Incentive-Vacation Queueing for Edge Crowd Computing

Sherif B. Azmy, *Graduate Student Member, IEEE*, Nizar Zorba, *Senior Member, IEEE*, and Hossam S. Hassanein, *Fellow, IEEE*

***Abstract*—Edge computing aims to push services closer to end-users, greatly enhancing latency and scale. Yet, there's untapped potential beyond the network's last mile, on the extreme edge. Extreme edge computing (XEC) is a computing paradigm that exploits computational resources in the end-user's immediate vicinity. Edge crowd computing (ECC) is an orchestrated sharing economy model within XEC that uses idle resources on user-owned devices for service provision, compensating owners. We analyze an orchestrated ECC where devices rent resources in exchange for incentives. Our incentive-vacation queueing (IVQ) model associates performance with incentive payments using vacation queueing, considering the multitenancy of devices through a server vacation dependent on incentives received. In this article, we offer a framework for analyzing any sharing economy system that can be modeled using IVQ. We discuss the relationship between incentives and vacations on performance, namely, the incentive-vacation or IVQ function. We examine two families of IVQ functions that can be adjusted to benefit either the orchestrator or the worker and introduce a performance metric for such preference. We derive analytical expressions for system performance that consider the random nature of worker devices' availability due to fluctuating incentives. The IVQ model explores commodifying user-owned resources in an ECC system, presenting a general approach for performance analysis in such environments.**

***Index Terms*—Extreme edge computing (XEC), Incentive, performance analysis, queueing, vacation.**

I. Introduction

CLOUD computing has emerged as a fundamental component in contemporary trade and service exchanges [1]. Its expansion is marked by three notable trends: 1) the swift progress of the Internet of Things (IoT) and explosion in number of its devices [2]; 2) the need for advanced applications, such as VR/AR; and 3) the imperative of ensuring security and privacy. This growth may soon surpass the centralized cloud and its backhaul network's capacity [3]. To address this, Fog and Edge Computing have been introduced to decentralize and meet demand by bringing services closer to users [4]. Nonetheless, the capabilities of idle, powerful consumer devices like PCs, smart devices, wearables, vehicles, and appliances are underutilized [5], [6]. These devices have evolved from specialized hardware to versatile computing tools [3].

User devices have the potential to create an underutilized tier of edge computing, which we term extreme edge computing (XEC). This involves processing done exclusively on user devices, characterized by their proximity to end-users and modest processing power [7]. XEC harnesses collective idle computational resources across numerous devices rather than relying on singular device capabilities [8]. This has become possible due to nascent technologies, such as unikernels and microcontainers [9], [10]. While inherently decentralized, XEC can also adopt a semi-decentralized architecture. edge crowd computing (ECC) represents this semi-decentralized model, where service providers distribute their offerings through third-party devices within the user's vicinity, paralleling the service provision models of companies like Uber or Airbnb [11].

ECCs present a notable use case in game streaming services over wireless networks. For low-capability devices like smartphones, the service provider uses edge servers for streaming sophisticated games, aiming to reduce latency and enhance the user experience. Mobility-related connectivity issues, for instance, on a train, may disrupt service access, preventing enjoyment of the game [12]. Although mobile devices can store the game, they often lack the necessary processing power and battery life to run it effectively. To counter unreliable connections, the service provider might transfer game assets directly to the user's device, which temporarily coordinates gameplay by utilizing nearby devices, such as those of fellow passengers, by sharing the game's workload [13]. These "worker" devices would receive compensation from the service provider. While not yet viable with current technology, this model is foreseen with the advent of 6G and D2D URLLC advancements in XEC and ECC frameworks [14], [15].

Edge Cloud Computing (ECC) is promising but comes with unique challenges. This study tackles two main concerns: 1) the need for user-owned devices to perform reliably despite being multitenant and 2) the diversity in usage, connection, and capabilities among these devices. Deploying ECC on personal devices is complex, primarily due to their and their owners' unpredictable reliability, often linked to the owners' behavior.

Manuscript received 14 August 2023; revised 9 November 2023 and 16 December 2023; accepted 18 December 2023. Date of publication 27 December 2023; date of current version 9 April 2024. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant ALLRP549919-20; in part by the Distributive Ltd.; and in part by the Qatar National Library. *(Corresponding author: Nizar Zorba.)*

Sherif B. Azmy is with the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON K7L 3N6, Canada (e-mail: sherif.azmy@queensu.ca).

Nizar Zorba is with the College of Engineering, Qatar University, Doha, Qatar (e-mail: nizarz@qu.edu.qa).

Hossam S. Hassanein is with the School of Computing, Queen's University, Kingston, ON K7L 3N6, Canada (e-mail: hossam@cs.queensu.ca).

Digital Object Identifier 10.1109/JIOT.2023.3347442

Similar to mobile crowd sensing (MCS), which leverages incentives to manage participant engagement [16], we argue that such strategies can be adapted for computational tasks in ECC. However, while MCS rewards are for human input, ECC incentives compensate for any device performance dips when sharing computational tasks. Thus, incentives are crucial in ECC for converting computational resources into tradable assets [11].

Several studies have explored Edge Computing frameworks where there is an interaction between customers and service providers. Harth-Kitzerow and Garrido [17] tackled the issue of mutual self-interest—workers wanting maximum reward for least effort and customers wanting maximum service for least payment—by introducing a validation entity to ensure job execution fidelity by re-evaluating job samples. Additionally, [18] presents a macroeconomic analysis based on a double auction model, focusing on enhancing social welfare considerations that align the objectives of both service providers and customers. Similarly, [19] proposes a dynamic pricing strategy in a sharing economy context to improve social welfare. Although social welfare is an important overarching metric, it is essential to acknowledge the diverse array of more immediate system performance indicators, such as latency and throughput, especially given the variable and delay-sensitive nature of edge computing environments [20].

XEC and ECC systems are emerging technologies that have not been extensively explored in existing research. The untapped computational resources of advanced edge devices are now more accessible, thanks to advancements in virtualization, software-defined networking, and containerization [9], [12], [21], [22]. These technologies enhance network adaptability and autonomous organization, facilitating seamless computational handover from user devices to proximal devices [23]. The efficacy of new semi-decentralized XEC systems, like ECC, remains unquantified. Performance largely depends on workers' profit motives. Deployers of ECC orchestrators can gain from understanding the correlation between incentives and performance. This insight is crucial for reducing recruitment costs and maximizing profits while delivering services on time and to customer satisfaction. Therefore, investigating the incentives-performance relationship is imperative.

This study examines ECC systems with orchestrator oversight, where incentives mitigate uncertainty. We introduce the incentive-vacation queueing (IVQ) framework, a queueing-theory method, to evaluate how incentives influence a worker's output. IVQ focuses on multitenant worker devices adopting a vacation scheme, whereby the device attends to its owner's requirements post ECC service. We assess our model via vacation queueing theory, specifically the M/G/1 queue with restricted pure vacations, noting that incentives, being stochastic, alter vacation lengths. This article's contributions are as follows.

1) We propose the IVQ model with an incentive-vacation function, investigate the impact of its convexity on the system's preference of workers and orchestrator, and the performance of workers in the system.
2) We formulate a general form of a variable preference incentive-vacation function and propose two incentive-vacation function families, the log-family and the rational-family. Furthermore, we provide a general framework for analyzing systems in which incentives can impact performance.
3) We derive closed-form expressions for performance in an ECC system as well as propose two sharing economy-based performance metrics: a) revenue per job and b) revenue per second.
4) We investigate the impact of incentives distributed by the orchestrator on the worker's performance.

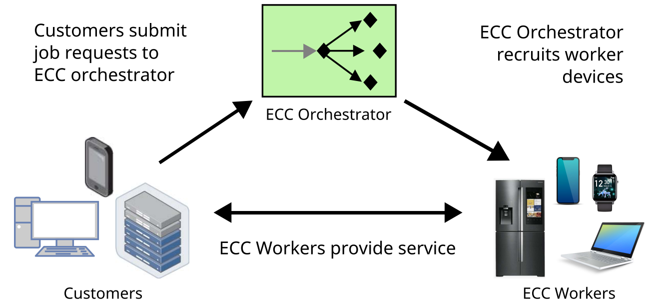


Fig. 1. ECC system.

The structure of this document is as follows: Section II presents a synopsis of extreme edge systems and outlines the system architecture. Section III explores vacation queueing, its connection to incentives, and introduces the IVQ model, examining the effects of their convex relationship on the system. Section IV analyzes the decision-making process of the orchestrator concerning worker incentives within IVQ systems. Section V interprets the analytical results through a numerical illustration. Section VI discusses practical implications of implementing the IVQ model. This article concludes in Section VII with a summary and prospects for subsequent research.

II. Extreme Edge System Overview

XEC is laden with heterogeneity and uncertainty because it relies on user-owned infrastructure that suffers from numerous sources of uncertainty [24], [25]. Efficient utilization of XEC infrastructure involves taming with uncertainty and constraining it so that the system's behavior can be predicted. However, the nature of user-owned devices is sporadic as they are used by their owners. Thus, an XEC system should seek to utilize the idle resources abundant on these user-owned devices in spite of their owner's access to them. A service can be deployed on such devices that are present in the end-user's immediate vicinity to ultimately provide timely service to other customers. While this is possible in theory, achieving reliable service provision is a challenge of XEC. In this section, we describe an ECC system, an instance of XEC, in which the service provider's orchestrator recruits user-owned devices, or workers, to provide service to customers. The worker devices agree to provide their resources as infrastructure for the service provider, an ECC orchestrator, in exchange for an incentive payment. Fig. 1 illustrates an orchestrated ECC system, comprised of three main entities.

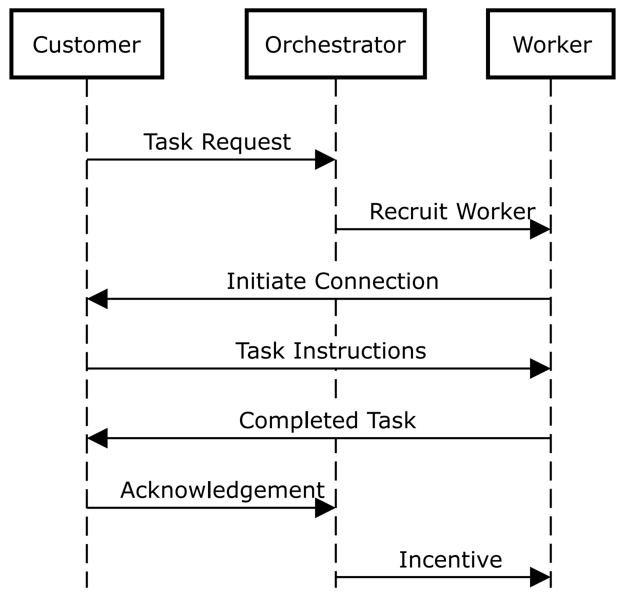


Fig. 2. Operation of an ECC system: Customers request service from the orchestrator that would recruit workers for service provision.

1) *Customer Devices:* These are the client devices that request and use the services provided by the ECC service provider.
2) *Worker Devices:* These are user-owned devices that rent their idle computational resources to the ECC service provider in exchange for an incentive.
3) *Orchestrator:* The central component of the orchestrated ECC system that coordinates the activities of the customer and worker devices, to ensure that jobs are completed efficiently. In addition, the orchestrator is also responsible for providing incentives to the worker devices. In addition to acting as a liaison between customers and workers, the orchestrator also tracks the performance of workers and their profitability to avoid unexpected worker churning.

Fig. 2 illustrates the operation of an orchestrated ECC system. First, the customer submits a job request to the nearest orchestrator. The orchestrator can also be a latent component that is available on the customer device itself and activated when the service provider is out of reach. The orchestrator then recruits a worker to perform this job. After receiving the job request from the orchestrator, the worker and the customer directly connect to each other so that the detailed job instructions are transmitted (or required assets). Finally, the worker services the job and returns it to the customer device which acknowledges its completion. Upon acknowledgment, the orchestrator processes the incentive payment to the worker device.

This system can be viewed as a server farm [26] in which the ECC orchestrator represents a scheduler with a queue in which customers' jobs – with attached job incentives – arrive, and are then distributed to the ECC workers as illustrated in Fig. 3.

Generally, customer service requests can be modeled into an orchestrator arrival rate, λ_{orch}, which is split by the orchestrator according to some scheduling mechanism into worker arrival rates $\lambda_{\text{orch}} = \sum_{i=1}^{K} \lambda_i$, assuming a total of K workers. Nevertheless, these workers are not entirely dedicated to servicing the edge, since they are also user devices, making them multitenant devices that provide service for more than one entity. In this case, workers serve both the XEC system (in exchange for an incentive) in addition to their owners. As an example, the owner of a personal computer can rent its idle computational resources to the XEC orchestrator in addition to using the same device for themselves. Therefore, worker devices need to manage their resources in a manner that allows them to serve both the XEC system and its owner. Such scenario is of great importance for current and future trends of infrastructure-less sharing economy-based companies, such as Uber, Airbnb, etc. This article presents methods of modeling and improving such systems.

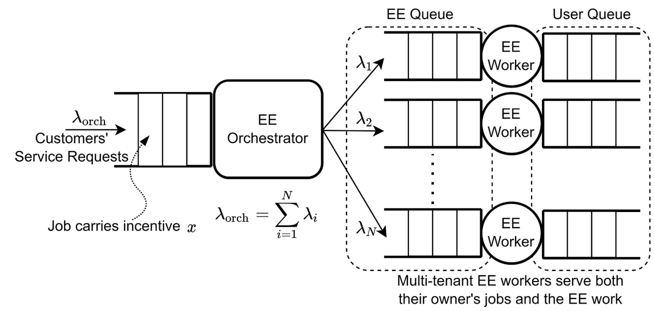


Fig. 3. ECC job distribution model: The orchestrator distributes jobs to multitenant workers.

In this article, we investigate performance from the perspective of an individual worker. We model the service time spent on serving the owner's jobs as a vacation whose duration is dictated by the amount of incentive the worker receives. In the following section, we will briefly overview vacation queueing prior to introducing the proposed IVQ model. Table I lists the notations used throughout this article.

III. Modeling of Incentive-Vacation Queueing (Worker Perspective)

In an ECC system, the availability of worker devices and their resources is heavily influenced by the users' behavior. This spontaneity introduces a degree of uncertainty in the availability of those resources. However, the uncertainty stemming from the user's behavior can be mitigated by the use of incentives that target the human user.

In this section, we employ vacation queueing to abstract the user behavior and the influence of stochastic incentives over it. We give an overview of vacation queueing, and introduce the IVQ model which, using vacation queueing, captures the impact of incentives in an ECC system. We provide an analysis of the system, the relationship between vacations and incentives, and the performance of an orchestrator-based ECC system.

A. Overview of Vacation Queueing

Vacation queueing is a type of queueing in which the server becomes unavailable for a period of time called a vacation.

TABLE I
NOTATIONS AND SYMBOLS

Symbol	Description
V	Vacation duration
$\mu = \frac{1}{S}$	Worker's service rate
$\lambda = \frac{1}{\tau}$	Job arrival rate to worker's queue
$\rho, \widetilde{\rho}$	Server utilization, PVQ utilization
X	Total Queue Incentive (TQI)
$V_{\text{min}}, V_{\text{max}}$	Minimum and maximum vacations
$X_{\text{min}}, X_{\text{max}}$	Minimum and maximum TQI
$r(\cdot)$	Vacation-Incentive (VI) function
$r_{\text{cvx}}(x)$	Convex VI Seed Function
$r_{\text{ccv}}(x)$	Concave anti-convex VI Function
$r_{\text{lnr}}(x)$	Linear VI Function
α	Orchestrator-Worker Preference Parameter
$(\cdot)^+$	Shorthand for $\max(0, \cdot)$
β	Logarithmic scale parameter
$r_{\text{log}}(x, \alpha, \beta)$	log-family of VI Functions
$r_{\text{rat}}(x, \alpha)$	rational-family of VI Functions
$r_{\text{abs,cvx}}(x)$	Convex Corner Function
$r_{\text{abs,ccv}}(x)$	Concave Corner Function
$V_{\%}$	Vacation proportion of service cycle
Q_v	Queue Length
L_v	Number of jobs in the system
T_{Qv}	Waiting time
$R, R_{\text{min}}, R_{\text{max}}$	Per job revenue, minimal and maximal
$\lambda_{\text{min}}, \lambda_{\text{max}}$	Minimal and maximal throughput
Q_w	Worker's Queue
x_i	Job Incentive

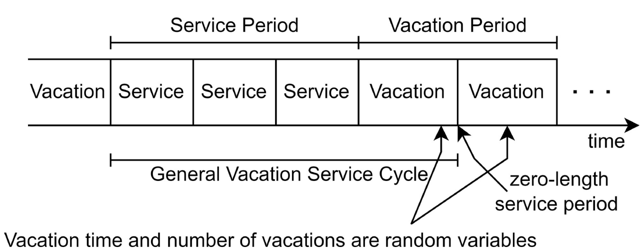


Fig. 4. General vacation model: Service periods can be followed by vacation periods.

Implementing a vacation policy introduces a degree of flexibility in the modeling of real systems as vacations abstract the server's other duties into a single random variable, V, that represents the duration of a server vacation [27], [28], [29]. For example, single server vacation queueing could be classified according to the vacation policy. The vacation policy can be *exhaustive* or *nonexhaustive*, with regards to whether the server starts its vacation only after having finished the queue or not. There are different types of vacation queues as well as to whether there is a threshold (i.e., a specific number of vacations has occurred or not), whether it is preemptive or not, or whether the service is gated or not [28].

Fig. 4 depicts the general vacation server's activity over time. If the type of vacation model allows consecutive service with no vacation in between, then the *service period* is the total period for which the server was busy. Similarly, if consecutive vacations have no service in between (i.e., zero-length vacation), then the *vacation period* is the total duration of consecutive vacations. Generally, the *service cycle* spans the service period and a single vacation. This is the case for general vacation models [27]. In this work, we use a certain type of vacation queueing, *P*-limited vacation queueing (PVQ) to model ECC workers in an extreme edge scenario.

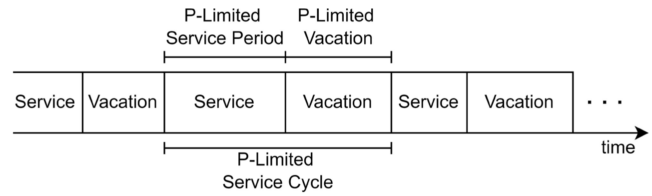


Fig. 5. P-limited vacation model: Each service period must be followed by a vacation period.

B. P-Limited Vacation Queueing

P-Limited, or pure limited, Vacation Queueing is a type of nonexhaustive vacation queueing in which the server takes a vacation after each departure, limiting its service period to a single job [27], [28]. If no jobs were queued for service at a vacation completion instance, the server keeps repeating its vacations until a job arrives. Fig. 5 illustrates the server's activity over time in PVQ.

What makes PVQ interesting is its compatibility with multitenancy as it would abstract the worker's activities in the vacation. The modeling of the vacation as a random variable (or a function of a random variable) simplifies the complexity of the worker's non-XEC workload into the vacation duration V. In addition, PVQ's analysis is simple as it can be regarded as a modification of the service time by introducing the length of the vacation to it. In other words, an $M/G/1$[1] queue changes to have a modified service time, becoming an $M/\widetilde{G}/1$ queue, in which the modified service time, $\widetilde{S}$, becomes a sum of the $M/G/1$ service time, S, and the vacation time, V, i.e., $\widetilde{S} = S + V$ which is possible due to the stochastic decomposition property [28]. $\widetilde{S}$ also represents the duration of the PVQ service cycle. A consequence is that the stability condition then becomes

$$\widetilde{\rho} = \rho + \lambda \mathbb{E}[V] < 1 \tag{1}$$

where λ is the arrival rate, $\rho = \lambda S$ is the $M/G/1$ server utilization, $\widetilde{\rho}$ is the PVQ server utilization, and $\mathbb{E}[V]$ is the average vacation duration.

C. Incentive Vacation Queueing

IVQ is a form of PVQ in which vacation duration, V, is defined as a function of the total incentive, X, a worker is receiving. This relationship is illustrated in Fig. 6. The incentive X is a random variable making V a function of a random variable. By defining V as a function of X, V inherits the randomness of X and becomes a random variable. Depending on the incentive's origin, X can be defined as the sum of incentives to each of the jobs in a worker's queue at a time instant, thus looking at incentives coming directly from the customers, or it can be defined as a set payment value that the orchestrator chooses. In both cases, we name the IVQ incentive variable, X, as the total queue incentive (TQI).

[1]An M/G/1 queue represents a system with Markovian (Poisson) arrivals, a general service time distribution, and one server.

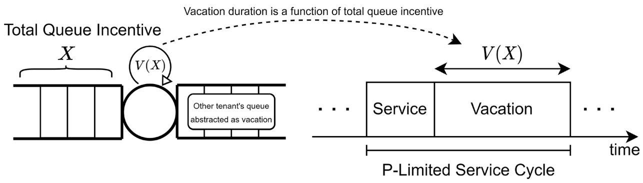


Fig. 6. In IVQ, the impact of incentives is represented via vacation duration.

The implications of both cases are covered in the following section on the performance modeling of an IVQ system.

Incentives have varying forms; however, they can be categorized into two main categories: 1) monetary and 2) nonmonetary incentives. Unlike monetary incentives in which money is used to incentivize, nonmonetary incentives include rewards, such as recognition, badges, points, or discounts. Nonmonetary incentives also extend to include forms of incentive that stem from an existing context, such as a game (e.g., in-game rewards or rank) or a community (e.g., contributions of a Google Maps guide), or the gamification of an activity (e.g., gamified focus timer) [30]. The design of incentives has various aspects according to the context, such as the setting in which the ECC system takes place, what audience comprises the worker devices and their demographics, or even if it is a closed setting [31].

In IVQ, we define V as a function of the incentive $r(X)$. The choice of $r(X)$ has to be chosen such that the system is stable and the performance is achieved. The proper selection of $r(X)$ and its properties are of great effect on the IVQ as the vacation is the abstraction of user behavior. Aspects that extend beyond just incentives can be captured by $r(X)$. For example, $r(X)$ can be defined in terms that relate to worker's trustworthiness, benchmark capabilities, and reliability, and thus allow the study of randomness stemming from these factors. However, in this work, we isolate our analysis to only random behavior stemming from the presence of incentives. The vacation duration is inversely proportional to the incentive, which requires $r(X)$ to be monotonically decreasing as more incentives cause shorter vacations. In IVQ, the stability condition becomes

$$\widetilde{\rho} = \rho + \lambda\mathbb{E}[V] = \rho + \lambda\mathbb{E}[r(X)] < 1 \tag{2}$$

or equivalently, it can be rewritten as

$$\mathbb{E}[r(X)] < \frac{1}{\lambda} - \frac{1}{\mu} \tag{3}$$

where $\mu = 1/S$ is the service rate.

Equation (3) is useful as it shows that $\sup V = 1/\lambda - 1/\mu$, i.e., the least upper bound for the vacation is the difference between the interarrival rate and the service time. This is mainly due to the fact that under PVQ a queue is stable (i.e., not infinitely growing) if and only if the service cycle as at least long as the interarrival time. As such, the duration of the vacation, V is either zero or approaches $1/\lambda - 1/\mu$, i.e., $V \in [0, 1/\lambda - 1/\mu)$. Thus, the operation range of IVQ is dictated by either pegging λ and μ or—as will be shown throughout this section—by the choice of boundaries $V_{\min} < V = r(X) < V_{\max}$ in the design of $r(X)$. In this work, we opt for the latter case, i.e., choosing a convenient $V_{\min}$ and

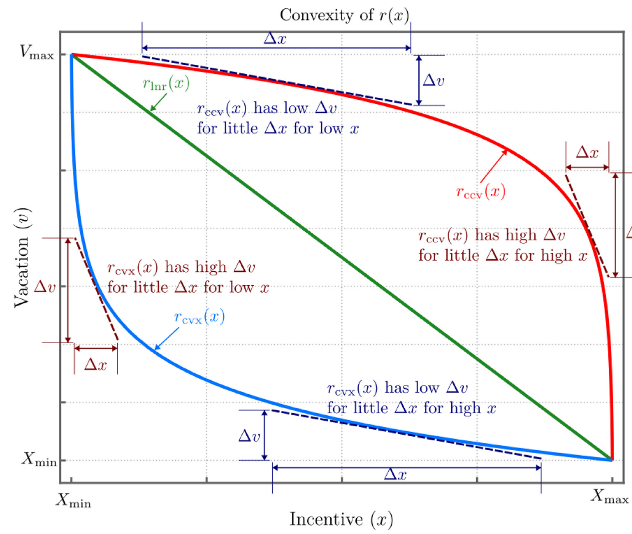


Fig. 7. Convexity of $r(x)$ has an impact on the marginal value of vacations and thus the choice of IVQ function can be biased to favor one party over another. The α is used to control the IVQ function's preference.

$V_{\max}$ since the arrival rate λ is controlled by the orchestrator in an orchestrated ECC setting, given the worker's μ, and since the choice of proper boundaries would guarantee the queue stability. Thus, the choice of $r(X)$ has to respect $r(X_{\min}) = V_{\max}$ and $r(X_{\max}) = V_{\min}$, i.e., $X \in [X_{\min}, X_{\max}] \mapsto V \in [V_{\min}, V_{\max}]$, to guarantee stability.

D. Convexity of $r(X)$

The choice of $r(X)$ has significant implications on the behavior of the system. Convexity, in particular, impacts whether the system leans toward—or favors—the orchestrator or the worker. To illustrate this, we look at the incentive-vacation function, which is—as previously mentioned—monotonically decreasing. As a consequence, the second derivative, $r''(x)$ which gives us information about $r(x)$'s convexity ($r(x)$ is convex $r''(x) > 0$, concave $r''(x) < 0$, or linear $r''(x) = 0$) is also giving us information about how $r(x)$ decreases. For the worker, it is favorable if this quantity decreases as it would imply that the worker is not losing much vacation (i.e., portion of the cycle that goes to servicing the worker's owner) per unit incentive. On the other hand, it would be preferrable for the orchestrator to have this quantity increasing, as it would mean that they would be gaining more service per unit incentive. A convex function, in that regard, is orchestrator-favoring, while a concave function is worker-favoring, and in between a linear function is one that treats both fairly. Fig. 7 illustrates this preference in the incentive-vacation function.

As a consequence, the choice of a worker's incentive-vacation function needs to take various factors into consideration. For example, if the worker is new to the system and not much information is available about them, both the orchestrator and the worker can agree on a convex $r(x)$ until the worker proves their worth, and then move to a different $r(x)$ that decreases the worker's vacation per unit incentive. In the following section, we propose a general formulation of

$r(x)$ as a incentive-vacation function whose convexity can be tuned by a parameter, α.

E. Variable Convexity Incentive-Vacation Function

The objective of this section is to model a wide class of incentive-vacation functions that exist in practice, and through a tuning parameter, α, we can mathematically control the convexity of the incentive-vacation function. Given a seed convex function, $r_{\text{cvx}}(x)$ that starts at the point $(X_{\min}, V_{\max})$ and ends on $(X_{\max}, V_{\min})$, we obtain a *concave anti-convex function*, $r_{\text{ccv}}(x)$ that starts and ends on the same points, but decreases concavely over the interval $[X_{\min}, X_{\max}]$. In addition, we construct a linear function, $r_{\text{lnr}}(x)$, between the points, and combine the three functions in a single α-parameterized functions that allows choice of $r_{\text{cvx}}(x)$, $r_{\text{lnr}}(x)$, $r_{\text{ccv}}(x)$ or a convex combination of them.

The concave anti-convex of $r_{\text{cvx}}(x)$, $r_{\text{ccv}}(x)$ can be obtained by reversing the behavior of the slope over the interval $[X_{\min}, X_{\max}]$, i.e.,

$$r'_{\text{cvx}}(x) = r'_{\text{ccv}}(X_{\max} + X_{\min} - x) \tag{4}$$

which expresses the relation between the slope of $r_{\text{ccv}}(x)$ and $r_{\text{cvx}}(x)$. The function $r_{\text{ccv}}(x)$ can then be obtained by integrating both sides and substituting with the inverse of the argument of $r'_{\text{ccv}}(x)$ to get

$$r_{\text{ccv}}(x) = -r_{\text{cvx}}(X_{\max} + X_{\min} - x) + V_{\max} + r_{\text{cvx}}(X_{\max}) \tag{5}$$

where $r_{\text{ccv}}(x)$ is the *concave anti-convex function* for any $r_{\text{cvx}}(x)$. The linear function, $r_{\text{lnr}}(x)$ is defined as

$$\begin{aligned} r_{\text{lnr}}(x) = &\frac{V_{\min} - V_{\max}}{X_{\max} - X_{\min}} x \\ &+ V_{\max}\left(1 - \frac{X_{\min}(V_{\min} - V_{\max})}{V_{\max}(X_{\max} - X_{\min})}\right). \end{aligned} \tag{6}$$

We then combine all three functions in an α-parametrized convex combination

$$r(x, \alpha) = (-\alpha)^+ r_{\text{cvx}}(x) + (-|\alpha|)^+ r_{\text{lnr}}(x) + (\alpha)^+ r_{\text{ccv}}(x) \tag{7}$$

where $(\cdot)^+ = \max(0, \cdot)$, $\alpha \in [-1, 1]$, and the expression becomes $r_{\text{cvx}}(x)$ for $\alpha = -1$, $r_{\text{lnr}}(x)$ for $\alpha = 0$, and $r_{\text{ccv}}(x)$ for $\alpha = 1$. As a consequence of this definition, $r(x, \alpha)$ is convex for $\alpha \in [-1, 0)$ and concave for $\alpha \in (0, 1]$.

The selection of the seed $r_{\text{cvx}}(x)$, thus, dictates the sort of *vacation function family* that $r(x, \alpha)$ belongs to. To illustrate, we shall derive two vacation families: 1) the *log-family* and 2) the *rational-family*, that cover the modeling of different realistic applications. For instance, Uber transportation is a high stakes as customers directly interact with workers and an unfavorable interaction would negatively impact Uber's reputation. This is due to the fact that user's satisfaction can easily fall if the minimum service is was poorly provided [32]. Such a service would benefit from using a log incentive-vacation function as it has a high degree of bias in preferring the orchestrator (Uber in this case) to the worker (the driver) for low α. This preference reverses for high α if the driver is a reputable and reliable driver. On the other hand, the rational function could be useful in a food delivery service, such as Uber Eats, which does not often involve a prolonged interaction with the driver, and also has a larger population of drivers to recruit than transportation. Having a rational incentive-vacation function provides a fairer relationship between vacations and incentives that does not excessively bias the system toward neither the orchestrator nor the worker.

It is important to note that both, the log and the rational families are two flexible examples that cover a wide range of functions. An orchestrator can mix-and-match different $r_{\text{cvx}}(x)$'s with concave functions that are not anti-convex of $r_{\text{cvx}}(x)$. The framework provided in this work is a general framework for the analysis for any system in which an incentive-vacation function $r(x, \alpha)$ is generated from a convex seed $r_{\text{cvx}}(x)$. The choice of such function depends on the context of the service and the goals of the orchestrator.

The log-family stems from the choice of $r_{\text{cvx}}^{(\log)}(x, \beta) = \log_\beta(Ax+B)$, whose parameters A and B can be found through the initial conditions

$$-\log_\beta(AX_{\min} + B) = V_{\max}, \quad -\log_\beta(AX_{\max} + B) = V_{\min} \tag{8}$$

to acquire

$$A = -\frac{\beta^{V_{\max}} - \beta^{V_{\min}}}{X_{\min} - X_{\max}}, \qquad B = \frac{\beta^{-V_{\max}} X_{\max} - \beta^{-V_{\min}} X_{\min}}{X_{\max} - X_{\min}} \tag{9}$$

that result in

$$r_{\text{cvx}}^{(\log)} = -\log_\beta\left(\frac{\beta^{-V_{\max}-V_{\min}}\left(\beta^{V_{\max}}(x - X_{\min}) + \beta^{V_{\min}}(X_{\max} - x)\right)}{X_{\max} - X_{\min}}\right). \tag{10}$$

Thus, we obtain the concave anti-convex of $r_{\text{cvx}}^{(\log)}(x, \beta)$ as

$$\begin{aligned} r_{\text{ccv}}^{(\log)} = &V_{\max} + V_{\min} \\ &+ \log_\beta\left(\frac{\beta^{-V_{\max}-V_{\min}}\left(\beta^{V_{\max}}(X_{\max} - x) + \beta^{V_{\min}}(x - X_{\min})\right)}{X_{\max} - X_{\min}}\right) \end{aligned} \tag{11}$$

Then, we could combine this with $r_{\text{lnr}}(x)$ to obtain

$$\begin{aligned} &r_{\log}(x, \alpha, \beta) \\ &\quad = (-\alpha)^+ r_{\text{cvx}}^{(\log)}(x, \beta) + (1 - |\alpha|)^+ r_{\text{lnr}}(x) + (\alpha)^+ r_{\text{ccv}}^{(\log)}(x, \beta). \end{aligned} \tag{12}$$

Similarly, we can derive the rational-family of vacation functions by taking

$$\begin{aligned} r_{\text{cvx}}^{(\text{rat})}(x) &= \frac{1}{Cx + D} \\ &= \frac{V_{\max} V_{\min}(X_{\min} - X_{\max})}{(V_{\min} - V_{\max})x - X_{\max} V_{\min} + V_{\max} X_{\min}} \end{aligned} \tag{13}$$

with the concave anti-convex of $r_{\text{cvx}}^{(\text{rat})}(x)$ being

$$r_{\text{ccv}}^{(\text{rat})}(x) = \frac{V_{\max}^2 (x - X_{\max}) + V_{\min}^2 (X_{\min} - x)}{V_{\max}(x - X_{\max}) + V_{\min}(X_{\min} - x)} \tag{14}$$

which gives us the rational-family

$$\begin{aligned} &r_{\text{rat}}(x, \alpha) \\ &\quad = (-\alpha)^+ r_{\text{cvx}}^{(\text{rat})}(x) + (1 - |\alpha|)^+ r_{\text{lnr}}(x) + (\alpha)^+ r_{\text{ccv}}^{(\text{rat})}(x). \end{aligned} \tag{15}$$

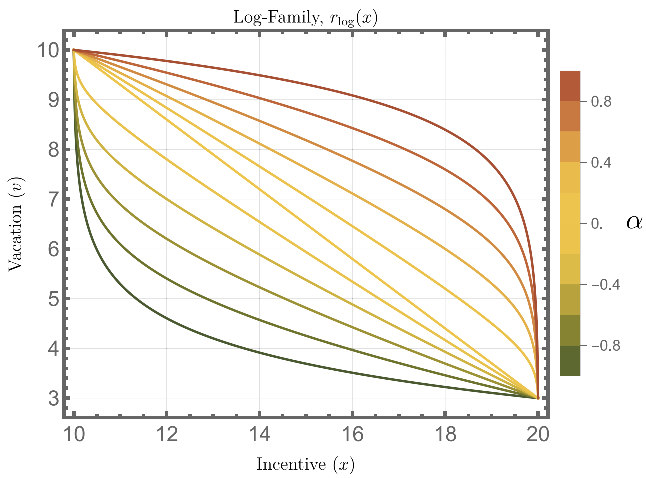


Fig. 8. Log-family for variable α. The log-family's preference does not rely only on α, but it also has an inherent bias toward one party.

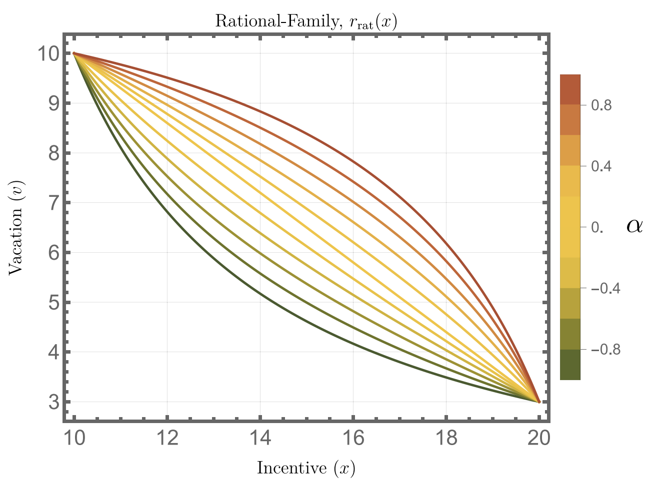


Fig. 9. Rational-family for different α. In comparison to Fig. 8, the rational family is less obtuse, not having an inherent preference to either the orchestrator or the worker and relying only on α.

Both families have different behaviors with respect to how the marginal value of vacation (i.e., vacation variation per unit incentive) changes, and they correspond to different scenarios. Figs. 8 and 9 shows a plot of both incentive-vacation families for an arbitrary choice of $X_{\min}$, $X_{\max}$, $V_{\min}$, $V_{\max}$ and variable α.

F. Measuring $r(x)$'s Preference

The convexity of the function $r(X)$ can give an indicator to whether the incentive-vacation function prefers—or is biased—toward the orchestrator or the worker: decreasing convex implies bias toward the orchestrator while decreasing concave leans toward the worker. While the second derivative of $r(X)$ gives a good indicator, it does not provide a common ground for comparing different $r(X)$'s. In this section, we describe a method to estimate any $r(X)$'s preference in terms of two extreme $r(X)$'s: 1) an extreme $r_{\text{abs,cvx}}(X)$ that favors the orchestrator most and 2) an extreme $r_{\text{abs,ccv}}(X)$ that favors the worker most, and through the help of the linear $r_{\text{lnr}}(X)$.

The second derivative, while it does not provide a solid common ground for comparing how an $r(X)$'s preference to another, but its sign provides a binary measure of that preference, i.e., the $r(X)$ is orchestrator-preferring if $\operatorname{sgn}(r''(X)) = -1$ and worker-preferring if $\operatorname{sgn}(r''(X)) = +1$. Our suggested metric measures how much deviation is between $r_{\text{lnr}}(X)$ and $r(X)$. This can be captured by means of the signed area enclosed between $r(X)$ and $r_{\text{lnr}}(X)$, i.e.,

$$\begin{aligned}\int_{X_{\min}}^{X_{\max}} (r(x) - r_{\text{lnr}}(x))\, dx &= \int_{X_{\min}}^{X_{\max}} r(x)\, dx - \int_{X_{\min}}^{X_{\max}} r_{\text{lnr}}(x)\, dx\\ &= R(X_{\max}) - R(X_{\min}) - R_{\text{lnr}}(X_{\max}) + R_{\text{lnr}}(X_{\min})\\ &= R(X_{\max}) - R(X_{\min}) - \frac{1}{2}(V_{\max} + V_{\min})(X_{\max} - X_{\min})\end{aligned} \tag{16}$$

where $R(x)$ is the anti-derivative of $r(x)$.

We now construct the two functions of maximal preference to each the orchestrator and the worker

$$\begin{aligned} r_{\text{abs,cvx}}(x) &= \begin{cases} V_{\max}, & x = X_{\min} \\ V_{\min}, & X_{\max} \geq x > X_{\min} \\ 0, & \text{otherwise} \end{cases} \\ r_{\text{abs,ccv}}(x) &= \begin{cases} V_{\min}, & x = X_{\max} \\ V_{\max}, & X_{\max} > x \geq X_{\min} \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{17}$$

Then, we measure the proportion of twice the area acquired in (16) and (17), since the area between the $r_{\text{lnr}}(x)$ and $r_{\text{cvx}}(x)$ is the same as $r_{\text{ccv}}(x)$ due to rotational symmetry. Thus, instead of taking the proportion of the area with respect to the triangle bounded by either $r_{\text{abs,cvx}}(x)$ or $r_{\text{abs,ccv}}(x)$ and $r_{\text{lnr}}(x)$, the same proportion can be directly acquired by taking the proportion of the area bounded by $r_{\text{cvx}}(x)$ and $r_{\text{ccv}}(x)$ to the whole rectangle bounded by $r_{\text{abs,cvx}}(x)$ and $r_{\text{abs,ccv}}(x)$. Therefore, our metric, γ becomes

$$\begin{aligned}\gamma &= 2\frac{\int_{X_{\min}}^{X_{\max}} (r(x) - r_{\text{lnr}}(x))\, dx}{\int_{X_{\min}}^{X_{\max}} (r_{\text{abs,ccv}}(x) - r_{\text{abs,cvx}}(x))\, dx}\\ &= 2\frac{R(X_{\max}) - R(X_{\min}) - \frac{1}{2}(V_{\max} + V_{\min})(X_{\max} - X_{\min})}{(V_{\max} - V_{\min})(X_{\max} - X_{\min})}\end{aligned} \tag{18}$$

where $(V_{\max} - V_{\min})(X_{\max} - X_{\min})$ is the area of the whole rectangle whose diagonal is $r_{\text{lnr}}(x)$. As such, we have a common ground for comparing the amount of preference a vacation function $r(x)$. This measurement technique is illustrated in Fig. 10. For a specific incentive-vacation family, a relationship between γ and α, $\gamma = f(\alpha)$ can be derived that would allow comparing different families' preference capacity. Thus, γ is a performance metric that can estimate the parameter α. This allows the performance analysis of the IVQ model to extend beyond vacation queueing models to any model involving incentives that is translatable to $M/\widetilde{G}/1$ model. This is possible due to the one-to-one isomorphism of the PVQ model to a modified $M/G/1$, a $M/\widetilde{G}/1$ queue. Vacations, in that sense, are an abstraction of the dynamics of the model that modify the service time. This versatility allows the IVQ to be a powerful model for performance analysis in presence of incentives.

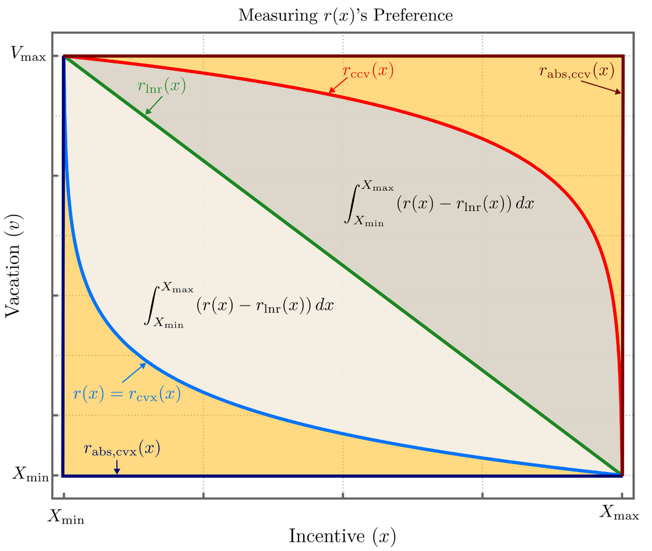


Fig. 10. Measuring preference of an IVQ function: Performance metric, γ, is the ratio of the area encompassed by the $r_{\text{cvx}}(x)$ and $r_{\text{ccv}}(x)$ to the enveloping square.

IV. Performance Analysis

A. IVQ Performance Metrics

For IVQ in the context of ECC and XEC, there are a few performance metrics that are of concern [28]. For this work, we focus on the metrics that are related to the worker. In general, we have six main parameters: 1) the service time (or equivalently service rate); 2) $S = 1/\mu$; 3) the interarrival time (or equivalently the arrival rate); 4) $\tau = 1/\lambda$; 5) the four parameters deciding $r(x)$, namely, the minimum and maximum incentive, $X_{\min}, X_{\max}$; and 6) the minimum and maximum vacation duration $V_{\min}, V_{\max}$. Knowledge of these six parameters along the first and second moments of the vacation random variable, i.e., $\mathbb{E}[V] = \mathbb{E}[r(X)]$ and $\mathbb{E}[V^2] = E[(r(X))^2]$ are sufficient to characterize the performance metrics covered in this work. We proceed to define the performance metrics in both the actual vacation variable, V, and the vacation as a proportion of the service cycle, $V_\% = V/\tau = \lambda V$. This allows us to express the relationship between the service rate and the arrival rate as a proportional relationship with a factor $(1-V_\%)$, i.e.,

$$\mu = \lambda\left(\frac{1}{1 - V_\%}\right).$$

The most fundamental metric is the average queue length [27], [28] that we express as

$$\begin{aligned}\mathbb{E}[Q_v] &= \lambda^2 \frac{2S(S + \mathbb{E}[V]) + \mathbb{E}\left[V^2\right]}{2(1-\widetilde{\rho})} + \lambda \frac{\mathbb{E}\left[V^2\right]}{2\mathbb{E}[V]} \\ &= \frac{\lambda 2S(\lambda S + \mathbb{E}[V_\%]) + \mathbb{E}\left[V_\%^2\right]}{2(1-\widetilde{\rho})} + \frac{\mathbb{E}\left[V_\%^2\right]}{2\mathbb{E}[V_\%]} \qquad (19)\end{aligned}$$

where $\widetilde{\rho} = \lambda/\mu + \lambda\mathbb{E}[V] = \lambda/\mu + \mathbb{E}[V_\%]$ in (2).

The queue length can then be used to obtain the number of jobs in the system by adding the current job being processed whose service time is equivalent to PVQ server utilization $\widetilde{\rho} = (S + \mathbb{E}[V])/\tau$, thus the average number of jobs in the system becomes

$$\begin{aligned}\mathbb{E}[L_v] &= \lambda^2 \frac{2S(S + \mathbb{E}[V]) + \mathbb{E}\left[V^2\right]}{2(1-\widetilde{\rho})} + \lambda \frac{\mathbb{E}\left[V^2\right]}{2\mathbb{E}[V]} + \widetilde{\rho} \\ &= \frac{\lambda 2S(\lambda S + \mathbb{E}[V_\%]) + \mathbb{E}\left[V_\%^2\right]}{2(1-\widetilde{\rho})} + \frac{\mathbb{E}\left[V_\%^2\right]}{2\mathbb{E}[V_\%]} + \widetilde{\rho} \\ &= \mathbb{E}[Q_v] + \widetilde{\rho}. \qquad (20)\end{aligned}$$

Consequently, the mean waiting time can also be obtained via the product of the interarrival time (which is the same as the length of the PVQ service cycle) and the queue length

$$\begin{aligned}\mathbb{E}\left[T_{Q_v}\right] &= \lambda \frac{2S(S + \mathbb{E}[V]) + \mathbb{E}\left[V^2\right]}{2(1-\widetilde{\rho})} + \frac{\mathbb{E}\left[V^2\right]}{2\mathbb{E}[V]} \\ &= \frac{2S(\lambda S + \mathbb{E}[V_\%]) + \tau\mathbb{E}\left[V_\%^2\right]}{2(1-\widetilde{\rho})} + \frac{\tau\mathbb{E}\left[V_\%^2\right]}{2\mathbb{E}[V_\%]} \\ &= \frac{\mathbb{E}[Q_v]}{\lambda} = \frac{\mathbb{E}[L_v] - \widetilde{\rho}}{\lambda}. \qquad (21)\end{aligned}$$

It is important to note that while the mean waiting time captures the job-related latency, it does not capture the customer-worker-orchestrator end-to-end latency.

The queue length can also be used to obtain the mean worker's per-job revenue, R, as

$$\mathbb{E}[R] = \frac{\mathbb{E}[X]}{\mathbb{E}[Q_v]}. \qquad (22)$$

Per-job revenue is an important metric from the worker's perspective as it allows the worker to assess the profitability of joining the ECC system. It is of concern for the orchestrator to ensure that workers would be available for recruitment, and thus allow the persistence of service provision.

The parameters of $r(x)$, i.e., $[X_{\min}, X_{\max}] \mapsto [V_{\min}, V_{\max}]$ allow us to identify the maximum and minimum throughput, respectively

$$\begin{aligned}\lambda_{\max} &= \frac{1}{V_{\min} + S} = \frac{1}{r(X_{\max}) + S} \\ \lambda_{\min} &= \frac{1}{V_{\max} + S} = \frac{1}{r(X_{\min}) + S} \qquad (23)\end{aligned}$$

which in turn is used to formulate the maximal and minimal revenue per unit time as

$$\begin{aligned}R_{\max} &= \frac{2X_{\max}(\lambda_{\max} V_{\min} - S\lambda_{\max} + 1)}{\lambda_{\max}^2(2SV_{\min} + S)} \\ &= \frac{2X_{\max}\left(V_{\%,\min} - S\lambda_{\max} + 1\right)}{\lambda_{\max}\left(2SV_{\%,\min} + S\lambda_{\max}\right)} \qquad (24)\end{aligned}$$

and

$$\begin{aligned}R_{\min} &= \frac{2X_{\min}(\lambda_{\min} V_{\max} - S\lambda_{\min} + 1)}{\lambda_{\min}^2(2SV_{\max} + S)} \\ &= \frac{2X_{\min}\left(V_{\%,\max} - S\lambda_{\min} + 1\right)}{\lambda_{\min}\left(2SV_{\%,\max} + S\lambda_{\min}\right)}. \qquad (25)\end{aligned}$$

It should be clear from and 25 that the minimal and maximal revenue are directly impacted by the choice of parameters $X_{\min}$ and $V_{\max}$. Having information about the revenue allows the worker to evaluate the benefit of remaining in the ECC or to change to another ECC. It is also of concern for the orchestrator to ensure that workers do not churn and that they would not be able to provide the service, or to prepare in advance to allocate dedicated edge and fog resources.

B. Impact of Incentive Origin in ECC

In the IVQ model, customers pay orchestrators, who then compensate workers. Workers can receive incentives through: 1) orchestrator-determined rates; 2) orchestrator-mediated matchmaking with commission; or 3) direct payment from customers in a decentralized manner (if an orchestrating entity is completely absent, this would be a decentralized XEC scenario). From the perspective of the worker, both the second and third methods are equivalent, as the TQI can be cast as $X = \sum_{i\in Q_w} x_i$ where Q_w represents the worker's queue, and x_i represents the incentive attached to the i^{th} job in the queue. It is evident that treating X on its own, as opposed to treating it as a sum of x_i's leads to different conclusions in IVQ. For readers interested in the latter case that arises in the second and third methods, we provide a brief analysis in a previous work [7].

The job incentive x_i is attached by a customer to a job they wants and passes the request to the orchestrator in the ECC. The choice of incentive stems from the valuation of the job and its completion as well as the market price. The market price, influenced by factors like supply, demand, regulations, and mechanisms, is unpredictable. These influences can be captured using game-theoretic and stochastic models, yielding a price often represented as a random variable with a general distribution at equilibrium [33]. Yet, given any underlying process, the orchestrator treats every customer equally. This aligns with the principle of indifference [34], suggesting incentives' distribution is uniform, i.e., $x_i \sim \text{Unif}(x_{\min}, x_{\max})$ for all $i \in Q_w$, where $x_{\min}$ and $x_{\max}$ represent the minimum and maximum incentive attached to a job and $\text{Unif}(a, b)$ represents a uniform distribution over the interval (a, b). In fact, x_i's can be represented in terms of the TQI by defining $X_{\min} = |Q_w|x_{\min}$ and $X_{\max} = |Q_w|x_{\max}$, where $|Q_w|$ is the number of the incentive-contributing jobs in the worker's queue. The same argument extends to the orchestrator's choice of incentives for the workers, and as such $X \sim \text{Unif}(X_{\min}, X_{\max})$ if we assume that the set of workers the orchestrator is overseeing have equivalent reputation, trust, and performance, i.e., the orchestrator has no reason to differentiate one worker from another. Without loss of generality, the performance analysis in this work is applicable for any worker in the ECC system, as such we proceed to analyze an orchestrator-origin (Uniform TQI), i.e., $X \sim \text{Unif}(X_{\min}, X_{\max})$.

C. Impact of Orchestrator Origin Incentives

In this section, we derive the first and second moments for the Uniform TQI case for both the log-family of functions, $r_{\log}(x, \alpha)$, and the rational-family, $r_{\text{rat}}(x, \alpha)$. We use the notation $\mathbb{E}_\Pi[\,\cdot\,]$ to indicate expectation over the uniform distribution for X, i.e., $\mathbb{E}_\Pi[y] = \int_{-\infty}^{\infty} y f_{X,\Pi}(x)\,dx$ where

$$f_{X,\Pi}(x) = \begin{cases} \frac{1}{X_{\max}-X_{\min}}, & X_{\min} \le x \le X_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

indicates that $X \sim f_{X,\Pi}(x) = \text{Unif}(X_{\min}, X_{\max})$.

1) Log Incentive-Vacation Family: For the log-family, i.e., $r(x) = r_{\log}(x, \alpha)$, we now list the mathematically obtained moments, starting with the first moment is formulated as

$$\begin{aligned}\mathbb{E}_\Pi\left[V_{\log}(x,\alpha)\right] &= (-\alpha)^+\mathbb{E}_\Pi\left[r_{\log,cvx}(x)\right] + (1-|\alpha|)^+\mathbb{E}_\Pi[r_{\text{lnr}}(x)] \\ &\quad + (\alpha)^+\mathbb{E}_\Pi\left[r_{\log,ccv}(x)\right]\end{aligned} \tag{27}$$

where the first moment for the convex component is

$$\mathbb{E}_\Pi\left[r_{\log,cvx}(x,\beta)\right] = \frac{\beta^{V_{\max}}(V_{\max}-V_{\min})}{\beta^{V_{\min}}-\beta^{V_{\max}}} + V_{\max} + \frac{1}{\log(\beta)} \tag{28}$$

the first moment of the linear component is

$$\mathbb{E}_\Pi[r_{\text{lnr}}(x)] = \frac{V_{\max}+V_{\min}}{2} \tag{29}$$

while the first moment of the decreasing anti-convex states as

$$\mathbb{E}_\Pi\left[r_{\log,ccv}(x,\beta)\right] = \frac{\beta^{V_{\max}}V_{\max} - V_{\min}\beta^{V_{\min}}}{\beta^{V_{\max}}-\beta^{V_{\min}}} - \frac{1}{\log(\beta)}. \tag{30}$$

We formulate the second moment for the log-family as

$$\begin{aligned}\mathbb{E}_\Pi\left[V_{\log}^2(x,\alpha,\beta)\right] &= \left((\alpha)^+\right)^2\mathbb{E}_\Pi\left[r_{\log,ccv}(x,\beta)^2\right] \\ &\quad + 2\left((\alpha)^+\right)\left((1-|\alpha|)^+\right)\mathbb{E}_\Pi\left[r_{\log,ccv}(x,\beta)r_{\text{lnr}}(x)\right] \\ &\quad + \left((-\alpha)^+\right)^2\mathbb{E}_\Pi\left[r_{\log,cvx}(x,\beta)^2\right] \\ &\quad + 2\left((-\alpha)^+\right)\left((1-|\alpha|)^+\right)\mathbb{E}_\Pi\left[r_{\log,cvx}(x,\beta)r_{\text{lnr}}(x,\alpha)\right] \\ &\quad + \left((1-|\alpha|)^+\right)^2 r_{\text{lnr}}(x)^2\end{aligned} \tag{31}$$

where the second moment for the log convex component is obtained as

$$\begin{aligned}\mathbb{E}_\Pi\left[r_{\log,cvx}^2(x,\beta)\right] &= \frac{1}{\log^2(\beta)\left(\beta^{V_{\max}}-\beta^{V_{\min}}\right)} \\ &\quad \Big(\beta^{V_{\max}}(V_{\min}\log(\beta)(V_{\min}\log(\beta)+2)+2) \\ &\quad - \beta^{V_{\min}}(V_{\max}\log(\beta)(V_{\max}\log(\beta)+2)+2)\Big)\end{aligned} \tag{32}$$

while the second moment of the linear component is

$$\mathbb{E}_\Pi\left[r_{\text{lnr}}^2(x)\right] = \frac{1}{3}\left(V_{\max}^2 + V_{\max}V_{\min} + V_{\min}^2\right) \tag{33}$$

and its correspondent second moment of the decreasing log anti-convex is

$$\begin{aligned}\mathbb{E}_\Pi\left[r_{\log,cvx}^2(x,\beta)\right] &= \frac{1}{\log^2(\beta)\left(\beta^{V_{\max}}-\beta^{V_{\min}}\right)} \\ &\quad \Big(\beta^{V_{\max}}(V_{\min}\log(\beta)(V_{\min}\log(\beta)+2)+2) \\ &\quad - \beta^{V_{\min}}(V_{\max}\log(\beta)(V_{\max}\log(\beta)+2)+2)\Big).\end{aligned} \tag{34}$$

In relation to the cross terms, the log concave-linear is obtained as

$$\mathbb{E}_{\Pi}\left[r_{\log,cvx}(x,\beta)r_{\text{lnr}}(x)\right] = \frac{1}{4\log(\beta)\left(\beta^{V_{\max}}-\beta^{V_{\min}}\right)^2}\left(-4\beta^{V_{\max}+V_{\min}}\left(\log(\beta)\left(V_{\max}^2+V_{\min}^2\right)+V_{\max}+V_{\min}\right) + \beta^{2V_{\max}}(2V_{\min}\log(\beta)(V_{\max}+V_{\min})+3V_{\max}+V_{\min}) + \beta^{2V_{\min}}(2V_{\max}\log(\beta)(V_{\max}+V_{\min})+V_{\max}+3V_{\min})\right) \tag{35}$$

and the log convex-linear cross term is

$$\mathbb{E}_{\Pi}\left[r_{\log,cvx}(x,\beta)r_{\text{lnr}}(x)\right] = \frac{1}{4\log(\beta)\left(\beta^{V_{\max}}-\beta^{V_{\min}}\right)^2}\left(-4\beta^{V_{\max}+V_{\min}}\left(\log(\beta)\left(V_{\max}^2+V_{\min}^2\right)+V_{\max}+V_{\min}\right) + \beta^{2V_{\max}}(2V_{\min}\log(\beta)(V_{\max}+V_{\min})+3V_{\max}+V_{\min}) + \beta^{2V_{\min}}(2V_{\max}\log(\beta)(V_{\max}+V_{\min})+V_{\max}+3V_{\min})\right). \tag{36}$$

Upon inspecting (28)–(36), we find that the terms constituting $\mathbb{E}_{\Pi}[V_{\log}(x,\alpha)$ are in terms of $V_{\min}$, $V_{\max}$, and the scaling parameter β. Thus, the combination of these parameters, especially $V_{\min}$ and $V_{\max}$, greatly impacts the moments of the vacation duration.

2) Rational Incentive-Vacation Family: For the rational-family, i.e., $r(x) = r_{\text{rat}}(x,\alpha)$, the first moment is of the form

$$\mathbb{E}_{\Pi}[V_{\text{rat}}(x,\alpha)] = (-\alpha)^{+}\mathbb{E}_{\Pi}\left[r_{\text{rat,cvx}}(x)\right] + (1-|\alpha|)^{+}\mathbb{E}_{\Pi}[r_{\text{lnr}}(x)] + (\alpha)^{+}\mathbb{E}_{\Pi}\left[r_{\text{rat,ccv}}(x)\right] \tag{37}$$

where the first moment for the rational convex component is

$$\mathbb{E}_{\Pi}\left[r_{\text{rat,cvx}}(x)\right] = \frac{V_{\max}V_{\min}\ln\left(\frac{V_{\max}}{V_{\min}}\right)}{V_{\max}-V_{\min}} \tag{38}$$

and first moment of the decreasing rational anti-convex is

$$\mathbb{E}_{\Pi}\left[r_{\text{rat,ccv}}(x)\right] = \frac{V_{\max}V_{\min}\ln\left(\frac{V_{\min}}{V_{\max}}\right)}{V_{\max}-V_{\min}} + V_{\max} + V_{\min}. \tag{39}$$

The second moment for the rational-family is of the form

$$\mathbb{E}_{\Pi}\left[V_{\text{rat}}^2(x,\alpha)\right] = \left((\alpha)^{+}\right)^2\mathbb{E}_{\Pi}\left[r_{\text{rat,ccv}}(x)^2\right] + 2\left((\alpha)^{+}\right)\left((1-|\alpha|)^{+}\right)\mathbb{E}_{\Pi}\left[r_{\text{rat,ccv}}(x)r_{\text{lnr}}(x)\right] + \left((-\alpha)^{+}\right)^2\mathbb{E}_{\Pi}\left[r_{\text{rat,cvx}}(x)^2\right] + 2\left((-\alpha)^{+}\right)\left((1-|\alpha|)^{+}\right)\mathbb{E}_{\Pi}\left[r_{\text{rat,cvx}}(x)r_{\text{lnr}}(x)\right] + \left((1-|\alpha|)^{+}\right)^2 r_{\text{lnr}}(x)^2 \tag{40}$$

where the second moment for the rational convex component is

$$\mathbb{E}_{\Pi}\left[r_{\text{rat,cvx}}^2(x)\right] = V_{\max}V_{\min} \tag{41}$$

second moment of the decreasing rational anti-convex is

$$\mathbb{E}_{\Pi}\left[r_{\text{rat,ccv}}^2(x)\right] = V_{\max}^2 + 3V_{\max}V_{\min} + \frac{2V_{\max}V_{\min}(V_{\max}+V_{\min})\ln\left(\frac{V_{\min}}{V_{\max}}\right)}{V_{\max}-V_{\min}} + V_{\min}^2 \tag{42}$$

for the rational concave-linear cross term is

$$\mathbb{E}_{\Pi}\left[r_{\text{rat,ccv}}(x)r_{\text{lnr}}(x)\right] = \frac{1}{2}\left(V_{\max}^2+V_{\min}^2\right) \tag{43}$$

and the rational convex-linear cross term is

$$\mathbb{E}_{\Pi}\left[r_{\text{rat,cvx}}(x)r_{\text{lnr}}(x)\right] = V_{\max}V_{\min}\left(\frac{V_{\max}+V_{\min}}{V_{\max}-V_{\min}}\ln\left(\frac{V_{\max}}{V_{\min}}\right)\right). \tag{44}$$

Similar to how the log-family moments are, the rational family's moments in (37)–(44) depend on the choice of $V_{\min}$ and $V_{\max}$ that form the bounds of the vacation duration.

With these closed-form expressions, the performance metrics in the Uniform TQI are direct substitutions of the expressions. The performance metrics, as a consequence, are directly impacted by the values of $V_{\min}$ and $V_{\max}$ agreed upon by the workers and the orchestrator, even after being filtered by $r(x,\alpha)$. The choice of $r(x,\alpha)$ to be monotonically decreasing guarantees the existence of an inverse function $X = r^{-1}(v,\alpha)$ which allows forming a bijection between vacations and incentives. This has great impact on the tractability and convergence of both $\mathbb{E}[V]$ and $\mathbb{E}[V^2]$.

V. Results and Discussion

This section provides an analysis based on the closed-form formulations obtained for an IVQ system. We look at three main performance metrics for IVQ and how they behave under the log and rational vacation families. In particular, we look at the PVQ utilization (2), the mean number of jobs in the system (20), and the mean revenue per job (22), both the log case and the rational case. We also evaluate the job satisfaction proportion with a delay threshold for a single worker. Throughout this section, we vary the minimum incentive and its corresponding maximum vacation parameter as a percentage proportion of the service cycle and regard it for different α. We analyze a scenario in which $\mu = 50$, $\lambda = 25$, $X_{\min} = 2$, $X_{\max} = 10$, $V_{\%,\min} = 0.1$, and $V_{\%,\max} = 0.5$.

In Fig. 11, the same PVQ utilization is plotted twice, once with different minimum incentive values on the x-axis, and another for the corresponding maximum vacation, $V_{\max} = r(X_{\min})$. We only look at $\alpha = \{-1, 0, 1\}$ for both families (where $\alpha = 0$ is the linear case which is the same for both families) as these values of α cover the preference range from orchestrator-favoring to worker-favoring. It can be seen in Fig. 11 that the PVQ utilization drops for higher incentives (and equivalently, increases for higher vacations). This drop is due to the fact that the PVQ utilization is the proportion of both service time and vacation to the interarrival time (i.e., $\widetilde{\rho} = (S + \mathbb{E}[V])/\tau$). Thus, for a large interarrival time τ, the PVQ utilization increases with longer vacations; otherwise, the worker would remain idle for a duration $\tau - S$ (3) written in terms of τ and S). It can also be seen that increasing

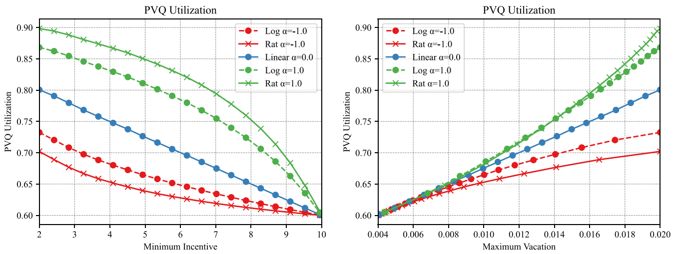


Fig. 11. Variations in PVQ utilization with incentive and vacation parameters: (a) PVQ utilization as a function of minimum incentive across different α values. (b) PVQ utilization as a function of the maximum incentive for different α values. Lowering α favors the orchestrator and lowers the PVQ utilization, showing how it varies with incentive and vacation parameters.

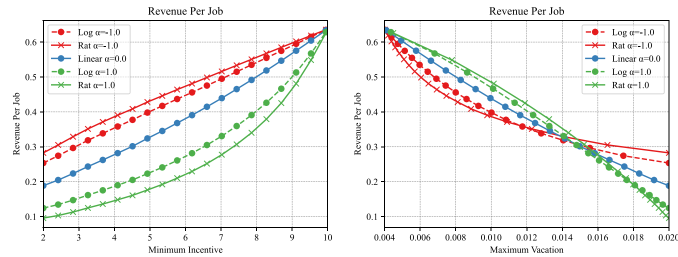


Fig. 13. Relationship between incentives, vacations, and mean revenue per job: (a) mean revenue per job as a function of minimum incentive for various α values. (b) Mean revenue per job versus maximum vacation for various α values, showing the intersection point of vacation families and the differing rates of revenue reduction across α values.

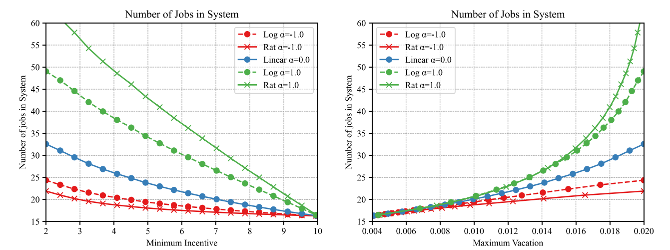


Fig. 12. Effect of incentives and vacations on the number of jobs in the system: (a) number of jobs processed as a function of minimum incentive across different α values. (b) Number of jobs processed in relation to maximum vacation across different α values, demonstrating how job quantity is influenced by varying incentive and vacation parameters.

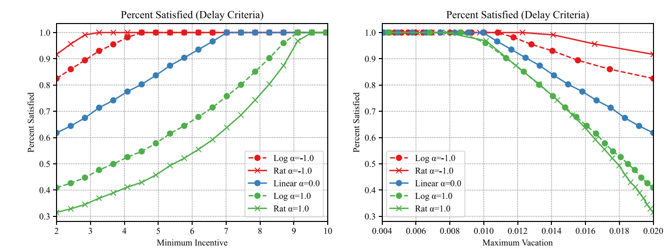


Fig. 14. Impact of incentives and vacations on job satisfaction within delay criteria: (a) proportion of jobs meeting delay criteria versus minimum incentive for different α values. (b) Proportion of jobs meeting delay criteria versus maximum vacation for different α values. Results show the thresholds at which job satisfaction is achieved for varying incentive and vacation parameters.

α causes the PVQ utilization curve to bend upward. This is due to the fact that having a low-α *penalizes* vacations, thus favoring the orchestrator and reducing the average PVQ utilization. This reflects on changing the maximum vacation as well. However, from the perspective of changing the maximum vacation it can be seen that the log and the rational families behave similarly. However, this is not the case when we look at it from the perspective of changing the minimum incentive. In Fig. 11, we can observe the variation of each family by looking at the difference between $\alpha = 1$ and $\alpha = -1$ curves and see that the log-family, compared to the rational-family, has a more conservative variation. In terms of incentive utilization, the log-family performs better (lower curve) than the rational-family for high α, and vice-versa for low α because of its convexity not being as symmetric as the rational function's since it has less vacation per unit incentive for lower incentives, i.e., the marginal cost of performance is cheaper at low incentives.

Fig. 12 shows the number of jobs the worker processed (queueing jobs and in-service). Increasing the incentives (decreasing the vacations) reduces the number of jobs in the system significantly. However, for low α in both vacation families, changing the incentives has a dampened impact for higher incentives but is somehow significant for lower incentives. This is because that for lower incentives a low α gives the orchestrators less vacation for fewer incentives. While it seems that using a lower α provides better performance, it is based on the assumption that the workers would accept the jobs at a low incentive. This is not always the case. From Fig. 12, it may seem that a higher α value causes poor performance, however for high α, workers would be more willing to sacrifice vacations for the sake of processing more jobs at the user's inconvenience to get a better profit, increasing their net service.

In Fig. 13, we now look at the mean revenue per job. Contrary to the observations from Figs. 11 and 12 which demonstrated better performance for lower α, it comes at a higher average cost per job. For high-α vacations, they come at a lower cost to the orchestrator while giving the workers more freedom in their vacations. Looking at the behavior from the perspective of the maximum vacation, we can see that longer vacations reduce the revenue per job. The rate of reduction in revenue, however, decreases for low α, and increases for high α. Moreover, a point of intersection between different vacation families can be seen when changing the maximum vacation.

To illustrate the usefulness of the metrics, we define the maximum threshold for waiting time, a deadline, of 0.8 s beyond which the job will expire and be dropped with no reward. We provide a plot of the proportion of jobs for which the delay criteria has been satisfied at different incentives and vacations in Fig. 14. It can be seen that for low incentives, there is a proportion of jobs that remain unsatisfied until a specified incentive beyond which (or equivalently, maximum vacation prior to which) all jobs are completed within their deadline. For low α, ensuring that all jobs are processed prior to their deadline occurs at a low incentive (at an overall higher incentive cost), while the high-α jobs are satisfied at a high incentive. Nevertheless, the system administrator can utilize the closed-form results obtained to identify an optimal incentive at which the target service level comprised of the different performance metrics would be satisfied for a specific α.

VI. Practical Considerations

In practice, there are some considerations when using the IVQ model. Namely, considerations regarding the choice of vacation function and how they are priced.

A. Choice of $r(X)$ and α

In an ECC system, the IVQ function can be thought of as a contract between the orchestrator and the workers. Both entities can negotiate both the choice of $r(X)$ and α. For example, an orchestrator whose service has a high cost of negative experience can decide to use the log-family, with an initial $\alpha = -1$ for any worker it recruits. As the worker grows to profit and deem the ECC system as a profitable system, and as it—from the perspective of the orchestrator—becomes more trustworthy and reliable, the value of α can then increase over time. The value of initial α in the system can also be negotiated by the worker in case it is not feasible for the worker to achieve its profitability target. As such, the orchestrator's choice of $r(X)$ is context-based. However, the space of admissible IVQ functions can searched for an optimal $r(X)$ if the context can be properly modeled.

B. Human and Device Heterogeneity

There are a number of factors that influence the incentive-vacation $r(X)$, and they need to be crafted in a manner that guarantees the system's stability and performance, as mentioned earlier. In the IVQ, we have focused on the influence of incentives on the vacation as the main factor. However, we briefly point to two other factors that impact the worker's vacation as well as overall availability. The first factor, which is the biggest source of heterogeneity, is the worker's behavior. People differ in how, when, and what they use their devices for, which impacts how long a worker device is available for, and how much resources are available. In IVQ, this translates to how much vacation can a worker device take to address the multitenancy. The second factor is heterogeneity inherent to the devices themselves as they are of different capabilities. Both factors can influence the choice of $r(X)$ and even introduce more variables and parameters to it. However, the Internet of Behaviors (IoB) can allow proper characterization of worker capabilities through the analysis of human behavior [35]. While IoB focuses on human-centric applications, the impact of IoB can extend to XEC applications.

C. Pricing of Incentives

As previously mentioned in Section IV-B, the incentive is related to the market price for computational tasks. One way to regard an IVQ system is by looking at it as a set of contracts: a contract between the customer and the orchestrator, and a contract between the orchestrator and the worker [36]. All parties in a contract seek an agreement in spite of their different objectives. For the customer, that is the service level agreement (SLA). The SLA indirectly impacts the agreement between the orchestrator and the worker, however workers can be recruited to provide similar service to another customer. As such, the risk present at the orchestrator-worker dynamic can be transferred to the customer, ultimately influencing the price. Techniques for drafting optimal contracts to maximize the service level as well as the profit [37] are crucial for the success of XEC systems.

VII. Conclusion

The rising demand for cloud services is predicted to surpass the capacity of cloud computing. To cope with such demands, paradigms like Edge and Fog computing have physically relocated service provision closer to the customer. While this proximity achieves better latency and reduces operational costs and the burden on the backhaul network, it necessitates more complex management. XEC represents a promising frontier in which user-owned devices, rich with resources, can be exploited for service provision, albeit at the cost of dealing with the uncertainty and unreliability of these devices and their owners. In this article, we describe an orchestrator-based edge system that rents multitenant devices from users to provide an edge service, offering an incentive in return. To serve both the extreme edge and their own users, we propose the IVQ model, which utilizes the XEC worker as a server that takes a vacation to perform tasks unrelated to the extreme edge, with the effect of incentives reflected in the length of these vacations. We model the behavior of systems through our IVQ model and derive closed-form expressions that relate the performance of such a P-Limited IVQ system with uniform incentives. It has been clearly demonstrated that increasing incentives enhances performance and reduces the sojourn time for jobs in the system. The IVQ model is useful for analyzing extreme edge systems where user-owned devices have the potential to become a significant part of the infrastructure, particularly in systems with an XEC orchestrator recruiting and distributing jobs to XEC workers.

Furthermore, the principles underpinning our proposed model are not confined to vacation queueing configurations alone but are applicable to a wider array of service systems that follow similar incentive-performance interplays. This indicates that our findings have implications for a diverse set of contexts where the strategic implementation of incentives is key to optimizing operational outcomes, thereby offering a generalizable approach to understanding and enhancing system performance.

References

[1] "Press release: Cloud will be the centerpiece of new digital experiences." gartner. Accessed: Oct. 30, 2022. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2021-11-10-gartner-says-cloud-will-be-the-centerpiece-of-new-digital-experiences

[2] H. Guo and J. Liu, "Collaborative computation offloading for Multiaccess edge computing over fiber–wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4514–4526, May 2018.

[3] D. Milojicic, "The edge-to-cloud continuum," *Computer*, vol. 53, no. 11, pp. 16–25, 2020.

[4] H. Tabatabaee Malazi et al., "Dynamic service placement in multi-access edge computing: A systematic literature review," *IEEE Access*, vol. 10, pp. 32639–32688, 2022.

[5] R. Meneguette, R. De Grande, J. Ueyama, G. P. R. Filho, and E. Madeira, "Vehicular edge computing: Architecture, resource management, security, and challenges," *ACM Comput. Surveys*, vol. 55, no. 1, pp. 1–46, 2021.

[6] (LF Edge, San Francisco, CA, USA). *The Home Edge Project*. (2020). [Online]. Available: https://wiki.lfedge.org/display/HOME/Home+Edge+Project

[7] S. B. Azmy, N. Zorba, and H. S. Hassanein, "Incentive-vacation queueing for extreme edge computing systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2023, pp. 94–99.

[8] E. Covi et al., "Adaptive extreme edge computing for wearable devices," *Front. Neurosci.*, vol. 15, Mar. 2021, Art. no. 611300.

[9] N. El Ioini, A. El Majjodi, D. Hastbacka, T. Cerny, and D. Taibi, "Unikernels motivations, benefits and issues: A multivocal literature review," in *Proc. 3rd Eclipse Secur. AI, Archi. Model. Conf. Cloud Edge Contin.*, 2023, pp. 39–48.

[10] A. J. Ferrer, J. M. Marques, and J. Jorba, "Ad-Hoc edge cloud: A framework for dynamic creation of edge computing infrastructures," in *Proc. 28th Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2019, pp. 1–7.

[11] T. Cai et al., "Scalable on-chain and off-chain blockchain for sharing economy in large-scale wireless networks," *IEEE Wireless Commun.*, vol. 29, no. 3, pp. 32–38, Jun. 2022.

[12] A. J. Ferrer, *Beyond Edge Computing—Swarm Computing and Ad-Hoc Edge Clouds*. Cham, Switzerland: Springer, 2023.

[13] S. Bhuyan, S. Zhao, Z. Ying, M. T. Kandemir, and C. R. Das, "End-to-end characterization of game streaming applications on mobile platforms," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 6, no. 1, pp. 1–25, 2022.

[14] C. Suraci, S. Pizzi, A. Molinaro, and G. Araniti, "MEC and D2D as enabling technologies for a secure and lightweight 6G eHealth system," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11524–11532, Jul. 2022.

[15] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "Optimizing computation offloading in satellite-UAV-served 6G IoT: A deep learning approach," *IEEE Netw.*, vol. 35, no. 4, pp. 102–108, Jul./Aug. 2021.

[16] S. B. Azmy, N. Zorba, and H. S. Hassanein, "Quality estimation for scarce scenarios within mobile crowdsensing systems," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10955–10968, Nov. 2020.

[17] C. Harth-Kitzerow and G. M. Garrido, "Verifying outsourced computation in an edge computing marketplace," 2022, *arXiv:2203.12347*.

[18] A. Zavodovski, S. Bayhan, N. Mohan, P. Zhou, W. Wong, and J. Kangasharju, "DeCloud: Truthful decentralized double auction for edge clouds," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 2157–2167.

[19] M. Siew, D. Cai, L. Li, and T. Q. Quek, "A sharing-economy inspired pricing mechanism for multi-access edge computing," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2020, pp. 1–6.

[20] Y. Zhu, B. Mao, and N. Kato, "A dynamic task scheduling strategy for multi-access edge computing in IRS-aided vehicular networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 4, pp. 1761–1771, Oct.–Dec. 2022.

[21] B. Mao, F. Tang, Z. M. Fadlullah, and N. Kato, "An intelligent route computation approach based on real-time deep learning strategy for software defined communication systems," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1554–1565, Jul.–Sep. 2021.

[22] S. Yazdani, N. Ramzan, and P. Olivier, "Enhancing edge computing with Unikernels in 6G networks," in *Proc. IEEE 34th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, 2023, pp. 1–6.

[23] H. Guo and J. Liu, "UAV-enhanced intelligent offloading for Internet of Things at the edge," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2737–2746, Apr. 2020.

[24] I. M. Amer, S. M. A. Oteafy, S. A. Elsayed, and H. S. Hassanein, "QoS-based task replication for alleviating uncertainty in edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2022, pp. 5147–5152.

[25] N. Zorba and A. Perez-Neira, "Robust power allocation schemes for Multibeam opportunistic transmission strategies under quality of service constraints," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 6, pp. 1025–1034, Aug. 2008.

[26] M. Harchol-Balter, *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[27] H. Takagi, *Queueing Analysis: Discrete-Time Systems*. Amsterdam, The Netherlands: North-Holland, 1991.

[28] N. Tian and Z. G. Zhang, *Vacation Queueing Models: Theory and Applications*. New York, NY, USA: Springer, 2006.

[29] L. Tadj and G. Choudhury, "Optimal design and control of queues," *TOP*, vol. 13, pp. 359–412, Dec. 2005.

[30] X. Zhang et al., "Incentives for mobile crowd sensing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 54–67, 1st Quart., 2016.

[31] H. Vahdat-Nejad, E. Asani, Z. Mahmoodian, and M. H. Mohseni, "Context-aware computing for mobile crowd sensing: A survey," *Future Gener. Comput. Syst.*, vol. 99, pp. 321–332, Oct. 2019.

[32] R. Imran, M. Odeh, N. Zorba, and C. Verikoukis, "Quality of experience for spatial cognitive systems within multiple antenna scenarios," *IEEE Trans. Wireless Commun.*, vol. 12, no. 8, pp. 4153–4161, Aug. 2013.

[33] M. J. Osborne and A. Rubinstein, *Models in Microeconomic Theory*. Cambridge, U.K.: Open Book Publ., 2020.

[34] E. T. Jaynes, *Probability Theory: The Logic of Science*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[35] J. Sun, W. Gan, H.-C. Chao, P. S. Yu, and W. Ding, "Internet of Behaviors: A survey," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11117–11134, Jul. 2023.

[36] R. Zeng, C. Zeng, X. Wang, B. Li, and X. Chu, "Incentive mechanisms in federated learning and a game-theoretical approach," *IEEE Netw.*, vol. 36, no. 6, pp. 229–235, Nov./Dec. 2022.

[37] J. Chen, J. Farooq, and Q. Zhu, "QoS-based contract design for profit Maximization in IoT-enabled data markets," *IEEE Internet Things J.*, vol. 10, no. 11, pp. 10080–10094, Jun. 2023.

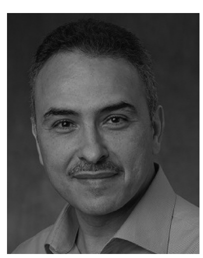
Sherif B. Azmy (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from Qatar University, Doha, Qatar, in 2017, and the M.A.Sc. degree in electrical and computer engineering from Queen's University, Kingston, ON, Canada, in 2020, where he is currently pursuing the Ph.D. degree in electrical and computer engineering.

His research interests include operations research, mobile crowdsensing, the Internet of Things, and modeling of extreme edge computing and sharing economy systems.

Nizar Zorba (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from Jordan University of Science and Technology, Irbid, Jordan, in 2002, and the Ph.D. degree in signal processing for communications from UPC Barcelona, Barcelona, Spain, in 2007.

He is currently a Professor with the Electrical Engineering Department, Qatar University, Doha, Qatar. He has authored six international patents, two books, contributed to seven book chapters, and coauthored over 160 papers in peer-reviewed journals and international conferences.

Dr. Zorba is an Area Editor of the IEEE COMMUNICATIONS LETTERS and an Associate Editor of IEEE INTERNET OF THINGS MAGAZINE and IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is also the Symposium Chair of IEEE Globecom 2023 and IEEE ICC 2023. He is a Former Chair of the IEEE Communication Society Technical Committee on Communications Systems Integration and Modeling.



Hossam S. Hassanein (Fellow, IEEE) received the B.Sc. degree in electrical engineering from Kuwait University, Kuwait City, Kuwait, in 1984, the M.Sc. degree in computer engineering from the University of Toronto, Toronto, ON, Canada, in 1986, and the Ph.D. degree in computing science from the University of Alberta, Edmonton, AB, Canada, in 1990.

He is a Leading Researcher in the areas of broadband, wireless and mobile networks architecture, protocols, control and performance evaluation. His record spans more than 600 publications in journals, conferences, and book chapters, in addition to numerous keynotes and plenary talks in flagship venues.

Dr. Hassanein has received several awards, including the 2003 Ontario Champions of Innovations Award, the 2016 IEEE Communications Society Communications Software Committee Technical Achievement Award, and the 2020 IEEE IoT, Ad Hoc, and Sensor Networks Committee Technical Achievement and Recognition Award, in recognition of his significant contributions. He is also the recipient of the 2015 Excellence in Graduate Students Supervision at Queen's University. He is the Former Chair of the IEEE Communication Society Technical Committee on IoT, AdHoc, and Sensor Networks. He is an IEEE Communications Society Distinguished Speaker.