Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

DATA-DRIVEN UNCERTAINTY REVENUE MODELING FOR COMPUTATION RESOURCE ALLOCATION IN RECOMMENDATION SYSTEMS

Feixue Liu[1,2], Yuqing Miao[1,2], Yun Ye[3], Peisong Wang[1,2], Xinglu Liu[1,2], Kai Zhang[2], and Wai Kin (Victor) Chan[1,2]

[1]Tsinghua-Berkeley Shenzhen Institute, Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, P.R. CHINA
[2]Institute of Data and Information, Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, P.R. CHINA
[3]Ant Group, Shanghai, P.R. CHINA

ABSTRACT

In recent years, as the resource consumption of computation-intensive recommendation systems (RS) significantly increased, and the supply of large-scale resources encountered bottleneck, computation resource allocation for improving computing efficiency grabbed the attention of the industry. The simulation of revenue is a focal point in the allocation problem. However, due to the complex engineering architecture of RS, no existing research proposes a simulation model that addresses the relationship between resource allocation strategies and benefits. This paper, based on real data from Alipay's advertising RS and integrating queuing theory, models the relationship from resource allocation decisions to revenue considering exposure randomness of traffic. We further merge allocation tasks with capacity planning (CP) to establish a two-stage joint optimization model and use the revenue model above as the objective. The proposed model outperforms the baseline with 1.9% in revenue, and represents flexible adaptation of exposure rate, providing insights for the simulation of industrial RS.

1 INTRODUCTION

In recent years, with the continuous growth of internet users, the computation resource (CR) consumption of highly AI-dependent and computationally intensive RS significantly increased. At the same time, due to the impact of macroeconomic and environmental factors, the supply of large-scale CR encounters bottlenecks, highlighting the importance of improving computation efficiency in the traffic control domain of RS. Therefore, more precise advertising revenue modeling and computation resource allocation (CRA) have become focal issues in the Internet industry. Current approaches to depicting revenue models in RS are relatively naive and do not consider the system's random factors. Therefore, there's an urgent need for exploration into optimization of revenue modeling and CRA.

In the present paper, we refer to the architecture of Alipay's advertising recommendation system, consider the randomness of exposure in the recommendation system, and based on real data, analyze the impact of decisions on system state indicators such as Response Time (RT) and system exposure rate. This allows us to construct a path from CRA decisions to total system revenue, thereby optimizing and making decisions on the system's CP and CRA tasks. Additionally, we compare our method with industrial approaches under the same conditions, demonstrating the superiority of our method.

The paper is organized as follows. In the next section, we will discuss related work. The problem description and modeling of revenue will be introduced in Section 3. Two-stage CP and CRA model will be established in Section 4. Numerical experiments and results will be shown in Section 5 and conclusions will be discussed in Section 6.

2 LITERATURE REVIEW

CRA in recommendation systems is a new research field. The characteristics of RS, such as huge user traffic flow and strong dynamics, lead to challenges in modeling. This section introduces the relevant work on traffic models, current strategies for RS computation allocation, and the existing RS computation allocation simulator of Alipay, which lays the foundation for our proposed model combining traffic flow arrival and CR capacities.

With the development of the internet industry, data traffic has become a major factor influencing the internet economy. Traffic modeling and simulation research, to our knowledge, primarily originated from scholars analyzing network traffic in the telecommunications field. Faced with network service quality, traffic modeling provides a guaranteed performance to user data flows (Chandrasekaran 2009). Similarly, in RS, we face analogous issues: traffic modeling is essential to ensure effective recommendations, impacting exposure rates, which is the ratio of the exposed traffic to all traffic and revenue.

Traditional methods of modeling traffic are mainly based on discrete event simulation. According to traffic characteristics, Poisson, Pareto, Weibull, and Markov process distributions are general choices (Neame 2003; Ntlangu and Baghai-Wadji 2017). Markov processes are tractable but hardly adapt to high-speed networks. Pareto-based traffic models are suitable for high-speed data transmission networks because of long-term correlation (Adas 1997). In internet traffic, the arrival of requests from users to internet servers can be modeled by Poisson process, and the service process by exponential process (Alakiri et al. 2014; Olaniran and Abdullah 2020; Fras et al. 2013). Characterizing internet traffic by Poisson process has advantages: a)Well-suited for high-speed event analysis (Sapegin et al. 2015), and b) Common in traffic applications composed of a large number of independent flows due to Palm's Theorem (Chandrasekaran 2009).

Next, we turn to the current research on the CR allocation problem. DCAF proposed by Alibaba (Jiang et al. 2020), concentrates on a singular phase of CR allocation and was the pioneering approach to frame computation allocation problems as a multiple-choice knapsack problem (MCKP) (Kellerer et al. 2004). However, they merely considered RT as an evaluation metric for model correction. It was not until 2022 that Alibaba first raised the issue of RT control, which considers the regulation of RT at different stages, adopting a method based on manual feedback to adjust the RT of each stage. They latterly combined response time into the CRAS allocation model (Yang et al. 2021), as shown in (1), but merged the decision-making across the entire chain with the overall response time constraints. This model sets the objective as maximizing revenue and subjects to the computation capacity C and time limit RT. Subsequent studies, such as those on cascade RS in multi-phase settings like RL-MPCA (Zhou et al. 2023) and Greenflow (Lu et al. 2023), continue to operate within the DCAF framework, employing similar resolution methods and assumptions. Although Greenflow is the most advanced method to build CRA strategies, it does not consider the system uncertainty factor.

$$\text{Max}\left\{\sum_{jk} Q_j^k x_j^k \,\middle|\, \text{s.t.}\sum_{jk} q_j^k x_j^k \le C, \sum_j rt_j^k x_j^k \le RT^k, \sum_j x_j^k = 1, x_j^k \in \{0,1\} \,\middle|\, \forall k \in \mathscr{K}\right\} \tag{1}$$

While proposing optimization models, the industry has also explored the essence and various aspects of computational capability in traffic control. For example, it has been proposed that the essence of computational capability is the product of physical computing resources and time that CPU and QPS(Queries per second) have a proportional relationship. The Denghuo team analyzed and found a positive correlation between RT and queue length.

In summary, the current literature predominantly focuses on constructing revenue models from the perspective of advertising mechanisms rather than from the standpoint of system traffic control. They typically handle RT by setting RT thresholds based on regulatory experience and then treating it as a resource constraint or by directly using the timeout rate as an evaluation metric for direct system regulation. These methods of considering RT are often too rigid and lack a deep consideration of the impact of

response time on revenue. Inspired by multi-objective joint optimization problems in cloud computing that attempt to control both system resource utilization and RT minimization, a more nuanced approach to RT minimization in recommendation systems needs to be better considered. Additionally, in resource allocation problems, current literature mostly focuses on maximizing resource utilization rates without considering the value of the utilization rate itself, i.e., the size of the system load and its chain reactions on other aspects. Particularly in the recommendation systems domain, excessive system load can lead to other issues, ultimately affecting platform revenue. Therefore, in offline resource allocation tasks, there is a need for a capacity planning and allocation model that considers the overall resources, allocation strategies, and the RT to the final revenue, providing comprehensive control over system load, reliability, utilization, and traffic computation capability budget allocation decisions.

3 PROBLEM STATEMENT AND MODELING OF REVENUE

This research approaches computational systems in recommendation system engineering and operational platforms by simulating and modeling their profitability and planning for capacity and computational allocation. Initially, for CP, considering the CR in RS is comprised of CPUs, we need to pre-order the number of CPU cores, where each CPU's computational capability is C. We pre-order s CPUs at a cost of $C_{opportunity}$ per CPU, ultimately acquiring a total CR of $C * s$. Then, based on our pre-ordered total CR, we address the computation allocation issue. Since the recommendation system processes numerous visitation requests for advertisement recommendations daily, the computation system must allocate computation budgets for recommending products or ads to users of varying value, assigning more CR to higher-value users to enhance revenue. We assume there are multiple advertisement scenes k, with M^k requests $i = 1, ..., M^k$ requesting in scene k within a time frame. For each request, N actions $j = 1, ..., N$ can be taken, and x_{ij}^k represents whether action j is chosen by request i in scene k, where actions correspond to the input set of candidate ads for the advertising model or the queue length of recommended products. Under conventional estimation algorithms, when computational resources are ample, a longer queue implies greater computational consumption and higher revenue. We define R_{ij}^k and c_{ij}^k as the gain for request i assigned to action j in scene k and the cost of the request, respectively. It's important to highlight that here, the gain is inferred through an external recommendation algorithm module and does not consider real revenue after accounting for system engineering factors, thus introducing a deviation in estimation.

Our model's objective function aims to maximize the actual revenue of the advertising platform, where the actual revenue is the difference between revenue and opportunity costs. Actual revenue is considered to be a discounted portion of the predicted revenue, and this discount is attributable to factors related to the advertisement exposure mechanism of the recommendation system. Advertisement exposure refers to a mechanism where only the ads that are displayed to users can generate final revenue, which is a stochastic element influenced by both system factors, such as network speed and system load, and user factors, like a user's willingness to view an ad, which is difficult to predict. Consequently, our focus is on the quantifiable aspects from the platform's perspective, namely, the system factors. The determination of whether an ad is exposed thus becomes a binomially distributed random variable influenced by response time, system load, and our resource allocation scheme. Therefore, in this paper, we primarily analyze the decisions and the system data metrics influenced by these decisions. Below, we first analyze and model the relationships within the chain based on industrial data, and then we present the revenue chain from decisions to final revenue.

3.1 The Relationship Among Actions, Computational Resource Consumption and System Load

One of the most crucial elements in computational resource allocation is cost, which is the resource consumed after a request is decided upon an action. In the current system, this element is represented by the number of floating-point operations. By estimating the connection between computational resource

consumption and the actions, we can predict the future computational resource consumption for selecting a certain action for a request, thus allowing for better allocation.

We grouped all request data according to different queue lengths and calculated the average computational resource consumption for each scene. The relationship graph, as shown in Figure 1, indicates that the average value of computational resource consumption is directly proportional to queue length for different media scenes.

After that, we will discuss the system load. We consider our CPU as a single-core CPU, where CPU utilization is viewed as the percentage of the current CR used out of the total CR available to that single-core CPU. Correspondingly, the overall computation consumption divided by the computational capability of a single-core CPU, multiplied by the number of cores, will give us the average CPU utilization.

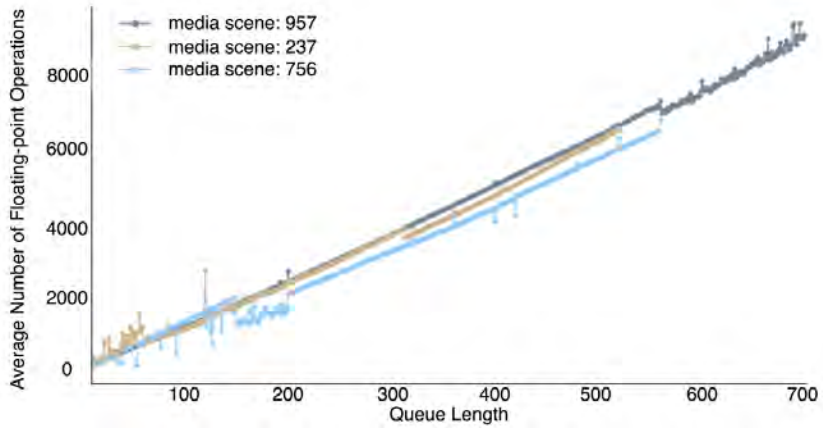

Figure 1: Relationship between queue length and average number of floating-point operations across media scenes.

3.2 The Relationship Between System Load and RT Ratio

Arrival of requests in a recommendation system occurs at rate λ according to a Poisson process, and service times have an exponential distribution with parameter μ. From arrival to receiving recommendations, with m service station, our problem can be abstracted into an M/M/m queueing model (Kendall 1953).

- λ Arrival rate, the average number of requests arriving per unit time.
- μ The average number of requests that can be processed by the CPU per unit time.
- $\frac{1}{\mu}$ The average processing time per request by the CPU, depends on our decisions.
- m The number of service stations, which corresponds to the number of CPU cores here.
- $\rho = \frac{\lambda}{m\mu}$ Service intensity, the proportion of time a single core spends processing requests, is equivalent to CPU utilization.

Based on our modeling using queue theory, the relationship between R (Response Time) and service intensity, or system load, which we simplify as follows:

$$R = \frac{1}{\mu}\left(1 + \frac{c(m,\rho)}{m(1-\rho)}\right) \approx \frac{1}{\mu}\frac{1}{1-\rho^m} \tag{2}$$

It can be observed that if we have both μ (the service rate) and ρ (the utilization rate), we can determine R (the response time). For ease of statistical analysis, we substitute R with the RT ratio for the calculation to standardize the unit which also represents the multiplier that RT increased. Here, we introduce the concept of the RT ratio, which refers to the ratio of the RT (Response Time) required by the current system to process all requests under different CPU utilization levels to the RT consumed under an adequate or baseline CPU utilization level. For example, if the adequate level is at a 30% CPU utilization rate, and

Table 1: Parameters of the fitted exposure rate functions.

	L	k	x_0	R^2
Scene 957	0.618	1.058	31.015	0.9964
Scene 237	0.602	1.224	36.117	0.9987

the RT ratio reaches 1.2 when the CPU utilization is at 50%, this means that the RT required to process requests at a 50% utilization level is 1.2 times the RT required at the 30% level.

According to experience and system stress tests, in different system loads, the total RT in the system generally increases proportionally when fixing the current scene to change the decisions for other scenes. Considering factors such as the system's multi-modular nature, RT volatility, and non-immediate feedback, a more robust modeling approach is required. At this point, we can extend a theoretically complete parameter-free exponential function to a more general exponential relationship. We integrated the system load sensitivity for different scenes into the exponential function relationship based on the sensitivity threshold, which is the $\rho_{threshold}$, shown in equation (3). The specific parameters within the formula can be obtained from short-term stress test data of requests within the system during official online decisions.

$$RT\ ratio = \frac{1}{1-(\rho-\rho_{threshold})^m} \tag{3}$$

3.3 The Relationship Between RT and Exposure Rate

The relationship between RT and exposure rate builds solely on business and data. We found that the response time of requests in different scenes follows a Gaussian distribution, and the trend in exposure rate along with RT can be closely approximated using an S-shaped function (4) parameterized by L,k,x_o and through experiments, we fit the data with two real ad scenes and find that the fit was very good, as shown in Table 1. The fit parameter R^2 is very close to 1, indicating an excellent fit.

$$\text{Exposure rate} = \frac{L}{1+e^{(k\cdot(x-x_0))}} \tag{4}$$

3.4 Overall Modeling of Chain from Allocation Decision to Revenue

Following the introduction of the relationships mentioned above, we summarize the chain from allocation decision to revenue. The comprehensive decision-making process of our system is depicted in the ensuing flowchart Figure 2, which elucidates how our decisions affect the ultimate expected revenue.

1. We initiate with CP to derive a capacity value, denoted as s, then compute the total computational capability available under this capacity to make decisions on CRA, obtaining x_{ij}^k. $x_{ij}^k = 1$ represents the CR to be consumed.
2. Subsequently, using the CR to be consumed, we calculate the system load. Here, we assume synchronous CPU usage, where CPUs with different cores operate in sync. At this juncture, it's essential to evaluate whether the consumed computational capability exceeds its upper limit, obtaining a judgment value represented by z; if exceeded, $z = 1$, otherwise, $z = 0$. If exceeded, we need to adjust the system load to a modified value $\rho_{adjusted}$ to facilitate subsequent calculations of the response time increase ratio.
3. Next, the factor by which the current overall response time r^k is increased is measured, compared to the baseline response time $rt_{baseline}^k$ when computational resources were plentiful, indicating the degree of slowdown due to the present CPU utilization rate.

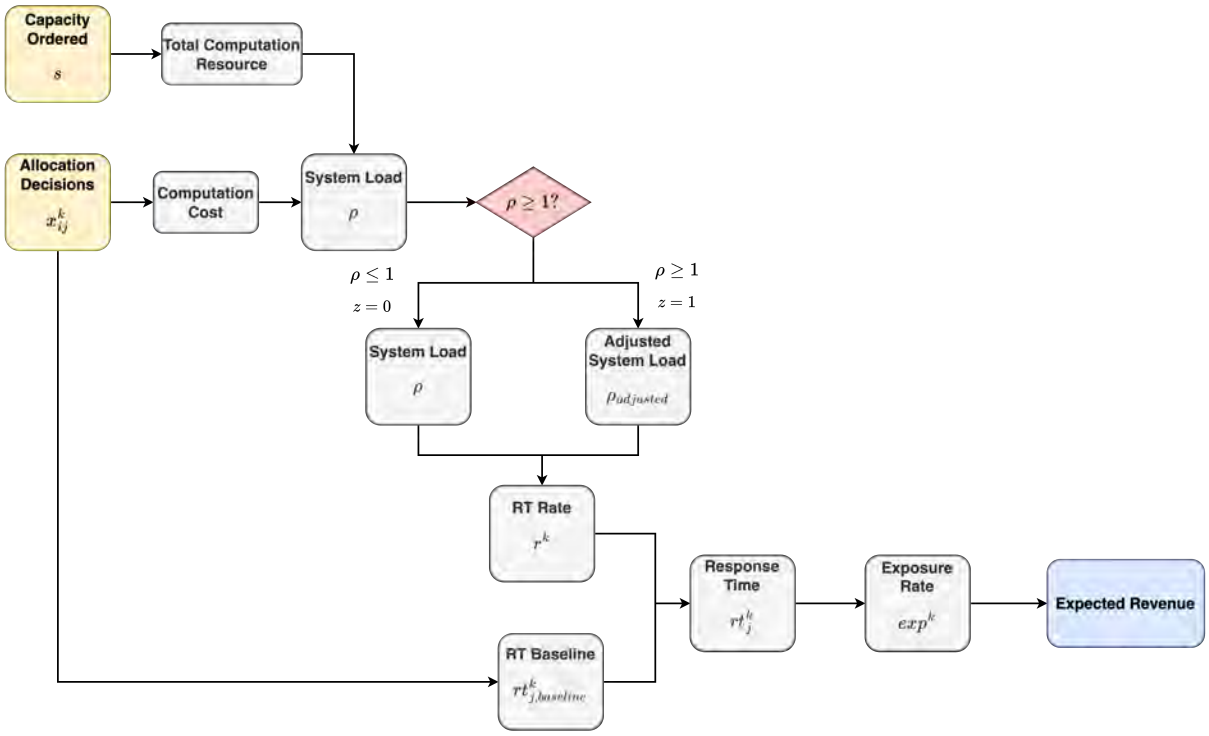

Figure 2: Flowchart of decision-making process.

4. By applying this multiplier r^k to the baseline response time $rt^k_{baseline}$, we can compute the anticipated response time rt^k_j for each ad slot under the current decision.
5. The relationship between this response time rt^k_j and the exposure rate exp^k allows us to understand the exposure situation under the current decision.
6. Because exposure occurs randomly, we can calculate the expected revenue $\mathbb{E}\left[R\left(x^k_{ij}\right)\right]$ for each ad slot, thereby obtaining a final revenue figure.

Through this logic, we can establish a two-stage joint optimization model for CP and CRA considering exposure rate.

4 FORMULATION

Explanations for all notations in the model are listed in Table 2.

Table 2: Notations.

Set	
$\mathscr{K}$	Set of all scenes, $k \in \mathscr{K}$, $k = 1,2...,K$.
$\mathscr{I}^{\|}$	Set of all requests on scene k $i \in \mathscr{I}^{\|}$, $k \in \mathscr{K}$, $i = 1,2,...,M^k$
$\mathscr{A}$	Set of all actions or type of queue length that a request can choose $j \in \mathscr{A}$, $j = 1,2,...,N$
Parameters	
R^k_{ij}	The predicted revenue that can be obtained by choosing the j-th action for the i-th request in the k-th scene.

c_{ij}^k	The predicted cost that can be obtained by choosing the j-th action for the i-th request in the k-th scene, measured by the number of floating-point operations in computer calculations.
C	The total computing capability that a single-core CPU can support, measured by the number of floating-point operations in computer calculations.
α^k	The response time per unit action in the k-th scene.
β^k	The parameter used to describe the relationship between different response time ratios and CPU utilization rates in the k-th scene.
γ^k	The parameter used to describe the relationship between exposure rates and response time in the k-th scene in the denominator.
η^k	The parameter used to describe the relationship between exposure rates and response time in the k-th scene in the numerator.
$rt_{threshold}^k$	The parameter used to describe the threshold of response time threshold in the k-th scene.
$C_{opportunity}$	The opportunity cost of using a single-core CPU.
S	The threshold of total available CPU.
ε	An infinitesimally small positive number.
M	Big-M.
$\rho_{adjusted}$	The adjusted CPU utilization when the total computation cost of the selected action set exceeds the current predetermined computing resource capacity.
Objective	
π	The total revenue after allocating different resources to requests in various scenes.
Variables	
exp^k	The exposure rate in the k-th scene
$rt_{j,baseline}^k$	The baseline response time choosing the j-th action in the k-th scene with abundant computing resources.
r^k	The proportion by which response time is extended in the k-th scene, under conditions of limited computational system resource and varying loads.
rt_j^k	The response time choosing the j-th action in the k-th scene.
ρ	The CPU utilization rate with the chosen action set.
z	An auxiliary variable used for selecting different logical branches based on whether the total computation cost of the selected action set exceeds the current predetermined computing resource capacity.
v^k	Auxiliary variable for the k-th scene used for the linearization of the original constraint.
Decision Variables	
x_{ij}^k	Whether the action j is chosen by request i in the scene k. If $x_{ij}^k = 1$, then the action j is chosen by request i in the scene k. x_{ij}^k is binary variable.
y_j^k	Whether the action j is chosen by all requests in the scene k. If $y_j^k = 1$, then the action j is chosen by all requests in the scene k. y_j^k is binary variable.
s	The number of ordered CPU cores. s is a nonnegative integer variable

Based on the previous analysis, we construct the chain from Decision to Revenue into a two-stage CP and CRA joint optimization model named TS-CPRA.

$$\text{Stage 1:} \quad \pi = \max \mathbb{E}\left[R\left(x_{ij}^k\right)\right] - C_{opportunity} \cdot s \tag{5}$$

$$\text{s.t.} \quad s \leq S \tag{6}$$

$$s \in \mathbb{Z}^+ \tag{7}$$

Stage 2: $$\mathbb{E}\left[R\left(x_{ij}^k\right)\right] = \sum_{ijk} \mathbb{E}\left[\tilde{Z_{ij}^k}\right] \cdot R_{ij}^k \cdot x_{ij}^k \tag{8}$$

s.t. $$\sum_j y_j^k = 1 \qquad \forall k \in \mathscr{K} \tag{9}$$

$$x_{ij}^k = y_j^k \qquad \forall i \in \mathscr{I}^{\parallel}, k \in \mathscr{K} \tag{10}$$

$$p^k = 1 - exp^k \qquad \forall k \in \mathscr{K} \tag{11}$$

$$\rho = \frac{\sum_{ijk} c_{ij}^k \cdot x_{ij}^k}{C \cdot s} \tag{12}$$

$$\rho - (1+\varepsilon) \leq M_1 \cdot z \tag{13}$$

$$\rho - (1-\varepsilon) \geq -M_1 \cdot (1-z) \tag{14}$$

$$r^k - \frac{1}{1-\left(\rho_{adjusted} - \beta^k\right)^s} \leq M_2 \cdot (1-z) \qquad \forall k \in \mathscr{K} \tag{15}$$

$$r^k - \frac{1}{1-\left(\rho_{adjusted} - \beta^k\right)^s} \geq -M_2 \cdot (1-z) \qquad \forall k \in \mathscr{K} \tag{16}$$

$$r^k - \frac{1}{1-(\rho - \beta^k)^s} \leq M_3 \cdot z \qquad \forall k \in \mathscr{K} \tag{17}$$

$$r^k - \frac{1}{1-(\rho - \beta^k)^s} \geq -M_3 \cdot z \qquad \forall k \in \mathscr{K} \tag{18}$$

$$rt_{j,baseline}^k = \alpha^k \cdot j \qquad \forall j \in \mathscr{A}, k \in \mathscr{K} \tag{19}$$

$$rt_j^k = rt_{j,baseline}^k \cdot r^k \qquad \forall j \in \mathscr{A}, k \in \mathscr{K} \tag{20}$$

$$exp^k = \frac{\eta^k}{1 + e^{\gamma^k (\sum_{ij} rt_j^k x_{ij}^k - rt_{threshold}^k)}} \qquad \forall k \in \mathscr{K} \tag{21}$$

$$x_{ij}^k \in \{0,1\}, y_j^k \in \{0,1\} \qquad \forall i \in \mathscr{I}^{\parallel}, j \in \mathscr{A}, k \in \mathscr{K} \tag{22}$$

In stage 1, the objective function (5), aims at maximizing the total advertisement revenue, incorporating the expected value calculation, which can be expressed as follows:

$$\pi = \max \sum_{ijk} \left(1 - p^k\right) \cdot R_{ij}^k \cdot x_{ij}^k - C_{opportunity} \cdot s \tag{23}$$

The term $\tilde{Z_{ij}^k}$ represents a binomially distributed random variable that takes the value 0 with probability p^k, used to determine whether a request is exposed. If the request is not exposed, then $\tilde{Z_{ij}^k} = 0$; if it is exposed, then $\tilde{Z_{ij}^k} = 1$.

Constraints (9), (10) indicate that every request occurring in a scene must select the same action, a requirement determined by the restrictions of the business system. Constraint (11) states that the probability p^k of a single request's revenue dropping to zero equals one minus the exposure rate. Constraint (12) calculates the CPU utilization rate, which equals the computational capability required by the given decision divided by the total pre-allocated computational capability. Constraints (13), (14) assess whether the CPU utilization rate exceeds one. If it does, the computational capability required by the decision exceeds the pre-allocated total, indicating system overload. Due to the function's relationship, when ρ exceeds one, the ratio of response time growth becomes negative, affecting calculations. Therefore, it is necessary to adjust the value of ρ to approach 1 infinitely. If $\rho \geq 1$, then $z = 1$; otherwise, $z = 0$. Constraints (15),

(16) adjust the value of ρ to $\rho_{adjusted}$ when $\rho \geq 1$, establishing a specific functional relationship between $\rho_{adjusted}$ and the proportion by which response time is extended in the k-th scene. Constraints (17), (18) directly constrain the CPU utilization rate ρ and the direct functional relationship with the proportion by which response time is extended in the k-th scene when $\rho \leq 1$. Constraint (19) calculates the baseline response time for choosing the j-th action in the k-th scene with abundant computing resources, proportional to j but varying across different scenes. That is, the response time for the same action under abundant computing resources differs across scenes. Constraint (20) calculates the actual response time based on the current system load, the proportion by which response time is extended in the k-th scene r^k, and $rt^k_{baseline}$. Constraint (21) specifies the exposure rate for scene k based on the response time corresponding to the selected decision. Constraint (6) imposes an upper limit on the capacity that can be reserved, ensuring that it cannot be indefinitely allocated. Constraint (22) enforces integer constraints on the decision variable s and binary constraints on x^k_{ij} and y^k_j.

5 NUMERICAL EXPERIMENTS

This section presents a recommendation system CRA example from Alipay. Given that real data cannot be disclosed, the parameters and examples within the model are adapted based on actual data. The results are obtained by the commercial solver SCIP with Pyomo.

5.1 Results of Capacity Planning and Computation Resource Allocation Task

In this section, we present a specific example and analysis of the results. The case study we employ consists of four scenes, each containing four requests. For each request, we discretize the queue length into an action decision variable where four actions are available for selection.

5.1.1 Optimal Solution Illustration and Sensitivity Analysis of Oppotunity Cost

Upon solving the model using the current parameters, and ensuring that requests in the same scene adhere to the same actions as described in equations (3) and (4), we derive the optimal specific decisions. We assume that the larger the number corresponding to the action j, the higher the benefit for the corresponding scene. There are 4 choices for j, ranging from 0 to 3. The optimal objective function value for j is $(3,2,2,3)$ instead of the highest choices $(3,3,3,3)$. This indicates that increased consumption affects system load, which in turn impacts total revenue. Furthermore, it is observed that the best s equals 7 which does not reach its maximum limit S, proving that more machines do not necessarily equate to better outcomes. This highlights the importance of considering opportunity cost. Under these decisions, the exposure rates for each scene are 43.79%, 48.81%, 37.36%, and 37.02% respectively, with a system load of 60%. This shows that the best strategy is not to use all the machines as much as possible, but to allocate while keeping a certain amount of idle computing power. The revenue at this setting is 8636.

We conduct a sensitivity analysis on two important parameters within the model. Table 3 shows that with the increase in the opportunity cost of machines, the number of CPUs used decreases. However, due to the substantial increase in machine costs, the overall revenue still decreases. At the same time, we uncovered an interesting finding: when computing power is relatively scarce, there is a tendency to reduce the action to achieve a higher exposure rate, thus increasing expected revenue. The opportunity cost changing from 50 to 200 results in a general increase in exposure rate across different scenes. This demonstrates that our model, by fully modeling the factors affecting revenue across the entire chain, enables the computing power allocation model to achieve adaptive regulation in order to reach higher revenue.

Table 3: Optimal solution and corresponding value in model under different opportunity cost of CPU.

	j			exp^k (%)			s			ρ (%)			π		
C_{opp}	50	100	200	50	100	200	50	100	200	50	100	200	50	100	200
Scene 1	3	3	2	49.2	49.22	56.21	7	6	5	63.55	60	69	8420	7600	6466
Scene 2	3	2	2	28.69	48.59	51.19									
Scene 3	1	1	0	21.73	29.15	62.64									
Scene 4	2	2	3	36.84	36.70	62.89									

5.1.2 Model Considering Exposure versus Model with Threshold Constrained Directly on RT

In this section, we validate the effectiveness of our model by comparing it with industrial revenue simulation and allocation model CRA-RT, which automatically interprets decisions to match corresponding scenes with exposure rates, implicitly restricting response time and facilitating revenue calculation.

We divide the experiments into two groups based on the size of the decision space. One group represents a small decision space, consisting of 4 scenes, each with different amounts of traffic and $3-5$ possible decisions and $9-10$ possible decisions respectively.

Firstly, for the experiment with 4 action spaces, we conduct multiple randomized experiments on traffic to simulate the fluctuations of traffic over a month. We find that our TS-CPRA model outperforms the model constrained by RT thresholds most of the time. The experimental results, as illustrated in Figure 3, show the TS-CPRA model's performance with a light sky blue line, while the salmon-colored line represents the CRA-RT model. It's evident that, in most cases, the decisions generated by our model yield benefits surpassing the baseline, with an average revenue increase of 1.9%, and the number of days that TS-CPRA model outperforms CRA-RT is 18 days. For experiments with a larger decision space, as shown in Table 4, our method still outperforms the CRA-RT approach under the same conditions, with an average improvement of 1.95%.

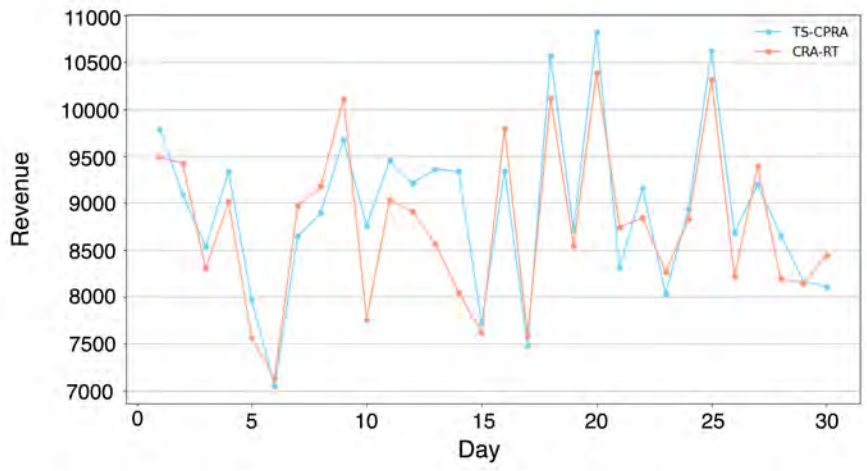

Figure 3: Revenue of requests with TS-CPRA and CRA-RT models in 30 days.

The analysis reveals that in CRA-RT model, the constraints are applied to individual slots, making a static model that doesn't adjust RT calculations based on overall decision-making suboptimal. However TS-CPRA model attempts to constrain RT in decision-making, the variability of RT is not specific to individual scenes but is influenced by all scenes simultaneously. This makes the allocation of computing resources among all scenes more flexible, thereby achieving greater benefits in certain scenes, which in turn increases the overall benefits. This further proves the correctness of our approach in revenue modeling,

Table 4: Optimal solution comparison under different decision spaces of models.

K	N	Maximum revenue of group i				Ave. opt. actual revenue π		P.I.
		Scene 1	Scene 2	Scene 3	Scene 4	TS-CPRA	CRA-RT	
4	10	4600	4900	2800	1200	5486	4830	11.9%
4	10	3600	1700	2500	4500	4369	4465	−2.2%
4	10	1800	3400	1800	2600	3763	3800	−0.9%
4	10	3600	4500	5000	4400	6335	6070	4.2%
4	10	4200	5500	2200	5300	6511	6275	3.6%
4	9	900	3400	3100	4300	4391	4450	−1.3%
4	9	3000	1000	3100	1700	2700	3110	−15.2%
4	9	4500	2000	4400	3200	5494	5135	6.5%
4	9	4100	4900	3700	3900	6805	6375	6.3%
4	9	2300	4400	2000	1800	4510	4210	6.6%

which considers the impact of all decisions on the system state. Additionally, directly constraining RT overlooks the functional relationship between decisions and exposure, which is an incomplete approach. In contrast, our model effectively quantifies the relationship between decisions and RT to exposure rate, thereby offering more optimal decisions and higher revenue.

6 CONCLUSION

Based on real-world internet data, this paper tackles modeling and simulation challenges for computation resource allocation in recommendation systems. We propose a simulation modeling approach that addresses the relationship between resource allocation strategies and benefits, consequently building a two-stage joint optimization model, which achieves a revenue increase of 1.9% compared with the current model of Alipay, and represents flexible adaptation of response time and exposure rate, providing references for the simulation of industrial recommendation computation systems. For future research, this work will be extended to online computation allocation blocks. While our current model addresses offline allocation issues, online tasks often rely on offline decisions and require more immediate allocation decisions to respond to fluctuations in traffic and changes in user behavior. This will improve the overall recommendation system from a more comprehensive and systematic perspective, significantly enhancing both user experience and platform revenue.

ACKNOWLEDGMENTS

This research was funded by the Ant Group through CCF-Ant Research Fund (CCF-AFSG RF20220216), the Science and Technology Innovation Committee of Shenzhen-Platform and Carrier (International Science and Technology Information Center), the Science and Technology Innovation Commission of Shenzhen (JCYJ20210324135011030), and the Guangdong Pearl River Plan (2019QN01X890).

REFERENCES

Adas, A. 1997. "Traffic models in broadband networks". *IEEE communications Magazine* 35(7):82–89.

Alakiri, O. H., A. Oladeji, C. B. Benjamin, C. C. Okolie and M. F. Okikiola. 2014. "The desirability of pareto distribution for modeling modern internet traffic characteristics". *International Journal of Novel Research in Engineering and Applied Sciences* 1(1):2–9.

Chandrasekaran, B. 2009. "Survey of network traffic models". *Waschington University in St. Louis CSE* 567.

Fras, M., J. Mohorko, and Ž. Čučej. 2013. "Limitations of a Mapping Algorithm with Fragmentation Mimics (MAFM) when modeling statistical data sources based on measured packet network traffic". *Computer Networks* 57(17):3686–3700.

Jiang, B., P. Zhang, R. Chen, X. Luo, Y. Yang, G. Wang *et al*. 2020. "DCAF: A Dynamic Computation Allocation Framework for Online Serving System". In *arXiv preprint arXiv:2006.09684*.

Kellerer, H., U. Pferschy, and D. Pisinger. 2004. *The Multiple-Choice Knapsack Problem*, 317–347. Berlin, Heidelberg: Springer Berlin Heidelberg.

Kendall, D. G. 1953. "Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain". *The Annals of Mathematical Statistics*:338–354.

Lu, X., Z. Liu, Y. Guan, H. Zhang, C. Zhuang, W. Ma *et al*. 2023. "GreenFlow: a computation allocation framework for building environmentally sound recommendation system". *arXiv preprint arXiv:2312.16176*.

Neame, T. 2003. *Characterisation and modelling of Internet traffic streams*. Ph. D. thesis, Citeseer.

Ntlangu, M. B. and A. Baghai-Wadji. 2017. "Modelling network traffic using time series analysis: A review". In *Proceedings of the International Conference on Big Data and Internet of Thing*, 209–215.

Olaniran, O. and M. Abdullah. 2020. "Subset selection in high-dimensional genomic data using hybrid variational Bayes and bootstrap priors". In *Journal of Physics: Conference Series*, Volume 1489, 012030. IOP Publishing.

Sapegin, A., A. Amirkhanyan, M. Gawron, F. Cheng and C. Meinel. 2015. "Poisson-based anomaly detection for identifying malicious user behaviour". In *Mobile, Secure, and Programmable Networking: First International Conference, MSPN 2015, Paris, France, June 15-17, 2015, Selected Papers 1*, 134–150. Springer.

Yang, X., Y. Wang, C. Chen, Q. Tan, C. Yu, J. Xu *et al*. 2021. "Computation Resource Allocation Solution in Recommender Systems". *arXiv preprint arXiv:2103.02259*.

Zhou, J., S. Mao, G. Yang, B. Tang, Q. Xie, L. Lin *et al.* 2023. "RL-MPCA: A Reinforcement Learning Based Multi-Phase Computation Allocation Approach for Recommender Systems". In *Proceedings of the ACM Web Conference 2023*, 3214–3224.

AUTHOR BIOGRAPHIES

FEIXUE LIU is a master's student at the Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, China. Her email address is lfx21@mails.tsinghua.edu.cn.

YUQING MIAO is a master's student at the Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, China. Her email address is miaoyq21@mails.tsinghua.edu.cn.

YUN YE is a Senior Algorithm Expert at Ant Group. She is responsible for the strategic mechanism optimization of Alipay advertisements. Her research interests includes optimal operational control of advertising, computational resource scheduling, hyperparameter optimization. Her email address is yeyun.yy@antgroup.com.

PEISONG WANG is a master's student at the Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, China. His email address is wps22@mails.tsinghua.edu.cn.

XINGLU LIU is a Ph.D. candidate at the Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, China. His email address is liuxl18@mails.tsinghua.edu.cn.

KAI ZHANG is presently an Associate Professor in the Shenzhen International Graduate School, Tsinghua University. Dr. Kai Zhang received the B.E. and Ph.D. all from Tsinghua University, China, in 1999, and 2004 respectively. His research interests include wireless optical communication systems, sensor fusion, highly accurate positioning, electric vehicle, and intelligent transportation systems. His email address is zhangkai@sz.tsinghua.edu.cn.

WAI KIN (VICTOR) CHAN is Professor of the Tsinghua-Berkeley Shenzhen Institute (TBSI) and Shenzhen International Graduation School (SIGS), Tsinghua University, China. His research interests include discrete-event simulation, agent-based simulation, big-data analytics and their applications in social networks, business big-data, service systems, healthcare, transportation, energy markets, and manufacturing. His email address is chanw@sz.tsinghua.edu.cn.