{\phi}), \tag{24}$$

where $\mathcal{P}(\cdot)$ is the predefined payment function of the service fee. Therefore, the expected utility of the MEC stratum will be

$$U_{mec} = R_{mec} - C_{mec} - P_{mec}, \tag{25}$$

where the total service fee $P_{mec} = R_{cld}$.

Based on the system model, we observe that the cloud control would like to increase π as large as possible. On the contrary, the MEC orchestrator expects to utilize the incentive of π to jointly deploy the optimal computing provision with the cooperation of the cloud stratum so as to achieve more utility than provide video analytic service alone. It motivates us to propose a **Nash bargaining** between the MEC orchestrator and the cloud controller.

V. Nash Bargaining

The bargainging problem is a non-cooperative game to address how the players share a surplus they have jointly generated to benefit themselves. As a basic two-player bargaining game used to model strategic interactions, Nash bargaining theory [22] provides an efficient and fair solution, called *Nash bargaining solution (NBS)*, to balance the utilities of both rational players.

In the proposed bargaining framework, the MEC orchestrator and the cloud controller bargain with each other and share a surplus when reaching an agreement. An agreement is represented by a feasible tuple $(\mathbf{a}, \mathbf{x}, \mathbf{n}, \pi)$. The Nash bargaining problem is defined as [22]

$$\begin{aligned} \mathcal{NB}: & \\ \max_{(U_{mec}, U_{cld}) \in \mathcal{U}} & \quad (U_{mec} - D_{mec})(U_{cld} - D_{cld}) \\ \text{s.t.} & \quad (U_{mec}, U_{cld}) \geq (D_{mec}, D_{cld}), \end{aligned} \tag{26}$$

where $\mathcal{U}$ is the feasible set of the utility pair (U_{mec}, U_{cld}) over all possible agreements. The D_{mec} and D_{cld} are the utilities of the MEC stratum and the cloud stratum under disagreement. Obviously, $D_{cld} = 0$ because the cloud stratum does nothing under disagreement.

Furthermore, we define a pair of utility (U^*_{mec}, U^*_{cld}) as a NBS which solves $\mathcal{NB}$. It should be noted that the NBS maximizes the *social welfare (SW)* [23], which is defined as the aggregate utility of the MEC stratum and the cloud stratum. Therefore, the objective is to select the optimal bandwidth allocation $\mathbf{a}$ and computing provision $\mathbf{x}$ so as to maximize the social welfare:

$$\begin{aligned} \mathcal{SW}: \quad \max_{\mathbf{a}, \mathbf{x}, \mathbf{n}} & \quad W\big(\mathbf{a}, \mathbf{x}, \mathbf{n}\big) = U_{mec} + U_{cld} \\ \text{s.t.} & \quad \text{C1, C2, C3, C4, C5, C6, C7} \\ & \quad \text{C8}: \Phi = \Phi^U \\ x^{\phi}_{ij} \in \{0, 1\}, & \quad n^{\phi}_j \in \{0, 1, \cdots, N^{\phi}_j\} \quad \forall i, j, \phi. \end{aligned} \tag{27}$$

According to (22)(25), the social welfare can be derived as

$$W\big(\mathbf{a}, \mathbf{x}, \mathbf{n}\big) = W\big(\mathbf{x}\big) = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \sum_{\phi \in \Phi} x^{\phi}_{ij} \Big(\mathcal{P}(q^{\phi}) - \Omega^{\phi}_j\Big), \tag{28}$$

where Ω^{ϕ}_j is the total operation cost, that is,

$$\text{C9}: \Omega^{\phi}_j = (\gamma_{mec} p^{\phi}_j + \gamma_{cld} p^{\phi}_{cld}) \cdot fr^{\phi}, \quad \forall j, \phi. \tag{29}$$

The derivation indicates that the social welfare is a linear combination of user association decision $\mathbf{x}$.

Similarly, the expected utility of the MEC stratum under the disagreement is calculated by

$$\begin{aligned} D_{mec} = \max_{\mathbf{a}, \mathbf{x}, \mathbf{n}} & \quad U_{mec} \\ \text{s.t.} & \quad \text{C1, C2, C3, C4, C5, C6, C7} \\ & \quad \text{C8}': \Phi = \Phi^{mec} \\ x^{\phi}_{ij} \in \{0, 1\}, & \quad n^{\phi}_j \in \{0, 1, \cdots, N^{\phi}_j\} \quad \forall i, j, \phi. \end{aligned} \tag{30}$$

Finally, the optimal tuple $\big(\mathbf{a}^*, \mathbf{x}^*, \mathbf{n}^*, \pi^*\big)$ of the NBS can be derived by

$$\left\{ \begin{array}{l} \mathbf{a}^*, \mathbf{x}^*, \mathbf{n}^* = \arg\max\limits_{\mathbf{a}, \mathbf{x}} W\big(\mathbf{a}, \mathbf{x}, \mathbf{n}\big) \\ \pi^* = \frac{1}{2e_{cld}} \Big[W\big(\mathbf{a}^*, \mathbf{x}^*, \mathbf{n}^*\big) - D_{mec} + 2C_{cld}\Big] \end{array} \right. . \tag{31}$$

VI. GBD-Based Social Welfare Calculation

We observe that the C5 is the main difficulty of $\mathcal{SW}$ because: *i)* mixed optimization variables, i.e., binary variable x^{ϕ}_{ij}, integer variable n^{ϕ}_j, and continuous variable a_{ik}; *ii)* second order terms, e.g., x^{ϕ}_{ij}/a_{ik} and $x^{\phi}_{ij} n^{\phi}_j$. That is to say, $\mathcal{SW}$ is a mixed-integer nonlinear programming (MINLP) [24] problem, which is intractable to be solved.

A. Linearization with RLT

In order to reduce the dififculty, we utilize the *reformulation linearization technique (RLT)* [25] to reformulate the problem. Firstly, to avoid divide-by-zero caused by a_{ik}, we introduce a microscale θ added to a_{ik} and then represent $a_{ik} + \theta$ by $\hat{a}_{ik}$. According to C1, the auxiliary variable $\hat{\mathbf{a}} = \{\hat{a}_{ik}, \forall i, j, \phi\}$ is constrained by

$$\text{C1}': \left\{ \begin{array}{l} \text{I}: \hat{a}_{ik} = \theta, \quad \forall l_{ik} \notin \mathcal{L}^{wl} \\ \text{II}: \hat{a}_{ik} \in [\theta, 1 + \theta], \quad \forall l_{ik} \in \mathcal{L}^{wl} \\ \text{III}: \sum\limits_{i \in \mathcal{N}} \hat{a}_{ik} \leq 1 + \theta |\mathcal{N}|, \quad \forall k \in \mathcal{K} \end{array} \right. . \tag{32}$$

Next, we adopt RTL to linearize the constraints. Let $\mathbf{y} = \{y^{\phi}_{ii'j}, \forall i, i', j, \phi\}$, where $y^{\phi}_{ii'j}$ denotes the product term $x^{\phi}_{ij} x^{\phi}_{i'j}$, and then the bound-factor product constraints of $\mathbf{y}$ can be derived by

$$\Xi^{\phi, y}_{ii'j} = \left\{ \begin{array}{l} \text{I}: y^{\phi}_{ii'j} \leq x^{\phi}_{ij} \\ \text{II}: y^{\phi}_{ii'j} \leq x^{\phi}_{i'j} \\ \text{III}: y^{\phi}_{ii'j} \geq x^{\phi}_{ij} + x^{\phi}_{i'j} - 1 \end{array} \right. . \tag{33}$$

Similarly, let $\mathbf{z} = \{z^{\phi}_{ij}, \forall i, j, \phi\}$, where z^{ϕ}_{ij} denotes the product term $x^{\phi}_{ij} n^{\phi}_j$, therefore, the bound-factor product constraints of $\mathbf{z}$ can be derived by

$$\Xi^{\phi, z}_{ij} = \left\{ \begin{array}{l} \text{I}: z^{\phi}_{ij} \geq 0 \\ \text{II}: z^{\phi}_{ij} \leq x^{\phi}_{ij} N^{\phi}_j \\ \text{III}: z^{\phi}_{ij} \leq n^{\phi}_j \\ \text{IV}: z^{\phi}_{ij} \geq n^{\phi}_j - (1 - x^{\phi}_{ij}) N^{\phi}_j \end{array} \right. . \tag{34}$$

So far, substituting $\hat{a}_{ik}$, $y^{\phi}_{ii'j}$ and z^{ϕ}_{ij} into C5 and C6, we have

$$\text{C5}': \left(\begin{array}{l} x^{\phi}_{ij} \Big(t^{\phi}_j + 2\varphi^{\phi} d^{prog}_{j,cld} + t^{\phi}_{cld} - L^{req}_i\Big) \\ \quad + \frac{x^{\phi}_{ij} b^{\phi}}{\hat{a}_{ik} W_k \gamma_{ik}} \\ \quad + \sum\limits_{l_{i'k} \in \mathcal{L}^{wl}} \sum\limits_{\phi \in \Phi} y^{\phi}_{ii'j} \frac{b^{\phi}}{r_{kj}} \\ \quad + \sum\limits_{i' \in \mathcal{N}} \sum\limits_{\phi \in \Phi} y^{\phi}_{ii'j} \frac{b^{\phi}_{cld}}{r_{j,cld}} \\ \quad + t^{\phi}_j \sum_{i' \in \mathcal{N}} y^{\phi}_{ii'j} \\ \quad - t^{\phi}_j z^{\phi}_{ij}, \qquad \forall i, j, \phi \end{array} \right) \leq 0, \tag{35}$$

$$\text{C6}': \sum_{j \in \mathcal{M}} \sum_{\phi \in \Phi} x^{\phi}_{ij} fr^{\phi} b^{\phi} \leq \hat{a}_{ik} W_k \gamma_{ik}, \quad \forall l_{ik} \in \mathcal{L}^{wl}. \tag{36}$$

Finally, we reformulate the original optimization problem $\mathcal{SW}$

as

$$\mathcal{SW}' : \max_{\hat{\mathbf{a}},\mathbf{x},\mathbf{n},\mathbf{y},\mathbf{z}} \quad W(\mathbf{x}) \tag{37}$$
$$\begin{aligned} \text{s.t. } & \mathrm{C1}', \mathrm{C2}, \mathrm{C3}, \mathrm{C4}, \mathrm{C5}', \mathrm{C6}', \mathrm{C7}, \mathrm{C8}, \mathrm{C9} \\ & x_{ij}^{\phi} \in \{0,1\}, \quad n_j^{\phi} \in \{0,1,\cdots,N_j^{\phi}\} \quad \forall i,j,\phi \\ & y_{ii'j}^{\phi} \in \Xi_{ii'j}^{\phi,y}, \quad z_{ij}^{\phi} \in \Xi_{ij}^{\phi,z} \quad \forall i,i',j,\phi. \end{aligned}$$

B. Solution Using GBD

Let $\mathbb{X}$ represent the set of discrete variables, i.e., $\mathbb{X} = (\mathbf{x},\mathbf{n},\mathbf{y},\mathbf{z})$. We leverage *Generalized Benders Decomposition (GBD)* [26] to solve $\mathcal{SW}'$ optimally. The basic idea of GBD is to decompose a MINLP problem into two subproblems, a *primal problem* for linear/nonlinear programming and a *master problem* for pure integer programming, and then iteratively solve them with guaranteed convergence [24].

1) Primal problem: The primer problem $\mathcal{SP}$ corresponds to $\mathcal{SW}'$ by fixing the set of discrete variables as $\mathbb{X}^{(\nu)}$ in each iteration, where ν stands for the iteration counter. We integrate the constraints $\mathrm{C5}'$ and $\mathrm{C6}'$ by

$$G\big(\hat{\mathbf{a}},\mathbb{X}^{(\nu)}\big) = \begin{pmatrix} \sum\limits_{j\in\mathcal{M}}\sum\limits_{\phi\in\Phi} x_{ij}^{\phi(\nu)} fr^{\phi} b^{\phi} - \hat{a}_{ik} W_k \gamma_{ik}, \\ \forall l_{ik} \in \mathcal{L}^{wl} \\ \hat{a}_{ik} H\big(\mathbb{X}^{(\nu)}\big) + \frac{x_{ij}^{\phi(\nu)} b^{\phi}}{W_k \gamma_{ik}}, \quad \forall i,j,\phi \end{pmatrix}_{\mathcal{D}\times 1}, \tag{38}$$

where $\mathcal{D}$ represents the dimension $|\mathcal{N}| + |\mathcal{N}||\mathcal{M}||\Phi|$ and

$$H\big(\mathbb{X}^{(\nu)}\big) = \begin{pmatrix} x_{ij}^{\phi(\nu)}\Big(t_j^{\phi} + 2\varphi^{\phi} d_{j,cld}^{prog} + t_{cld}^{\phi} - L_i^{req}\Big) \\ + \sum\limits_{l_{i'k}\in\mathcal{L}^{wl}}\sum\limits_{\phi\in\Phi} y_{ii'j}^{\phi(\nu)} \frac{b^{\phi}}{r_{kj}} \\ + \sum\limits_{i'\in\mathcal{N}}\sum\limits_{\phi\in\Phi} y_{ii'j}^{\phi(\nu)} \frac{b_{cld}^{\phi}}{r_{j,cld}} \\ + t_j^{\phi} \sum_{i'\in\mathcal{N}} y_{ii'j}^{\phi(\nu)} \\ - t_j^{\phi} z_{ij}^{\phi(\nu)}, \quad \forall i,j,\phi \end{pmatrix}.$$

Specifically, we observe that $\hat{a}_{ik}$ is not in the objective function of $\mathcal{SW}'$, which means that a standard primal problem could not be directly constructed. Therefore, we introduce slack variables $\boldsymbol{\alpha} = [\alpha_l, l = 1,2,\cdots,\mathcal{D}]$ and a modified primal problem can be formulated as an l_1-minimization problem:

$$\mathcal{SP}' : \min_{\hat{\mathbf{a}},\boldsymbol{\alpha}} \sum_{l=1}^{\mathcal{D}} \alpha_l \tag{39}$$
$$\begin{aligned} \text{s.t. } & \mathrm{C1}', \mathrm{C8} \\ & G\big(\hat{\mathbf{a}},\mathbb{X}^{(\nu)}\big) \le \boldsymbol{\alpha} \\ & \alpha_l \ge 0, \quad \forall l. \end{aligned}$$

Algorithm 1: GBD-Based Social Welfare Calculation

```
Initialization: $UBD = +\infty$, $LBD = -\infty$, $\mathcal{F} = \emptyset$,
  $\mathcal{IF} = \emptyset$, $\nu = 0$.
Select an initial feasible solution to $\mathcal{SW}'$: $\mathbb{X}^{(\nu)} = \mathbf{0}$.
Solve the modified primal problem $\mathcal{SP}'$ with fixed $\mathbb{X}^{(\nu)}$, and
  obtain optimal solution $\hat{\mathbf{a}}^{(\nu)}$ and its associated $\boldsymbol{\lambda}^{(\nu)}$.
Update $UBD = -W(\mathbf{x}^{(\nu)})$ and $\mathcal{F} = \mathcal{F} \cup \{\nu\}$.
while $UBD - LBD > \epsilon$ do
    Set $\nu = \nu + 1$.
    Solve the relaxed master problem $\mathcal{MP}$, and obtain
      optimal solution $\mathbb{X}^{(\nu)}$ and $\omega^{(\nu)}$.
    Update $LBD = \omega^{(\nu)}$.
    Solve the modified primal problem $\mathcal{SP}'$ with fixed $\mathbb{X}^{(\nu)}$,
      and obtain optimal solution $\hat{\mathbf{a}}^{(\nu)}$ and its associated
      $\boldsymbol{\lambda}^{(\nu)}$.
    if $\mathcal{SP}'$ is feasible then
        Update $UBD = \min\{UBD, -W(\mathbf{x}^{(\nu)})\}$.
        Update $\mathcal{F} = \mathcal{F} \cup \{\nu\}$.
    else
        Update $\mathcal{IF} = \mathcal{IF} \cup \{\nu\}$.
    end
end
Output: the optimal solution of $\mathcal{SW}' = (\hat{\mathbf{a}}^*, \mathbf{x}^*, \mathbf{n}^*, \mathbf{y}^*, \mathbf{z}^*)$.
```

In each iteration ν, we can obtain the continuous solution $\hat{\mathbf{a}}^{\nu}$ and its associated Lagrange multipliers $\boldsymbol{\lambda}^{(\nu)} = [\lambda_l^{(\nu)}, l = 1,2,\cdots,\mathcal{D}]$ by solving $\mathcal{SP}'$. In this way, a feasible solution to the original primal problem $\mathcal{SP}$ can be determined if $\sum_{l=1}^{\mathcal{D}} \alpha_l = 0$; otherwise, $\mathcal{SP}$ is infeasible. Accordingly, we let $\mathcal{F}$ and $\mathcal{IF}$ represent the sets of the iteration counter ν associated with feasible $\mathcal{SP}$ and infeasible $\mathcal{SP}$, respectively. Furthermore, the Lagrange function results from

$$\xi\big(\hat{\mathbf{a}}^{(\nu)},\mathbb{X},\boldsymbol{\lambda}^{(\nu)}\big) = \begin{cases} -W(\mathbf{x}) + \boldsymbol{\lambda}^{(\nu)T} G\big(\hat{\mathbf{a}}^{(\nu)},\mathbb{X}\big), & \nu \in \mathcal{F} \\ \boldsymbol{\lambda}^{(\nu)T} G\big(\hat{\mathbf{a}}^{(\nu)},\mathbb{X}\big), & \nu \in \mathcal{IF} \end{cases}. \tag{40}$$

2) Master problem: The relaxed master problem makes use of the Lagrange multipliers obtained in the primal problem so that an infinite number of cutting planes could be iteratively added as constraints in order to reduce its feasible region. The relaxed master problem is formulated as follows:

$$\mathcal{MP} : \min_{\mathbf{x},\mathbf{n},\mathbf{y},\mathbf{z},\omega} \quad \omega \tag{41}$$
$$\text{s.t. } \mathrm{C2}, \mathrm{C3}, \mathrm{C4}, \mathrm{C7}, \mathrm{C8}$$
$$\omega \ge \xi\big(\hat{\mathbf{a}}^{(\nu)},\mathbb{X},\boldsymbol{\lambda}^{(\nu)}\big) \quad \forall \nu \in \mathcal{F} \tag{42}$$
$$0 \ge \xi\big(\hat{\mathbf{a}}^{(\nu)},\mathbb{X},\boldsymbol{\lambda}^{(\nu)}\big) \quad \forall \nu \in \mathcal{IF} \tag{43}$$
$$\begin{aligned} & x_{ij}^{\phi} \in \{0,1\}, \quad n_j^{\phi} \in \{0,1,\cdots,N_j^{\phi}\} \quad \forall i,j,\phi \\ & y_{ii'j}^{\phi} \in \Xi_{ii'j}^{\phi,y}, \quad z_{ij}^{\phi} \in \Xi_{ij}^{\phi,z} \quad \forall i,i',j,\phi. \end{aligned}$$

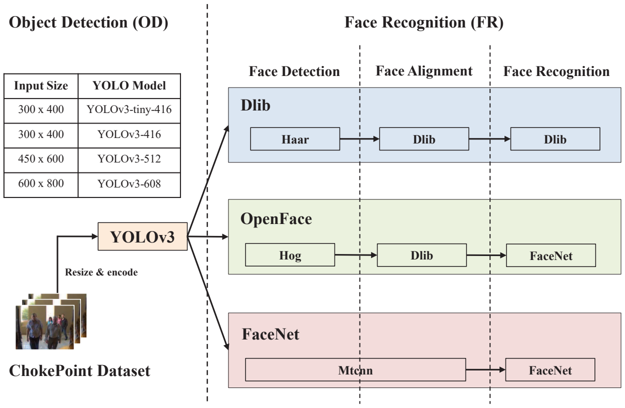
where (42)(43) are the feasible and infeasible cuts derived from (40) through the iterative process.

The overall description of the proposed GBD-based social welfare calculation is shown in Algorithm 1. We observe that Algorithm 1 requires an initial feasible solution to $\mathcal{SW}'$ so that $\mathcal{MP}$ will not be unbounded by adding a feasible cut obtained in $\mathcal{SP}'$. Obviously, $\mathbb{X}^{(\nu)} = \mathbf{0}$ is always feasible to $\mathcal{SW}'$. In the iterative process, the master problem $\mathcal{MP}$ provides the lower bound (LBD) and the feasible modified primer problem $\mathcal{SP}'$ regulates the upper bound (UBD). The optimal solution $\mathbb{X}^{(\nu)}$ obtained from $\mathcal{MP}$ is fixed and used subsequently in $\mathcal{SP}'$. The iterative process terminates when $UBD - LBD \le \epsilon$, where ϵ is the predefined convergence tolerance. The computational complexity of Algorithm 1 is affected by the cost of solving the modified primal problem

TABLE III
LIST OF MEASURED CONFIGURATION PARAMETERS

	Models		$\{0,1\}$	pixels	KB	KB	ms	ms	%	%	MiB	%
ϕ	Edge	Cloud	φ^{ϕ}	rs^{ϕ}	b^{ϕ}	b^{ϕ}_{cld}	t^{ϕ}_{j}	t^{ϕ}_{cld}	u^{ϕ}_{j}	u^{ϕ}_{cld}	m^{ϕ}_{j}	q^{ϕ}
1	YOLOv3-tiny + dlib	null	0	300 × 400	11.27	0.00	24.93	0.00	7.25	0.00	735	79.52
2	YOLOv3 + dlib	null	0	300 × 400	11.27	0.00	62.79	0.00	42.60	0.00	1923	83.36
3	YOLOv3 + dlib	null	0	450 × 600	21.29	0.00	81.86	0.00	54.40	0.00	2153	88.10
4	YOLOv3-tiny + OpenFace	null	0	300 × 400	11.27	0.00	30.21	0.00	6.53	0.00	801	79.18
5	YOLOv3 + OpenFace	null	0	300 × 400	11.27	0.00	67.26	0.00	42.10	0.00	1985	82.75
6	YOLOv3 + OpenFace	null	0	450 × 600	21.29	0.00	90.60	0.00	53.41	0.00	2215	90.47
7	null	YOLOv3-tiny + dlib	1	300 × 400	11.27	11.27	0.00	25.22	0.00	2.44	0	79.52
8	null	YOLOv3 + dlib	1	300 × 400	11.27	11.27	0.00	40.93	0.00	11.88	0	83.36
9	null	YOLOv3 + dlib	1	450 × 600	21.29	21.29	0.00	55.85	0.00	14.31	0	88.10
10	null	YOLOv3 + dlib	1	600 × 800	41.08	41.08	0.00	78.04	0.00	21.04	0	91.86
11	null	YOLOv3-tiny + OpenFace	1	300 × 400	11.27	11.27	0.00	38.11	0.00	2.38	0	79.18
12	null	YOLOv3 + OpenFace	1	300 × 400	11.27	11.27	0.00	51.15	0.00	11.83	0	82.75
13	null	YOLOv3 + OpenFace	1	450 × 600	21.29	21.29	0.00	75.10	0.00	14.40	0	90.47
14	null	YOLOv3 + OpenFace	1	600 × 800	41.08	41.08	0.00	109.80	0.00	21.03	0	92.49
15	null	YOLOv3-tiny + FaceNet	1	300 × 400	11.27	11.27	0.00	55.06	0.00	12.10	0	83.48
16	null	YOLOv3 + FaceNet	1	300 × 400	11.27	11.27	0.00	69.19	0.00	22.33	0	87.19
17	null	YOLOv3 + FaceNet	1	450 × 600	21.29	21.29	0.00	88.23	0.00	32.39	0	90.50
18	null	YOLOv3 + FaceNet	1	600 × 800	41.08	41.08	0.00	113.45	0.00	46.60	0	94.12
19	YOLOv3-tiny	dlib	1	300 × 400	11.27	3.34	19.69	7.43	5.97	0.78	455	79.52
20	YOLOv3	dlib	1	300 × 400	11.27	3.95	55.80	11.40	41.55	0.79	1639	83.36
21	YOLOv3	dlib	1	450 × 600	21.29	6.43	69.00	21.08	53.59	0.91	1869	88.10
22	YOLOv3	dlib	1	600 × 800	41.08	10.91	98.16	34.38	80.72	0.97	2561	91.86
23	YOLOv3-tiny	OpenFace	1	300 × 400	11.27	3.34	19.69	20.31	5.97	0.72	455	79.18
24	YOLOv3	OpenFace	1	300 × 400	11.27	3.95	55.80	21.62	41.55	0.73	1639	82.75
25	YOLOv3	OpenFace	1	450 × 600	21.29	6.43	69.00	40.33	53.59	0.82	1869	90.47
26	YOLOv3	OpenFace	1	600 × 800	41.08	10.91	98.16	66.15	80.72	0.85	2561	92.49
27	YOLOv3-tiny	FaceNet	1	300 × 400	11.27	3.34	19.69	37.26	5.97	9.62	455	83.48
28	YOLOv3	FaceNet	1	300 × 400	11.27	3.95	55.80	39.66	41.55	10.05	1639	87.19
29	YOLOv3	FaceNet	1	450 × 600	21.29	6.43	69.00	53.46	53.59	16.11	1869	90.50
30	YOLOv3	FaceNet	1	600 × 800	41.08	10.91	98.16	69.80	80.72	23.35	2561	94.12

Hints: Highlight rows with gray are selected configurations for numerical simulations



Input Size	YOLO Model
300 x 400	YOLOv3-tiny-416
300 x 400	YOLOv3-416
450 x 600	YOLOv3-512
600 x 800	YOLOv3-608

Fig. 3. Decomposable inference pipeline in the testbed.

$\mathcal{SP}'$ and the relaxed master problem $\mathcal{MP}$. Specifically, the computational complexity is dominated by the problem $\mathcal{MP}$, which could be solved via standard techniques, such as branch-and-bound (BB) or cutting planes. We assume that Algorithm 1 stops in K iterations, then K nonlinear programming (NLP) and K integer linear programming (ILP) are required to achieve the optimal solution. It should be noted that Algorithm 1 does not change the NP-hard property of the MINLP. Therefore, Algorithm 1 in the worst case may converge in an exponential number of iterations. Nevertheless, we provide the execution time of Algorithm 1 in the following evaluations to verify its efficiency in practice.

So far, the optimal social welfare can be calculated by Algorithm 1. Similarly, we solve the optimal disagreement utility D_{mec} using Algorithm 1 by replacing C8 with C8′ and optimizing U_{mec} instead of $W\big(\mathbf{a}, \mathbf{x}, \mathbf{n}\big)$. Finally, we obtain the optimal tuple $\big(\mathbf{a}^*, \mathbf{x}^*, \mathbf{n}^*, \pi^*\big)$ of the proposed Nash bargaining by (31).

VII. EVALUATION RESULTS

In this section, in order to evaluate the performance of the proposed DICE-IoT system, we implement a video analytic application on a real testbed as an example with the decomposable inference pipeline consisting of object detection and face recognition.

A. Parameter Measurement in Testbed

As shown in Fig. 3, we set up an experimental testbed, where either *YOLOv3* [10] or its lightweight version *YOLOv3-tiny* is applied to object detection with adjustable resolution of original video frames as input. The cropped face of any detected person is further pushed into face recognition by one of *dlib* [27], *OpenFace* [28], and *FaceNet* [11]. ChokePoint Dataset [29] is used as the inference input, which contains sequences of surveillance frames captured in the real world. The inference accuracy is defined as the weighted sum of both human detection quality calculated by Intersection over Union (IoU) and face recognition quality in F1 score. The relative weight is set to be 0.5.

To measure the parameters in Table II, the edge node equipped with an individual GPU instance, i.e., GeForce GTX 1060, is deployed over Openstack infrastructure. The cloud server is realized by Google Cloud Platform service, which provides multiple Tesla V100 GPU instances. It should be noted that even Raspberry Pi 3 Model B [30] can be an

TABLE IV
List of Key Simulation Parameters

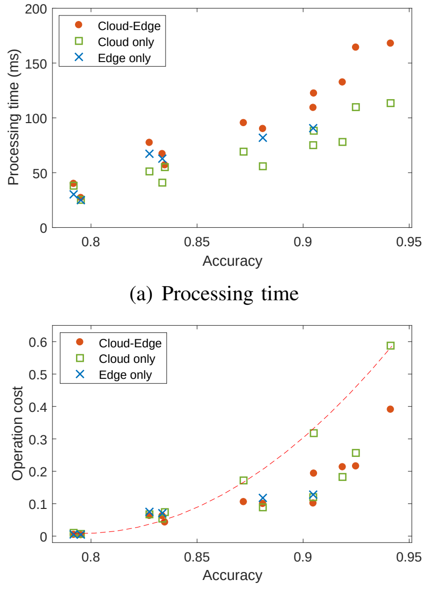
Parameter	Value
# of BS $\|\mathcal{K}\|$, MEC nodes $\|\mathcal{M}\|$	4, 4
Path loss model	$34 + 40\log(\text{dist.})$ dB
Log-normal shadowing	8 dB
Frequency bandwidth, W_k	unif(10, 20) MHz
Transmission power of webcams, ρ_i	23 dBm
The power of noise, σ_k^2	-174 dBm/Hz
Propagation delay to cloud, $d_{j,cld}^{prog}$	20 ms
Frame sampling rate, fr^{ϕ}	10
Backhaul capacity:	
i) BS to MEC node, r_{kj}	100 Mbps
ii) MEC node to the cloud, $r_{j,cld}$	50 Mbps
Power consumption to cost:	
i) At a MEC node, γ_{mec}	2.3874
ii) At the cloud, γ_{cld}	4.9613
Latency requirement, L_i^{req}	unif(50, 200) ms
Accuracy requirement, Q_i^{req}	unif(0.75, 0.92)
Available GPU utilization, u_j^{avl}	unif(0.5, 1)
Available GPU memory, m_j^{avl}	unif(1000, 3000) MiB
Convergence tolerance, ϵ	0.01
Microscale, θ	10^{-4}

Hints: unif = uniform distribution

alternative to execute live video analytics, however, it is not suitable to play the role of an edge computing due to the limited computing capacities and memory resources. As shown in Table III, we list 30 kinds of configurations that place the object detection and face recognition models in three different schemes, respectively: *1) Edge Only*: the inference pipeline is completely executed at the MEC stratum; *2) Cloud Only*: the inference pipeline is completely executed at the cloud stratum; *3) Cloud-Edge*: the inference pipeline is decomposed to both the MEC stratum and cloud stratum. From a large offline empirical analysis, the parameters of configurations are measured in Table III. Moreover, we measure the power consumption of GeForce GTX 1060 and Tesla V100 in idle state, that is, p_j^{idle} is 9.15W and p_{cld}^{idle} is 25.95W. According to their datasheets, the maximum power consumption p_j^{max} and p_{cld}^{max} are 120W and 250W, respectively. Then, the per-frame power consumption p_j^{ϕ} and p_{cld}^{ϕ} can be calculated by (3).

B. Numerical Simulations

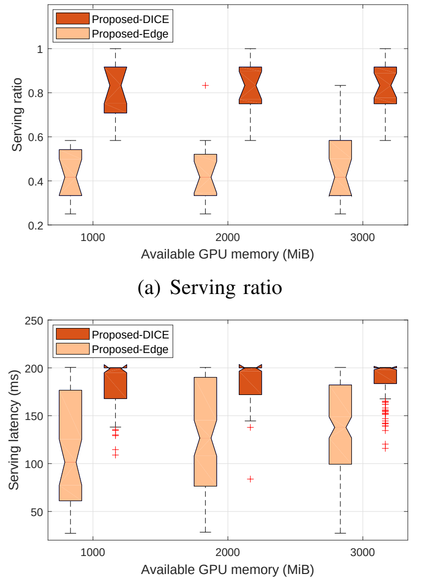
Based on the measured parameters in Table III, we evaluate the DICE-IoT system through simulations. We consider that $|\mathcal{K}|$ BSs are located in grid topology within an 800m×800m square area, that is, each BS is located in the center of a small square grid. At the MEC stratum, we deploy $|\mathcal{M}|$ MEC nodes and each BS is allowed to randomly access to at most 2 MEC nodes. Total $|\mathcal{N}|$ webcams are randomly and uniformly distributed within the range and each of them connects to its nearest BS. The key simulation parameters by default are list in Table IV. The modified primal problem $\mathcal{SW}'$ and the relaxed master problem $\mathcal{MP}$ are solved by the solver GUROBI [31] via YALMIP [32] on an Intel Core i7 3.6 GHz processor with 16 GB RAM. Specifically, cutting plane technique is adopted in YALMIP to solve the relaxed master problem $\mathcal{MP}$. All evaluation results are presented for one-hour analytic services. We compare six approaches: *1) Proposed-DICE*, *2) Proposed-Edge*, *3) Dedicated-DICE* *4) Dedicated-Edge*, *5) Greedy*, and *6) VideoEdge*, where the "*DICE*" and "*Edge*" represent the proposed DICE-IoT framework and the *Edge Only* system, respectively. Compared with the *Proposed* framework, the proposed pipeline sharing mechanism is disabled in the *Dedicated* approach, that is, each webcam is served by a dedicated pipeline. The *Greedy* adopts latency-aware heuristic instead of the GBD-based approach under the proposed DICE-IoT framework. More specifically, *Greedy* gives priority to determine service deployment to the webcam who has higher latency requirement, i.e., lower L_i^{req}, until no more webcams could be added. *VideoEdge* is the state-of-the-art framework



(a) Processing time

(b) Operation cost

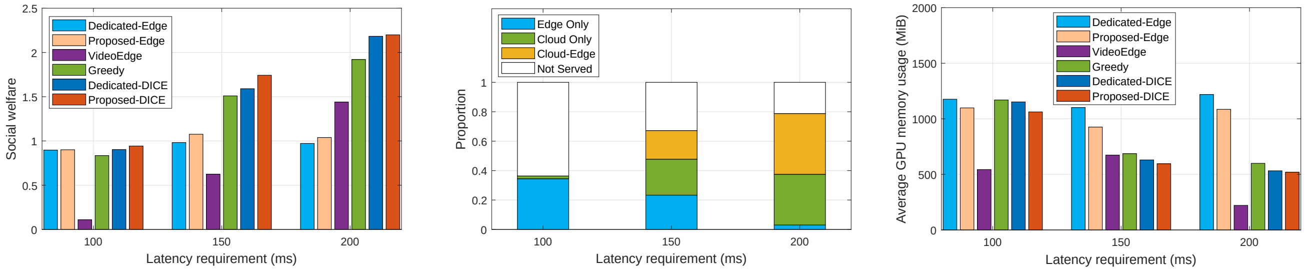
Fig. 4. Configuration Tradeoff Analysis.



(a) Serving ratio

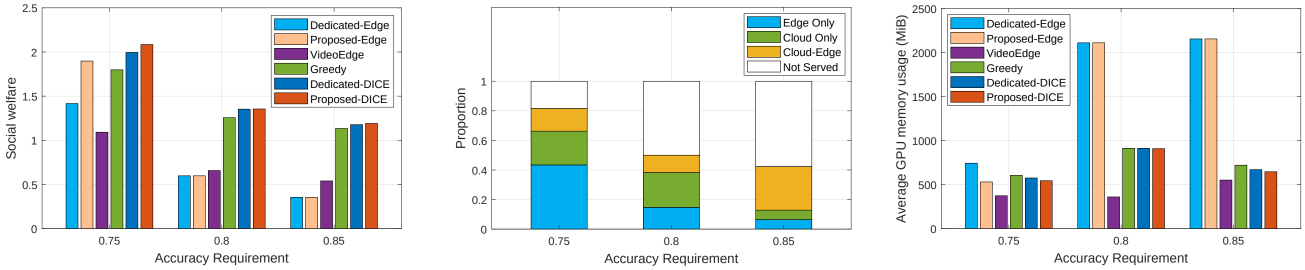
(b) Serving latency

Fig. 5. Performance versus available GPU memory: $|\mathcal{N}| = 12$, $u_j^{avl} = 1$, $L_i^{req} = 200$ms.



(a) Social welfare (b) Active proportion of *Proposed-DICE* (c) Avg. GPU memory utilization (MEC stratum)

Fig. 6. Performance versus latency requirement: $|\mathcal{N}| = 12$.



(a) Social welfare (b) Active proportion of *Proposed-DICE* (c) Avg. GPU memory utilization (MEC stratum)

Fig. 7. Performance versus accuracy requirement: $|\mathcal{N}| = 12$.

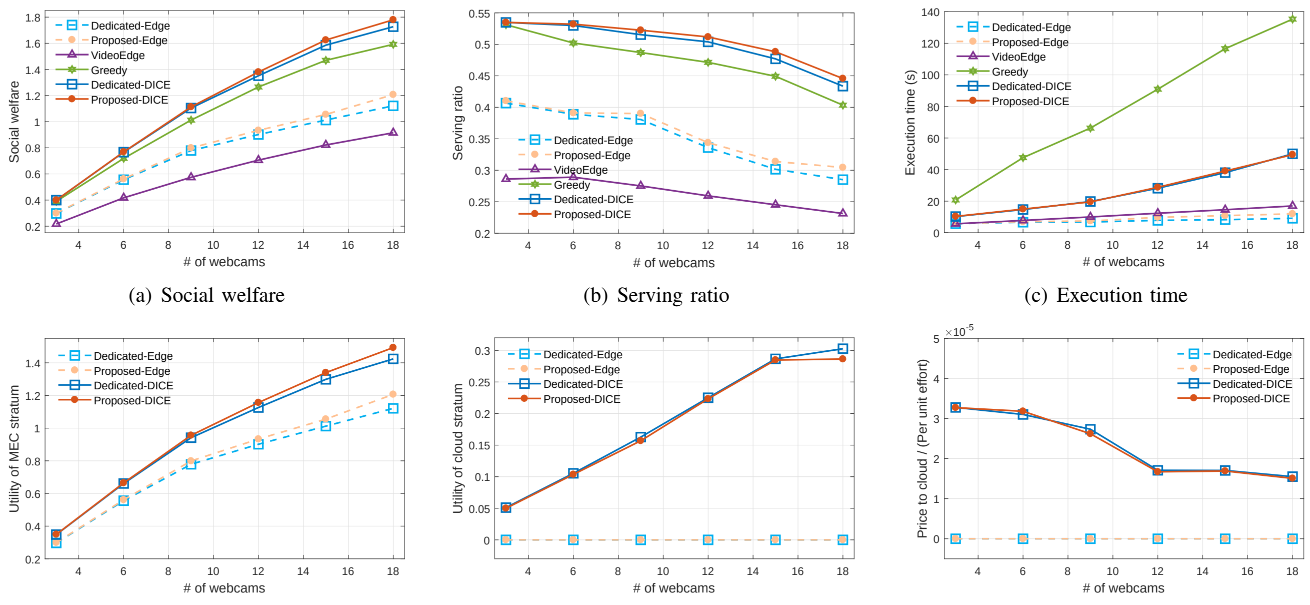
[16] using a greedy heuristic approach. Recall that *VideoEdge* doesn't consider the wireless network links as well as the latency of the analytic task. Therefore, we iteratively perform bandwidth allocation according to the results of *VideoEdge* by utilizing the proposed $\mathcal{SP}'$ and each time an active webcam with unsatisfied serving delay is changed into inactive until the delay requirements of all active webcams are satisfied. It is noted that the NBS is not held in either *Greedy* or *VideoEdge* since they could not achieve the optimal solution.

The extensive simulation results are provided as follows:

1) Configuration: We first present the tradeoff analysis among different configurations in the computing aspect, as shown in Fig. 4. We observe that both the processing time and the operation costs increase as the accuracy grows. Specifically, *Cloud Only* scheme achieves less processing time but requires more operation costs. Besides, *Edge Only* scheme could not provide high accuracy service since the edge node is not as powerful as the cloud. We define the payment function $\mathcal{P}(q^{\phi})$ as an accuracy-cost curve plus fixed 0.2 vertical offset, where the curve is constructed by those envelope points using polynomial curve fitting (See red dotted curve in Fig. 4(b)). We find out that some configurations are dominated by others in both processing time and operation costs, therefore, we select 3 configurations for each scheme (See highlight rows with gray in Table III) in the following simulations.

2) Available GPU memory: We assign all MEC nodes with the same available GPU memory, i.e., m_j^{avl}, and illustrate the performance versus available GPU memory, which is the key limitation to the capacity of DL-based video analytic services. In Fig. 5, a total of 30 independent simulations is performed and then presented by a boxplot. The *serving ratio* is defined as the ratio of active webcams successfully served by the framework over the total number. The serving latency is the actual service delay that the active webcams will suffer from. As shown in Fig. 5, generally the serving ratio increases and latency decreases against the available GPU memory. Besides, in Fig. 5(a), we observe that *Proposed-DICE* can serve more webcams than *Proposed-Edge* thanks to the cooperative computing provision with the cloud. However, as shown in Fig. 5(b), *Proposed-DICE* results in higher service latency compared with *Proposed-Edge* because: on the one hand, task offloaded to the remote cloud introduces additional wired transmission delay and long-distance propagation delay; on the other hand, more active webcams in *Proposed-DICE* share the limited wireless link bandwidth, wired link capacities, as well as the computing resources of the MEC stratum. Nevertheless, the serving latency of the active webcams in *Proposed-DICE* never exceeds 200ms maximum tolerance.

3) Latency requirement: In Fig. 6, we adjust the latency requirement of all webcams, i.e., L_i^{req}, and demonstrate the performance versus latency requirement. Intuitively, the social welfare increases as the relaxation of the latency tolerance in Fig 6(a). Specifically, the social welfare under either *Proposed-Edge* or *Dedicated-Edge* is not always increasing due to the limited resources at the MEC stratum. Due to the proposed pipeline sharing mechanism, *Proposed-DICE* and *Proposed-Edge* outperforms *Dedicated-DICE* and *Dedicated-Edge*, respectively. We also observe that the social welfare of *VideoEdge* is sensitive to the latency requirement since the latency is not considered in this approach. Besides, the active proportion of *Proposed-DICE* is shown in Fig. 6(b). As the latency requirement grows, we find out that the total number of active webcams increases and meanwhile the system prefers to serve active webcams by either *Cloud Only* or *Cloud-Edge* instead of *Edge Only*. That is because *Cloud Only* and *Cloud-*



(a) Social welfare

(b) Serving ratio

(c) Execution time

(d) Utility of the MEC stratum

(e) Utility of the cloud stratum

(f) Price to the cloud

Fig. 8. Performance versus the number of webcams.

Edge not only save resource usage at the MEC stratum but also support higher accuracy analytic services. In Fig. 6(c), the average GPU memory usage is calculated by the total GPU memory usage at the MEC stratum over the total number of active webcams. The results in Fig. 6(c) confirm that both the proposed pipeline sharing mechanism and the cooperative computing provision can reduce the GPU memory usage at the MEC stratum.

4) Accuracy requirement: Similarly, we adjust the accuracy requirement of all webcams, i.e., Q_i^{req}, and the performance versus accuracy requirement is shown in Fig. 7. The resource-quality tradeoff exposes that high accuracy requirement leads to more resource consumption and larger processing time. Therefore, the social welfare, as well as the number of active webcams, decreases against the accuracy requirement in Fig. 7(a) and Fig. 7(b). Nevertheless, both *Proposed-DICE* and *Dedicated-DICE* can achieve larger social welfare than *Proposed-Edge* and *Dedicated-Edge* under high accuracy requirement. In Fig. 7(b), we observe that *Edge Only* could not support high accuracy video analytics and *Cloud-Edge* has an advantage over *Edge Only* at this time. Besides, when the accuracy requirement is higher than 0.8, the average GPU memory usage of the MEC stratum can be greatly reduced with the assistance of the cloud stratum and further utilized to other applications, as shown in Fig. 7(c).

5) The number of webcams: Finally, we discuss the performance versus the number of webcams in Fig. 8. In general, the social welfare, the utility of the MEC stratum, the utility of the cloud stratum increase as the number of webcams grows, as shown in Fig. 8(a), Fig. 8(d) and Fig. 8(e). On the contrary, the serving ratio decreases in Fig. 8(b). The results in Fig. 8(a) and Fig. 8(b) show that the social welfare and the serving ratio of *VideoEdge* are even worse than *Edge Only* scheme because the latency of the analytic task is not considered and those active webcams in *VideoEdge* may still fail to be served by the framework due to the unsatisfied serving delay. Specifically, we observe from Fig. 8(b), Fig. 8(d) and Fig. 8(e) that the NBS provides a win-win-win solution to the IoT device, the MEC, and the cloud stratums thanks to the cooperative computing provision. That is, compared with *Edge Only* (non-cooperative) scheme, the proposed decomposable cloud-edge framework can provide more service opportunities to those IoT devices and meanwhile increase the utilities of both the MEC and the cloud stratums. Moreover, the proposed pipeline sharing mechanism also improves performance by comparing *Proposed-DICE*/*Proposed-Edge* with *Dedicated-DICE*/*Dedicated-Edge*. Under the agreement of the proposed Nash bargaining, the price to the cloud to compensate its per unit effort, i.e., π, is shown in Fig. 8(f), where the negotiated price decreases against the number of webcams. According to the NBS in (31), the reason is that the cloud stratum distributes a lot of effort as the number of webcams increases but the social welfare is improved little. Moreover, we show in Fig. 8(c) that the proposed GBD-based approach can greatly reduces the execution time compared with *Greedy*.

VIII. Conclusion

In this paper, we propose a DICE-IoT framework for live video analytics with joint latency and accuracy awareness. We provide a Nash bargaining between the MEC and the cloud to incentivize cooperative computing provision. A GBD-based approach is furthered utilized to optimize social welfare. We evaluate and discuss the key factors affecting system performance. The results show that the proposed intelligence framework can achieve a win-win-win solution to the IoT device, the MEC, and the cloud stratums. Thanks to the proposed pipeline sharing mechanism, the proposed framework outperforms other approaches in terms of social welfare, serving ratio, and GPU memory usage reduction. Moreover, it is noted that the proposed DICE-IoT framework is not limited to live video analytic applications, which could be further applied to general cloud-edge IoT framework subject to both latency and accuracy requirements after some modifications.

Acknowledgment

This work was supported by the Ministry of Science and Technology under Grant MOST 108-2218-E-002-060-MY3, 108-2221-E-002-033-MY3, 105-2221-E-001-003-MY3 and 108-2628-E-001-003-MY3, and the Academia Sinica under Thematic Research Grant.

References

[1] J. Chen, K. Li, Q. Deng, K. Li, and S. Y. Philip, "Distributed deep learning model for intelligent video surveillance systems with edge computing," *IEEE Transactions on Industrial Informatics*, 2019.

[2] K. Gai and M. Qiu, "Reinforcement learning-based content-centric services in mobile sensing," *IEEE Network*, vol. 32, no. 4, pp. 34–39, 2018.

[3] K. Gai, M. Qiu, and H. Zhao, "Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing," *Journal of Parallel and Distributed Computing*, vol. 111, pp. 126–135, 2018.

[4] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," *Journal of Network and Computer Applications*, vol. 59, pp. 46–54, 2016.

[5] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[6] Y.-Y. Shih, W.-H. Chung, A.-C. Pang, T.-C. Chiu, and H.-Y. Wei, "Enabling low-latency applications in fog-radio access networks," *IEEE network*, vol. 31, no. 1, pp. 52–58, 2016.

[7] B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, "Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 56–62, 2019.

[8] Y. Huang, F. Wang, F. Wang, and J. Liu, "Deepar: A hybrid device-edge-cloud execution framework for mobile deep learning applications," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 892–897.

[9] D. Crankshaw, G.-E. Sela, C. Zumar, X. Mo, J. E. Gonzalez, I. Stoica, and A. Tumanov, "Inferline: Ml inference pipeline composition framework," *arXiv preprint arXiv:1812.01776*, 2018.

[10] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[12] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. J. Freedman, "Live video analytics at scale with approximation and delay-tolerance," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, 2017, pp. 377–392.

[13] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, "Chameleon: scalable adaptation of video analytics," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. ACM, 2018, pp. 253–266.

[14] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, and Q. Li, "Lavea: Latency-aware video analytics on edge computing platform," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. ACM, 2017, p. 15.

[15] C. Long, Y. Cao, T. Jiang, and Q. Zhang, "Edge computing framework for cooperative video processing in multimedia iot systems," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1126–1139, 2017.

[16] C.-C. Hung, G. Ananthanarayanan, P. Bodik, L. Golubchik, M. Yu, P. Bahl, and M. Philipose, "Videoedge: Processing camera streams using hierarchical clusters," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 115–131.

[17] A. Tiwari, B. Ramprasad, S. H. Mortazavi, M. Gabel, and E. d. Lara, "Reconfigurable streaming for the mobile edge," in *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '19. New York, NY, USA: ACM, 2019, pp. 153–158.

[18] M. C.-C. Hung and K. C.-J. Lin, "Joint sink deployment and association for multi-sink wireless camera networks," *Wireless Communications and Mobile Computing*, vol. 16, no. 2, pp. 209–222, 2016.

[19] C. Liang, Y. He, F. R. Yu, and N. Zhao, "Enhancing video rate adaptation with mobile edge computing and caching in software-defined mobile networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 7013–7026, 2018.

[20] N. K. Sharma and G. R. M. Reddy, "Multi-objective energy efficient virtual machines allocation at the cloud data center," *IEEE Transactions on Services Computing*, vol. 12, no. 1, pp. 158–171, 2016.

[21] N. C. Luong, P. Wang, D. Niyato, Y. Wen, and Z. Han, "Resource management in cloud networking using economic analysis and pricing models: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 954–1001, 2017.

[22] J. F. Nash Jr, "The bargaining problem," *Econometrica: Journal of the Econometric Society*, pp. 155–162, 1950.

[23] L. Gao, G. Iosifidis, J. Huang, L. Tassiulas, and D. Li, "Bargaining-based mobile data offloading," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1114–1125, 2014.

[24] C. A. Floudas, *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press, 1995.

[25] H. D. Sherali and W. P. Adams, *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*. Springer US, 1999, vol. 31.

[26] A. M. Geoffrion, "Generalized benders decomposition," *Journal of optimization theory and applications*, vol. 10, no. 4, pp. 237–260, 1972.

[27] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1755–1758, 2009.

[28] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–10.

[29] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B. C. Lovell, "Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition," in *IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops*. IEEE, June 2011, pp. 81–88.

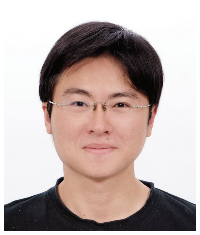
[30] D. Abbasinezhad-Mood, A. Ostad-Sharif, and M. Nikooghadam, "Novel anonymous key establishment protocol for isolated smart meters," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 2844–2851, 2019.

[31] I. Gurobi Optimization, "Gurobi optimizer reference manual," *URL http://www. gurobi. com*, 2015.

[32] J. Löfberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *Proceedings of the CACSD Conference*, vol. 3. Taipei, Taiwan, 2004.

Yi Zhang received the B.S. degree in software engineering from Software College, Xiamen University (XMU), China, in 2014. He received the M.S. degree from Graduate Institute of Communication Engineering (GICE), National Taiwan University (NTU), Taipei, Taiwan, in 2016. He has been an assistant engineer in Quanzhou Institute of Equipment Manufacturing, Haixi Institutes, Chinese Academy of Sciences (CAS), during 2016-2017. He is currently pursuing the Ph.D. degree in GICE at NTU. His primary research areas include wireless communication, fog computing, game theory and optimization theory.

Jiun Hao Liu received the B.S. and M.S. degrees from Department of Electrical Engineering and Graduate Institute of Electrical Engineering, National Taiwan University (NTU), Taipei, Taiwan, in 2017 and 2019, respectively. His primary research areas include wireless communication, cloud computing and edge computing.



Chih-Yu Wang received the B.S. and Ph.D. degrees in electrical engineering and communication engineering from National Taiwan University (NTU), Taipei, Taiwan, in 2007 and 2013, respectively. He has been a visiting student in University of Maryland, College Park in 2011. He joined Academia Sinica, Taipei, Taiwan in 2014. He is currently an Associate Research Fellow / Associate Professor in Research Center for Information Technology Innovation. His research interests include game theory, wireless communications, social networks, and data science.

He was a recipient of the K. T. Li Young Researcher Award from ACM Taipei/Taiwan Chapter and The Institute of Information and Computing Machinery in 2019, Ministry of Science and Technology Research Project for Excellent Young Scholars in 2019. His work was featured in 2018 and 2019 Significant Research Achievements of Academia Sinica.

Hung-Yu Wei is a Professor in Department of Electrical Engineering and Graduate Institute of Communications Engineering, National Taiwan University. Currently, he serves as Associate Chair in Department of Electrical Engineering. He received the B.S. degree in electrical engineering from National Taiwan University in 1999. He received the M.S. and the Ph.D. degree in electrical engineering from Columbia University in 2001 and 2005 respectively. He was a summer intern at Telcordia Applied Research in 2000 and 2001. He was with NEC Labs America from 2003 to 2005. He joined Department of Electrical Engineering at the National Taiwan University in July 2005. His research interests include next-generation wireless broadband networks, IoT, vehicular networking, fog/edge computing, cross-layer design for wireless multimedia, and game theoretical models for communications networks.

Dr. Wei received NTU Excellent Teaching Award in 2008 and 2018. He also received ”Recruiting Outstanding Young Scholar Award” from the Foundation for the Advancement of Outstanding Scholarship in 2006, K. T. Li Young Researcher Award from ACM Taipei/Taiwan Chapter and The Institute of Information and Computing Machinery in 2012, Excellent Young Engineer Award from the Chinese Institute of Electrical Engineering in 2014, Wu Ta You Memorial Award from MOSTin 2015, and Outstanding Research Award from MOST in 2020. He has been actively participating in NGMN, IEEE 802.16, 3GPP, IEEE P1934, and IEEE P1935 standardization. He serves as Vice Chair of IEEE P1934 Working Group to standardize fog computing and networking architecture. He serves as Secretary for IEEE Fog/Edge Industry Community. He also serves as an Associate Editor for IEEE IoT journal. He is an IEEE certified Wireless Communications Professional. He was the Chair of IEEE VTS Taipei Chapter during 2016 2017. He is currently the Chair of IEEE P1935 working group for edge/fog management and orchestration standard.