Optimization in Mobile Augmented Reality Systems for the Metaverse over Wireless Communications

Tianming Lan, Jun Zhao

School of Computer Science and Engineering

Nanyang Technological University, Singapore

tianming001@e.ntu.edu.sg, junzhao@ntu.edu.sg

Abstract—As the essential technical support for Metaverse, Mobile Augmented Reality (MAR) has attracted the attention of many researchers. MAR applications rely on real-time processing of visual and audio data, and thus those heavy workloads can quickly drain the battery of a mobile device. To address such problem, edge-based solutions have appeared for handling some tasks that require more computing power. However, such strategies introduce a new trade-off: reducing the network latency and overall energy consumption requires limiting the size of the data sent to the edge server, which, in turn, results in lower accuracy. In this paper, we design an edge-based MAR system and propose a mathematical model to describe it and analyze the trade-off between latency, accuracy, server resources allocation and energy consumption. Furthermore, an algorithm named LEAO is proposed to solve this problem. We evaluate the performance of the LEAO and other related algorithms across various simulation scenarios. The results demonstrate the superiority of the LEAO algorithm. Finally, our work provides insight into optimization problem in edge-based MAR system for Metaverse.

Index Terms—edge-based MAR system, resources allocation, non-convex optimization

I. INTRODUCTION

Metaverse has become more and more popular nowadays. And Augmented Reality (AR) is one of Metaverse's important technical supports [1], which can integrate the digital and physical elements [2]. Mobile Augmented Reality (MAR) can be more convenient than AR since people can use mobile devices to get access to the Metaverse everywhere.

To integrate MAR applications into the Metaverse, accurate and real-time recognition is crucial, because recognition failure and high latency will deteriorate the user experience [3]. However, on mobile devices, MAR applications, such as Target Recognition [4], are too energy-intensive to use. And limited computing resources will result in the long calculation latency. Hence, some research tried to offload image recognition tasks to the server that has strong computing power [5]–[7].

However, the use of servers, such as edge servers [8], [9], can result in long transmission delays and unstable networks, which may lead to poor user experience. Therefore, trade-offs become particularly important in such scenarios. Many researchers have begun to study the trade-off in this scenario which they call edge-based MAR systems [10], [11].

Motivation. On the one hand, when mobile devices send image recognition tasks to the edge server, high-resolution images could lead to high training accuracy, but low-resolution images could save transmission energy and latency. Therefore,

there will be a trade-off between energy, latency and accuracy. On the other hand, with the emergence of the Metaverse, the rapid growth of user and data volume will bring enormous pressure to edge servers. How to allocate the resources of multiple edge servers to all users in a reasonable manner while achieving the above trade-off is also an urgent and challenging issue to be addressed.

Challenges. This paper takes into account all the factors to the best of its ability, including latency, allocation of multi-server resources, accuracy, and user energy consumption. We also take the server energy consumption as one of the optimization goals because few research considered it. The aforementioned trade-off is formulated as a Mixed-Integer Non-convex Problem (MINP), which is generally known to be NP-hard. Such problems can not be easily solved with conventional optimization methods. This paper employs various mathematical techniques and proposes the Latency, Energy consumption, resources allocation and Accuracy comprehensive Optimization algorithm (LEAO) algorithm to address this issue.

After the comparison with other methods, we conclude that LEAO has a good performance in this scenario.

Our contributions are summarized as follows:

1) We find the trade-off problem between latency, accuracy, resources allocation and energy consumption in MAR scenario.We also take server energy consumption into account firstly. Finally, we propose an analytical model to formulate this problem.
2) We propose the LEAO algorithm to solve the trade-off with various mathematical techniques.
3) We design a MAR system based on our analytical model and we conduct experiments to verify the performance of LEAO in this system.

The organization of this paper is as follows: Section II will introduce the related work; Section III will describe the analytical model and how we formulate the problem; Section IV will introduce optimization algorithm detail; Section V will introduce our experimental methods and results; Section VI will summarize this paper.

II. RELATED WORK

The attempts at solving above-mentioned trade-offs start with cloud-based MAR systems [12]. Chen *et al.* proposed a real-time object recognition system [6]. This system can

improve accuracy and select appropriate image pixels automatically. Jain *et al.* proposed a method to reduce network latency [5]. Liu *et al.* proposed FACT algorithm to find the optimal point between accuracy and network latency [13]. But this work ignores the energy consumption in MAR system, which is a very crucial part. Wang *et al.* proposed LEAF algorithm to solve the trade-off problem between accuracy, energy consumption and latency [11]. However, this work only consider the limited scenario of one server. Although Ahn *et al.* proposed another algorithm to solve the abovementioned problem [14], it also only consider the one server scenario. Huang *et al.* considered delay and user location as optimization objectives, but did not consider energy consumption [15]. He *et al.* formulated an excellent model, but only accuracy is inside [16].

Furthermore, none of the works mentioned above consider the energy consumption of edge servers. Unlike cloud servers, edge servers face energy consumption issues as well.

III. PROBLEM FORMULATION

In this chapter, we will overview the edge-based MAR system of this paper in Fig. 1. And the next section describes analytical model which is used to formulate the problem.

A. System Overview

In Fig. 1, there are K mobile devices or users and N edge servers. Each mobile device will choose one and only one edge server to connect and off-load their computing tasks at one time. Throughout this paper, the n-th dimension of an N-dimensional vector x is denoted by x_n. We use an indicator $a_{k,n}$ to indicate the connection between mobile devices and servers. If $a_{k,n} = 1$, k-th mobile device connects with n-th server. We use matrix $\boldsymbol{A}$ to denote the set of variables $a_{k,n}$.

$$\boldsymbol{A} = [a_{k,n}|_{k\in\mathcal{K}, n\in\mathcal{N}}], \sum_{n\in\mathcal{N}} a_{k,n} = 1 \\ \mathcal{K} := \{1, ..., K\}, \mathcal{N} := \{1, ..., N\} \tag{1}$$

At the same time, the corresponding server will dynamically allocate the computing resources r_k to the k-th mobile device, $\boldsymbol{r} := \{r_1, ..., r_K\}$. The LEAO algorithm executes on the server, monitors the network information, and timely delivers the configuration to both mobile devices and servers.

In this paper, we analyze the entire image processing workflow: image generation, data transmission, image processing on servers, and accuracy evaluation. We take into account the system latency, as well as the energy consumption of both mobile devices and servers, and image recognition accuracy. We will formulate the above optimization objective in the following subsection.

B. Latency

Our latency model is constructed according to Equation (2), where L_k^t is the image transmission latency, L_k^{cn} is the core network latency and L_k^p is the processing latency.

$$L_k(s_k, a_{k,n}, r_k) = L_k^t(s_k) + L_k^{cn}(a_{k,n}) + L_k^p(s_k, r_k) \tag{2}$$

We denote the video frame resolution of the k-th mobile device as s_k, whose unit is pixel and σ is the number of bits in one pixel. Then, the transmission latency is modeled as Equation (3) and we have $\boldsymbol{s} := \{s_1, ..., s_K\}$.

$$L_k^t(s_k) = \sigma s_k / R_k \tag{3}$$

where R_k is the wireless data rate of the k-th mobile device.

And in Equation (4), $l_{k,n}$ is the core network latency between k-th mobile device and n-th server.

$$L_k^{cn}(a_{k,n}) = \sum_{n\in\mathcal{N}} a_{k,n} l_{k,n} \tag{4}$$

We denote the complexity of task of k-th mobile device by C_k. Then, the processing latency can be described by Equation (5), where we assume $C(s_k)$ is a convex function with respect to s_k.

$$L_k^p(s_k, r_k) = C(s_k)/r_k \tag{5}$$

C. Accuracy

In our system, we regard the image recognition accuracy as one of our optimization goal because it is directly related to user experience. We assume that the accuracy $A_k(s_k)$ is a concave function of video frame resolution s_k.

D. Energy Consumption

Our mobile device energy consumption model is shown in Equation (6), where E_k^{img} is the image generation and preview energy consumption, E_k^{com} is the wireless communication energy consumption, E_k^{bs} is the base energy consumption of mobile device. We use the sum of them to denote the total energy consumption of the k-th mobile device:

$$E_k(f_k, s_k, a_{k,n}, r_k) = E_k^{img} + E_k^{com} + E_k^{bs} \tag{6}$$

We denote the CPU frequency of k-th mobile device by f_k. In general, the most significant proportion of energy consumption is often E_k^{img}, which is the the product of delay and power:

$$E_k^{img}(f_k) = t_{pre} P^{pre}(f_k) \tag{7}$$

where t_{pre} is the time to pre-process an image and is assumed to be a constant. $P^{pre}(f_k)$ is the power of pre-processing an image and we suppose that it is convex with respect to f_k. Finally, we set E_k^{com} and E_k^{bs} as following equation.

$$E_k^{com}(s_k) = P^{tr}(R_k) L_k^t(s_k) \tag{8}$$

$$E_k^{bs}(f_k, s_k, a_{k,n}, r_k) = P^{bs}(f_k) L_k(s_k, a_{k,n}, r_k) \tag{9}$$

where $P^{tr}(R_k)$ is the transmission power of k-th mobile device and $P^{bs}(f_k)$ is the basic power of mobile device. We assume that $P^{bs}(f_k)$ is a convex function with respect to f_k. Furthermore, the server energy consumption is modeled as Equation (10).

$$E_n(a_{k,n}, s_k, r_k) = \sum_{k\in\mathcal{K}} a_{k,n} P_n(\frac{r_k}{S_n} F_n) L_k^p(s_k, r_k) \tag{10}$$

In this equation, F_n is the CPU frequency of n-th server and P_n is the power of n-th server. For each server n, only

Fig. 1. System Overview.

the computing resources that are used will generate energy consumption and S_n stands for the total available resources of n-th server. Finally, the argument of the P_n function is obtained by multiplying the proportion of computing resources used by each server r_k/S_n and the CPU frequency F_n.

E. Optimization problem

In this paper, we aim to minimize the overall energy consumption, latency and maximize the accuracy of each mobile device. This is a multi-objective optimization problem and we adopt the weighted sum method to express the objective function Q given in the Equation (11). Furthermore, to express the trade-off between different objectives, we introduce parameter λ_1^k and λ_2^k to reflect the preference between them. For example, a larger λ_1^k indicates that the system prefers a lower latency and a larger λ_2^k indicates that the system prefers the higher accuracy.

$$Q(f_k, s_k, a_{k,n}, r_k) = \sum_{n \in \mathcal{N}} \frac{E_n}{N} + \sum_{k \in \mathcal{K}} \frac{E_k + \lambda_1^k L_k - \lambda_2^k A_k}{K} \quad (11)$$

Besides, the optimization problem is shown in equation (12).

$$\mathbb{P}_0 : \min_{\{f, s, A, r\}} Q \quad (12)$$

$$s.t. \quad C_1 : A_k(s_k) \geq A_{min}, \forall k \in \mathcal{K}; \quad (12a)$$

$$C_2 : L^k(s_k, a_{k,n}, r_k) \leq L_{max}^k, \forall k \in \mathcal{K}; \quad (12b)$$

$$C_3 : f_{min} \leq f_k \leq f_{max}, \forall k \in \mathcal{K}; \quad (12c)$$

$$C_4 : s_{min} \leq s_k \leq s_{max}, \forall k \in \mathcal{K}; \quad (12d)$$

$$C_5 : a_{k,n} \in 0, 1, \forall k \in \mathcal{K}, n \in \mathcal{N}; \quad (12e)$$

$$C_6 : \sum_{n \in \mathcal{N}} a_{k,n} = 1, \forall k \in \mathcal{K}; \quad (12f)$$

$$C_7 : \sum_{k \in \mathcal{K}} a_{k,n} r_k \leq S_n, \forall n \in \mathcal{N}; \quad (12g)$$

where A_{min} is the minimum accuracy requirement of mobile device; L_{max}^k is the upper bound for the k-th mobile device latency. Constraints (12c) and (12d) are the constraints of the mobile device's CPU frequency and pixels of input images. (12e) and (12f) ensure that an mobile device only can choose one edge server. (12g) ensure that the total computing

resources allocated to mobile devices connected to n-th server do not exceed the computing resources of n-th server S_n.

$\mathbb{P}_0$ is MINP which is a NP-hard problem. To solve such an intractable problem, we analyze the properties of $\mathbb{P}_0$ firstly. In the objective function Q, the four variables f, s, A and r are all multiplying to each other. Therefore, it is obvious that $\mathbb{P}_0$ is not jointly convex with respect to $[f, s, A, r]$. Similarly, it is easy to prove that $\mathbb{P}_0$ is strictly convex with respect to f, s, and r separately since the second-order derivative of Q with respect to each of them are greater than zero. Due to the simplicity of this proof and the limitation of the length, a detailed description will not be provided here. Although $\mathbb{P}_0$ is not jointly convex with respect to $[f, s, A, r]$, we can use other problems to approximate the solution of $\mathbb{P}_0$, which will be explained in Section IV.

IV. OPTIMIZATION ALGORITHM

The first difficulty in solving the problem $\mathbb{P}_0$ arises from the discrete variable A. We relax the discrete variable $a_{k,n}$ into continuous variable $\hat{a}_{k,n}$, $\hat{A} = [\hat{a}_{k,n}|_{k \in \mathcal{K}, n \in \mathcal{N}}]$. By changing constraints (12e) and (12f) of $\mathbb{P}_0$ into (13a), (13b) and (13c) of $\mathbb{P}_1$, we transform $\mathbb{P}_0$ into an equivalent $\mathbb{P}_1$ since variable $\hat{a}_{k,n}$ also only can be 0 or 1.

$$\mathbb{P}_1 : \min_{\{f, s, \hat{A}, r\}} Q \quad (13)$$

$$s.t. \quad (12a), (12b), (12c), (12d);$$

$$C_5 : 0 \leq \hat{a}_{k,n} \leq 1, \forall n \in \mathcal{N}, \forall k \in \mathcal{K}; \quad (13a)$$

$$C_6 : \sum_{k \in \mathcal{K}} \sum_{n \in \mathcal{N}} \hat{a}_{k,n}(1 - \hat{a}_{k,n}) \leq 0; \quad (13b)$$

$$C_7 : \sum_{n \in \mathcal{N}} \hat{a}_{k,n} = 1, \forall k \in \mathcal{K}; \quad (13c)$$

$$C_8 : \sum_{k \in \mathcal{K}} \hat{a}_{k,n} r_k \leq S_n, \forall n \in \mathcal{N}; \quad (13d)$$

Since we only modify the constraints related to A, $\mathbb{P}_1$ is also strictly convex with respect to f, s and r and all the variables in $\mathbb{P}_1$ are continuous. However, it is still difficult to solve for the nonconvex part. The constraint (13b) in $\mathbb{P}_1$ is concave so that $\mathbb{P}_1$ is not convex with respect to $\hat{A}$, and $\mathbb{P}_1$ is also not jointly convex with respect to $[f, s, \hat{A}, r]$. Within this section, a systematic algorithm will be formulated step by step to address and resolve the problem $\mathbb{P}_1$.

A. Successive Convex Approximation (SCA) Algorithm

We plan to use SCA to solve the nonconvex part. However, to facilitate the solution, we need to penalize the concave constraint in (13b) to the objective function before we use SCA, which is shown in (14).

$$\mathbb{P}_2 : \min_{\{\boldsymbol{f},\boldsymbol{s},\hat{\boldsymbol{A}},\boldsymbol{r}\}} Q - \mu \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}} \hat{a}_{k,n}(\hat{a}_{k,n}-1) \qquad (14)$$

$$s.t. \quad (12a), (12b), (12c), (12d), (13a), (13c), (13d);$$

μ in $\mathbb{P}_2$ is the penalty parameter and we have $\mu \geq 0$. Denote with $\alpha(\mu)$ the optimal objective value. Based on Theorem 1 of [17], we show the equivalence of $\mathbb{P}_1$ and $\mathbb{P}_2$ in the following lemma.

Lemma 1. *(Exact Penalty) For all $\mu \geq \mu_0$ where*

$$\mu_0 = \frac{Q(f_k, s_k, a_{k,n}^0, r_k) - \alpha(0)}{max_{\hat{\boldsymbol{A}}}\{\hat{a}_{k,n}(\hat{a}_{k,n}-1) : (13a), (13b), (13c)\}} \qquad (15)$$

With any $a_{k,n}^0, \forall n \in \mathcal{N}, \forall k \in \mathcal{K}$ satisfying constraints (13a), (13b) and (13c), $\mathbb{P}_1$ and $\mathbb{P}_2$ have the same optimal solution.

Proof. Theorem 1 in [17] has proved this lemma. And the values of f_k, s_k and r_k are from last iteration. $\square$

SCA involves iteratively solving a sequence of convex problems. In each iteration, we use a surrogate convex function to approximate the non-convex function. Specifically, we approximate $\mathbb{P}_2$ with $\mathbb{O}_t$ which is the problem in t-th iteration shown in following equation. In $\mathbb{O}_t$, we approximate constraints (13b) with $\hat{a}_{k,n}^{(t)}(\hat{a}_{k,n}^{(t)}-1)+(2\hat{a}_{k,n}^{(t)}-1)(\hat{a}_{k,n}-\hat{a}_{k,n}^{(t)})$, and $t =: \{0, 1, ...\}$. At the end of one iteration, we get the value of $\hat{a}_{k,n}$ and we regard it as the $\hat{a}_{k,n}^{(t+1)}$ in next iteration. This part is as shown in Algorithm 1.

$$\mathbb{O}_t : \min_{\{\boldsymbol{f},\boldsymbol{s},\hat{\boldsymbol{A}},\boldsymbol{r}\}} Q - \mu \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}} (\hat{a}_{k,n}^{(t)}(\hat{a}_{k,n}^{(t)}-1)+$$
$$(2\hat{a}_{k,n}^{(t)}-1)(\hat{a}_{k,n}-\hat{a}_{k,n}^{(t)})) \qquad (16)$$

$$s.t. \quad (12a), (12b), (12c), (12d), (13a), (13c), (13d);$$

Algorithm 1 SCA

Input: optimization variable $\boldsymbol{x}$, objective function $O(\boldsymbol{x}|\boldsymbol{x}^{(t)})$, constraints set $\boldsymbol{C}$ and convergence condition τ
Output: optimal solution $\boldsymbol{x}^*$
 1: Initialization: Find an initial feasible point $\boldsymbol{x}^{(0)}$ of $\mathbb{P}_2$ and set t = 0.
 2: **for** iteration $t = 0, 1, ..$ **do**
 3: $\quad \boldsymbol{x}^{(t+1)} \leftarrow$ solve $O(\boldsymbol{x}|\boldsymbol{x}^{(t)})$ satisfying $\boldsymbol{C}$
 4: $\quad$ **if** $|(\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)})/\boldsymbol{x}^{(t)}| \leq \tau$ **then**
 5: $\quad\quad$ **break;**
 6: $\quad$ **end if**
 7: $\quad t \leftarrow t + 1$
 8: **end for**
 9: $\boldsymbol{x}^* \leftarrow \boldsymbol{x}^{(t)}$

From $\mathbb{P}_2$ to $\mathbb{O}_t$, we didn't change any constraints or other part of objective function. Therefore, the problem $\mathbb{O}_t$ is strictly convex with respect to $\boldsymbol{f}$, $\boldsymbol{s}$, and $\boldsymbol{r}$ separately. Furthermore, it's easy to prove that the problem $\mathbb{O}_t$ is also strictly convex with respect to $\hat{\boldsymbol{A}}$ due to the linearity of problem $\mathbb{O}_t$ with respect to $\hat{\boldsymbol{A}}$. Due to the limitation of the length, a detailed description will not be provided here.

However, at this stage, the problem is still not completely solved, as the problem $\mathbb{O}_t$ is not jointly convex in all variables, because there are many products of variables $\boldsymbol{f}, \boldsymbol{s}, \hat{\boldsymbol{A}}$, and $\boldsymbol{r}$ in the objective function and the constraints. We next introduce new variables and use Product Replacement to further address this problem.

B. Product Replacement Algorithm

To address the non-joint convexity of the problem $\mathbb{O}_t$ with respect to $[\boldsymbol{f}, \boldsymbol{s}, \hat{\boldsymbol{A}}, \boldsymbol{r}]$, we introduce new variables $\boldsymbol{w} := \{w_1, ..., w_K\}$ and $\boldsymbol{z} := \{z_1, ..., z_N\}$ to separate the product of $\boldsymbol{s}$ and $\boldsymbol{r}$, as well as $\hat{\boldsymbol{A}}$ and $\boldsymbol{r}$ according to the algorithm in [18].

The Section IV in [18] has proved that the objective function xy has the same KKT solution as $x^2w + y^2/4w$ when $w = y/2x$. Then we apply this method in formula (5) and we get new processing delay $\widetilde{L}_k^p(s_k, r_k, w_k)$. The new objective function is denoted by $\widetilde{Q}$.

$$\widetilde{L}_k^p(s_k, r_k, w_k) = C^2(s_k)w_k + \frac{1}{4w_k r_k^2} \qquad (17)$$

At the same time, this method also can be applied in constraint (13d), then we get the new constraint (18a) in following problem.

$$\mathbb{H}_t : \min_{\{\boldsymbol{f},\boldsymbol{s},\hat{\boldsymbol{A}},\boldsymbol{r},\boldsymbol{w},\boldsymbol{z}\}} \widetilde{Q} \qquad (18)$$

$$s.t. \quad (12a), (12b), (12c), (12d), (13a), (13c);$$

$$C_7 : \sum_{k\in\mathcal{K}} (\hat{a}_{k,n}^2 z_n + \frac{r_k^2}{4z_n}) \leq S_n, \forall n \in \mathcal{N}; \qquad (18a)$$

Via the proof in the Section IV in [18], we can draw the conclusion that the problem $\mathbb{H}_t$ and the problem $\mathbb{O}_t$ have the same KKT solution for variables $[\boldsymbol{f}, \boldsymbol{s}, \hat{\boldsymbol{A}}, \boldsymbol{r}]$.

However, the product of f_k and $\hat{a}_{k,n}$, s_k and r_k still exists in the objective function of $\mathbb{H}_t$. Introducing new variables to split these products would lead to a significant expansion of the variable space. To avoid the problem becoming overly complex, we treat $\boldsymbol{f}$ as a constant and optimize only $[\boldsymbol{s}, \hat{\boldsymbol{A}}, \boldsymbol{r}, \boldsymbol{w}, \boldsymbol{z}]$. This process is presented in Algorithm 2 where some operation symbols are from MATLAB.

C. LEAO

In this subsection, we use Block Coordinate Descent (BCD) to optimize $\boldsymbol{f}$ which was kept constant in last subsection. Firstly, we fix the $\boldsymbol{f}$ and optimize $[\boldsymbol{s}, \hat{\boldsymbol{A}}, \boldsymbol{r}, \boldsymbol{w}, \boldsymbol{z}]$ with Algorithm 1 and Algorithm 2. Secondly, we fix $[\boldsymbol{s}, \hat{\boldsymbol{A}}, \boldsymbol{r}, \boldsymbol{w}, \boldsymbol{z}]$ and optimize $\boldsymbol{f}$ using the KKT condition.

Combining all the above lemma and algorithms, we are finally able to design the algorithm LEAO, which is shown in the Algorithm 3. In the first step, we employ BCD to optimize $\boldsymbol{f}$ separately from the other variables. Then, we use SCA to

---

Algorithm 2 Product Replacement

Input: Problem $\mathbb{H}_t$ and convergence condition τ
Output: optimal solution $[s, \hat{A}, r, w, z]^*$
1: Initialization: Find an initial point of $[s, \hat{A}, r, w, z]$.
2: **for** iteration $i = 0, 1, ..$ **do**
3:    $[s, \hat{A}, r]^{(i)} \leftarrow$ use Algorithm 1 to solve $\mathbb{H}_t$ with fixed $[f, w, z]$
4:    $w^{(i)} = 1./(2C(s^{(i)}).* r^{(i)})$
5:    $z^{(i)} = sum(r^{(i)})./2sum(\hat{A}^{(i)}, 1)$
6:    **if** $|(\widetilde{Q}^{(i+1)} - \widetilde{Q}^{(i)})/\widetilde{Q}^{(i)}| \leq \tau$ **then**
7:      **break;**
8:    **end if**
9:    $i \leftarrow i + 1$
10: **end for**
11: $[s, \hat{A}, r, w, z]^* \leftarrow [s, \hat{A}, r, w, z]^{(i)}$

---

ensure that the optimization problem is convex with respect to $\hat{A}$. Finally, we introduce new variables to separate the product of s and r, as well as $\hat{A}$ and r, such that the problem becomes jointly convex with respect to $[s, \hat{A}, r]$.

Algorithm 3 LEAO

Input: problem $\mathbb{P}_1$, problem $\mathbb{H}_t$, convergence condition τ
Output: $[f, s, \hat{A}, r]^*$
1: $\hat{A} \leftarrow A$
2: **for** iteration $i = 1, 2, ..$ **do**
3:    $f^{(i)} \leftarrow$ use KKT condition to solve $\mathbb{P}_1$ with fixed $[s, \hat{A}, r]$
4:    $[s, \hat{A}, r]^{(i)} \leftarrow$ use Algorithm 2 to solve $\mathbb{H}_t$ with fixed f
5:    **if** $|(Q^{(i+1)} - Q^{(i)})/Q^{(i)}| \leq \tau$ **then**
6:      **break;**
7:    **end if**
8:    $i \leftarrow i + 1$
9: **end for**
10: $[f, s, \hat{A}, r]^* \leftarrow [f, s, \hat{A}, r]^{(i)}$

D. Convergence and Time Complexity Analysis

The convergence proof of SCA, Product Replacement and BCD can be found in [18]–[20]. In view of the fact that the main body of Algorithm LEAO is composed of these three algorithms, its convergence is guaranteed as well.

Since the objective function $\widetilde{Q}$ is jointly convex with respect to $[s, \hat{A}, r]$, we can solve it by KKT condition. Therefore, we assume the time complexity of this process is $O(1)$. Since the size of the variable space is $3K + N + KN$, the time complexity of solving this problem is $O(KN)$. Assuming the loop counts of the three algorithms mentioned above are ζ, η, θ, the time complexity of the final algorithm LEAO is $O(\zeta\eta\theta KN)$.

V. PERFORMANCE EVALUATION

In our simulation, the default configuration is 100 users and 10 servers and we generate $l_{k,n}$ randomly. Different

parameters default setting are showed in Table I. Besides, the parameter μ in Lemma 1 is set by experience.

TABLE I
PARAMETERS SETTING

Parameters	Value	Parameters	Value
τ	0.001	λ_1^k	$0 \sim 50$
σ	8 bits	λ_2^k	$0 \sim 1000$
S_n	$8 \sim 12$ TFLOPS	f_{min}	2.2 GHz
F_n	4.4GHz $\sim$ 4.6GHz	f_{max}	3.5 GHz
t_{pre}	0.4 ms	s_{min}	256^2 pixels
A_{min}	0.6	s_{max}	1024^2 pixels
$l_{k,n}$	100ms $\sim$ 130ms	L_{max}	250 ms

Based on the measurements of other works [11], [13], a list of specific function used in this paper is shown in Table II. Although the function $P^{pre}(f_k)$ is not a convex function, it's convex when f_k is in the range of Table I.

TABLE II
PROPOSED MODELS

Functions	Models
$C(s_k)$	$7 \times 10^{-10}s_k^{1.5} + 0.083$ TFLOPS
$A(s_k)$	$1 - 1.578e^{-6.5 \times 10^{-3}\sqrt{s_k}}$
$P^{tr}(R_k)$	$0.018R_k + 0.7$
$P^{bs}(f_k)$	$0.079f_k + 0.59$
$P^{pre}(f_k)$	$-0.01071f_k^3 + 0.06055f_k^2 - 0.1028f_k + 0.107$
$P_n(x)$	$0.083x^2 + 0.32$

Furthermore, in this paper, we compare LEAO with three algorithms.

- **Baseline**: The baseline algorithm has fixed variable values and we generate those values randomly by the same Gaussian random seed and the variable range is shown in Table I.
- **User Workload Optimized (UWO)**: This algorithm optimize mobile device's CPU frequency f and image resolution s. All other variable are fixed.
- **Resources Allocation Optimized (RAO)**: This algorithm optimize the resources allocation of mobile devices r and the connection map A. And other variable are fixed.

Optimality. Firstly, we compare the objective function value between different algorithm and the result is shown in Fig. 2. In one word, LEAO is almost the lowest one, independently of the parameter configuration. Fig. 2 (a) shows the relationship between Q and λ_2/λ_1. Since accuracy A_k is much smaller number than latency L_k, we only increase the λ_2 and Q decreases with it. Besides, the RAO is better than UWO, which means optimizing the resources allocation is more important than user workload.

Impact of λ. In Fig. 2 (b), the accuracy increases while the ratio λ_2/λ_1 increasing. Because the larger λ_2 will make the system pay more attention to the accuracy optimization. Vice versa, the system will focus more on the latency and energy consumption optimization while the λ_2 is a small number. Simultaneously, with the alteration of the value of the ratio λ_2/λ_1, the objective function associated with the optimal solution exhibits minimal variation, thereby demonstrating the stability of our optimization algorithm's performance.

(a) Optimality vs. λ_2/λ_1

(b) Accuracy vs. λ_2/λ_1

(c) Optimality vs. K

(d) Optimality vs. N

Fig. 2. The influence of different parameters on the experimental results.

Impact of K and N. In Fig. 2 (c), the objection function of LEAO keeps stable and even decreases as K increases indicating that our algorithm can adapt well to high user volume scenarios. Moreover, the fact that LEAO outperforms RAO implies that, as the number of users increases, optimizing only the resource allocation is not sufficient, and optimizing the users workload is equally important. And in Fig. 2 (d), the objective function value also slightly decreases as N increases, which indicates that the resource allocation optimization problem with multiple servers is still within the scope of our algorithm's capability.

VI. Conclusion

In this paper, we proposed an edge-based MAR system for Metaverse, which can reduce energy consumption and optimize resources allocation while maintaining high accuracy at the same time. Besides, we built a complete mathematical model to analyze the trade-off between latency, accuracy, energy consumption and resources allocation in edge-based MAR system. On this basis, we developed the LEAO algorithm to improve system performance and user experience by synchronously optimizing mobile device's CPU frequency, frame resolution, server assignment and server resources allocation. Finally, we evaluated the performance of LEAO algorithm by simulation and demonstrated its ability to achieve good experimental results.

Acknowledgement

References

[1] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev, *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.

[2] T. Tuan-Kiet, H.-T. HUYNH, D.-P. NGUYEN, L. Dinh-Dung, T. Thi-Hong, and Y. NAKASHIMA, "Demonstration of a visible light receiver using rolling-shutter smartphone camera," in *2018 International Conference on Advanced Technologies for Communications (ATC)*. IEEE, 2018, pp. 214–219.

[3] L. H. Lee, T. Braud, F. H. Bijarbooneh, and P. Hui, "Ubipoint: towards non-intrusive mid-air interaction for hardware constrained smart glasses," in *Proceedings of the 11th ACM Multimedia Systems Conference*, 2020, pp. 190–201.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[5] P. Jain, J. Manweiler, and R. Roy Choudhury, "Low bandwidth offload for mobile ar," in *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, 2016, pp. 237–251.

[6] T. Y.-H. Chen, L. Ravindranath, S. Deng, P. Bahl, and H. Balakrishnan, "Glimpse: Continuous, real-time object recognition on mobile devices," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015, pp. 155–168.

[7] R. Shea, A. Sun, S. Fu, and J. Liu, "Towards fully offloaded cloud-based ar: Design, implementation and experience," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 321–330.

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[9] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[10] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *The 25th annual international conference on mobile computing and networking*, 2019, pp. 1–16.

[11] H. Wang and J. Xie, "User preference based energy-aware mobile ar system with edge computing," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1379–1388.

[12] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui, "Mobile augmented reality survey: From where we are to where we go," *Ieee Access*, vol. 5, pp. 6917–6950, 2017.

[13] Q. Liu, S. Huang, J. Opadere, and T. Han, "An edge network orchestrator for mobile augmented reality," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 756–764.

[14] J. Ahn, J. Lee, D. Niyato, and H.-S. Park, "Novel qos-guaranteed orchestration scheme for energy-efficient mobile augmented reality applications in multi-access edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 631–13 645, 2020.

[15] Z. Huang and V. Friderikos, "Proactive edge cloud optimization for mobile augmented reality applications," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2021, pp. 1–6.

[16] Y. He, J. Ren, G. Yu, and Y. Cai, "Optimizing the learning performance in mobile augmented reality systems with cnn," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5333–5344, 2020.

[17] H. A. Le Thi, T. Pham Dinh, and H. V. Ngai, "Exact penalty and error bounds in dc programming," *Journal of Global Optimization*, vol. 52, no. 3, pp. 509–535, 2012.

[18] J. Zhao, L. Qian, and W. Yu, "Human-centric resource allocation in the metaverse over wireless communications," *arXiv preprint arXiv:2304.00355*, 2023.

[19] M. Razaviyayn, "Successive convex approximation: Analysis and applications," Ph.D. dissertation, University of Minnesota, 2014.

[20] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.