VOLUME 24 NUMBER 5 SEPTEMBER 2018 ISSN 1387-3954

MATHEMATICAL & COMPUTER MODELLING OF DYNAMICAL SYSTEMS

Methods, tools and applications in engineering and related sciences

Editor-in-Chief: I. Troch (TU Vienna)

Taylor & Francis

Taylor & Francis
Taylor & Francis Group

Mathematical and Computer Modelling of Dynamical Systems

Methods, Tools and Applications in Engineering and Related Sciences

ISSN: 1387-3954 (Print) 1744-5051 (Online) Journal homepage: http://www.tandfonline.com/loi/nmcm20

Platform modelling and scheduling game with multiple intelligent cloud-computing pools for big data

Wanyang Dai

To cite this article: Wanyang Dai (2018) Platform modelling and scheduling game with multiple intelligent cloud-computing pools for big data, Mathematical and Computer Modelling of Dynamical Systems, 24:5, 506-552, DOI: 10.1080/13873954.2018.1516677

To link to this article: https://doi.org/10.1080/13873954.2018.1516677

Published online: 20 Sep 2018.

MATHEMATICAL AND COMPUTER MODELLING OF DYNAMICAL SYSTEMS
2018, VOL. 24, NO. 5, 506–552
https://doi.org/10.1080/13873954.2018.1516677

ARTICLE

Platform modelling and scheduling game with multiple intelligent cloud-computing pools for big data

Wanyang Dai

Department of Mathematics and State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China

ABSTRACT

We develop a generic game platform that can be used to model various real-world systems with multiple intelligent cloud-computing pools and parallel-queues for resources-competing users. Inside the platform, the software structure is modelled as Blockchain. All the users are associated with Big Data arrival streams whose random dynamics is modelled by triply stochastic renewal reward processes (TSRRPs). Each user may be served simultaneously by multiple pools while each pool with parallel-servers may also serve multi-users at the same time via smart policies in the Blockchain, e.g. a Nash equilibrium point myopically at each fixed time to a game-theoretic scheduling problem. To illustrate the effectiveness of our game platform, we model the performance measures of its internal data flow dynamics (queue length and workload processes) as reflecting diffusion with regime-switchings (RDRSs) under our scheduling policies. By RDRS models, we can prove our myopic game-theoretic policy to be an asymptotic Pareto minimal-dual-cost Nash equilibrium one globally over the whole time horizon to a randomly evolving dynamic game problem. Iterative schemes for simulating our multi-dimensional RDRS models are also developed with the support of numerical comparisons.

ARTICLE HISTORY
Received 15 June 2017
Accepted 22 August 2018

KEYWORDS
Queueing scheduling game; Pareto optimal Nash equilibrium; Big Data; Blockchain; Quantum-cloud-computing

1. Introduction

In facing with the fourth industrial revolution (Industrial Revolution 4.0, see, e.g. [1]), we develop techniques to model the random dynamics of the popular Big Data (see, e.g. [2], [3]). More importantly, we will focus on developing a generic queueing game platform driven by the Big Data and it can be used to model various real-world systems including cloud-computing or quantum-cloud-computing with multi-supercomputer centres, multi-input multi-output (MIMO) wireless channels, and Internet of Energy (IoE). Inside the platform, the software structure is modelled as Blockchain. To illustrate the effectiveness of our platform, we establish RDRS models for the performance measures of its internal data flow dynamics (i.e. queue length and workload processes) under our designed scheduling policies in order to offer services to different users in an optimal and fair way. In the meanwhile, we also develop iterative schemes

CONTACT Wanyang Dai ✉ nan5lu8@nju.edu.cn Department of Mathematics and State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210093, China

for simulating our multi-dimensional reflecting diffusion with regime-switching (RDRS) models with the support of numerical comparisons.

Industrial Revolution 4.0 is a term to describe a world where machines interact and respond intelligently to the physical environment, such as, The Internet of Things (see, e.g. [4]). In this revolution, how to effectively and intelligently deal with the so-called Big Data will be the major task. Therefore, we model the game platform with multiple service pools (centres) and parallel-queues for rate resources-competing users (game players). The game platform is supposed to live in a random environment that switches with a finite state continuous time Markov chain (FS-CTMC) (see, e.g. [5], [6], [7]). All the users' data are packetized and each packet consists of control information and a particular user's data payload. According to their nature of service requirements, these data are classified into two types: real big data and virtue big data. For a real big data service, typically in a communication system or in a network storage system with distributed database such as Blockchain (see, e.g. [8], [9] and [10]), the user's data are in the form of real data packets (or batches of packets) to be transmitted over wireless channels or wireline links. However, in a virtue big data service, the user's data packets themselves are short data messages to indicate their service requests to cloud-computing-based service centres with expensive costs and high complexities. These requests include the services to query Blockchain and run intelligent engines in FinTech, supply chains, and health-cares (see, e.g. [11], [12]).

Big Data is concerned with data sets that are so large or complex that their sizes are beyond the ability of commonly used software tools to capture, curate, manage, and process data within a tolerable elapsed time (see, e.g. [2], [3]), and the size of Big Data is a constantly moving target (see, e.g. [13]). More recently, Big Data is conceptually characterized by its three-dimensional features in De Mauro [14]: high-volume (amount of data), high-velocity (speed of data in and out), and/or high-variety (range of data types and sources). Therefore, we model the random dynamics of packet batch arrivals (flow of Big Data) from each user as a triply stochastic renewal reward process (TSRRP) with arrival rate and mean reward driven by the FS-CTMC. In other words, both packet batch's inter-arrival and service times are generally distributed with random rates while each arrival batch may contain generally distributed random number of real or virtue data packets.

Note that, the newly defined TSRRP is different from the one in Dai [5] by allowing the batch size to be random other than the unity. Due to this relaxation, the TSRRP is suitable to model Big Data. For examples, the reward (packet batch) can be used to model the high-volume amount of instantaneous real-time data, the packet inter-arrival and service times can be used to describe the high-velocity of data in and out, the different random arrival and service rates can be used to classify high-variety range of data types and sources. Furthermore, the TSRRP is a generally modulated process and covers many practical processes such as Markov modulated compound Poisson processes (MMCPPs, see, e.g. [15], [16]) used in traffic modelling of communication systems as special cases. Furthermore, it also covers the renewal process assumed by Ye and Yao [17] and Bhardwaj et al. [18], the doubly stochastic renewal process used by Dai [5], and the renewal reward processes by Whitt [19], Dai and Jiang [12] as special cases. In addition, the TSRRP is different from the correlated process for bursty arrivals assumed in some existing studies (see, e.g. Chapter 5.5 of [20]) since the inter-arrival times of our TSRRP may not be identically and independently distributed (i.i.d.) even in a fixed state of the FS-CTMC.

The parallel-queues are used to buffer data packets from their corresponding users. Each queue may be served simultaneously by multiple intelligent cloud-computing or quantum-cloud-computing service pools while each pool may also serve multiple queues at the same time via running smart policies in the Blockchain (see, e.g. Figure 1 for such an example).

However, to reflect the dynamic evolving nature of real-world systems and to realize the decentralized operation in a Blockchain, the number of pools to serve a particular queue is random and the number of queues to be served by a particular pool is also random. The service pools can be generally defined, e.g. the cloud-processors-sharing centres/resources-sharing links (see, e.g. [17]) or the MIMO channels in the undergoing wireless systems (see, e.g. [5]) and the future quantum communications (see, e.g. [21] and [22]). Especially, our study with respect to multiple pools is new to the one by Dai [5], where a single pool case is concerned.

Our dynamical rate capacity available for resource-competing users at each service pool is modelled as a randomly evolving capacity region, i.e. a high-dimensional set-valued stochastic process driven by the FS-CTMC. It is a generalized capacity region of the ones in existing studies (see, e.g. [23], [24], [5], [20]) and can capture the exact capacity variation at each time instant for time-varying channels. Based on the capacity region and if the queued data packets are considered as bids for certain pre-negotiated or pre-designed utility functions, our rate scheduling policies can be designed

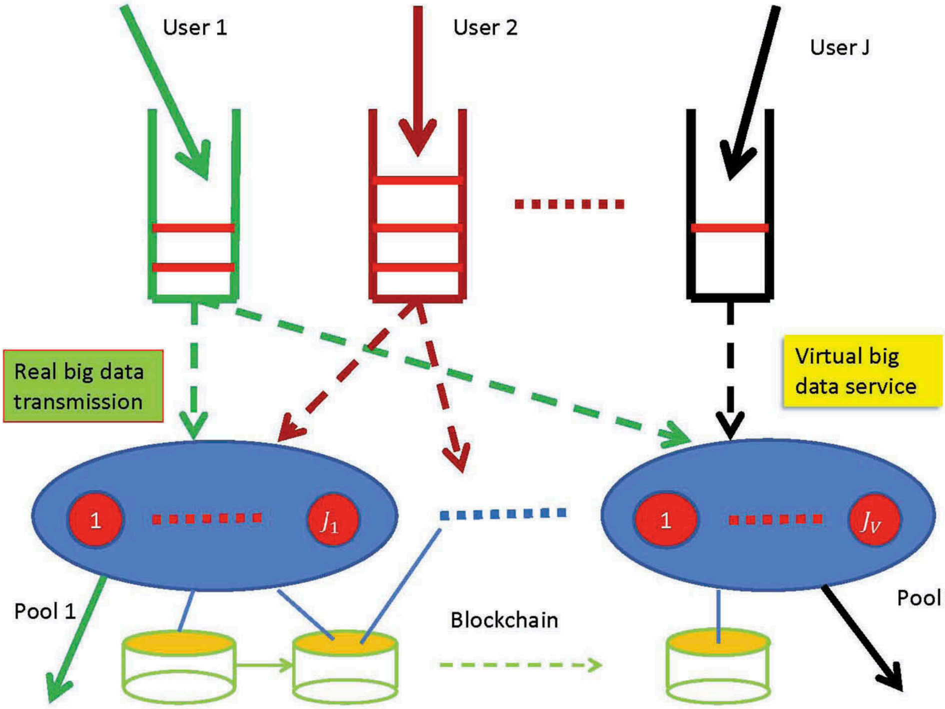

Figure 1. A game platform with parallel-queues and multiple cloud-computing service pools.

myopically at each time instant, e.g. a dynamical service rate allocation policy by a Pareto maximal-utility Nash equilibrium point myopically at each fixed time to a static game problem is designed in this paper.

Under these policies, our main objective is to model the performance measures of its internal data flow dynamics (i.e. queue length and workload processes) for the platform as RDRSs. Then, based on these RDRS models and under a so-called heavy traffic condition, if all the utility functions are strictly increasing and concave, our game-based scheduling policy is proven to be an asymptotic Pareto minimal-dual-cost Nash equilibrium one under diffusive scaling over the whole time horizon. These studies are new to the existing discussions by Nash [25], Rosen [26], Dai [5], Ye and Yao [17], Bhardwaj et al. [18], Harrison [27], etc. In supporting our main objective, we also conduct performance comparisons by developing iterative schemes to simulating multi-dimensional RDRSs. It is worth to point out that numerical methods for simulating single-dimensional RBMs are available by Asmussen et al. [28], Mousavi and Glynn [29]. Nevertheless, the schemes developed in this paper can be used to simulate multi-dimensional RDRSs or RBMs, which is new to the one by Dai [5] and even new in this area.

Finally, from the perspective of system configuration, our platform differs from the stochastic processing networks with concurrent resource occupancy (see, e.g. [27]) by adding two highlighted features. On the one hand, our whole platform is under an additional random environment driven by an FS-CTMC. On the other hand, our resource-sharing allocations are in terms of both pools and servers at the same time, where the service capacities for both pools and servers are randomly evolving and can be flexibly divided. Hence, they bring new complexity to our system modelling and scheduling policy design. Furthermore, from the perspective of network controls, we are aimed to unify the existing studies into this platform, which include the utility-maximization and Markovian decision based scheduling techniques by Dai [5], Ye and Yao [17], and Bhardwaj [18]. This type of schemes are myopic ones with the purpose to avoid directly solving high-dimensional Hamilton-Jacobi-Bellman (HJB) equations to reduce the computational complexity in online systems (see, e.g. [30]). More precisely, we make new contributions to the discussions by Dai [5] and Bhardwaj et al. [18] by developing suitable techniques for multiple service pools to replace currently available single pool based schemes. Our multi-pool-based discussions reflect the trend of cloud-computing-based services.

The remainder of the paper is organized as follows. In Section 1, we study our generic game platform model formulation with Big Data input flow pattern and Blockchain software structure, which is supported by real-world modelling applications in cloud-computing with multi-supercomputer centres, MIMO wireless channels, and Internet of Energy. In Section 2, we study the performance modelling of its internal data flow dynamics for the game platform via the RDRS models under our designed scheduling policies with the support of a simulation case study. In Section 3, we present our modelling justification, where our main results are formally proved. In Section 4, we present our conclusion of the paper. Finally, at the end of this paper, we provide an Appendix where a functional central limit theorem for TSRRPs is proved.

2. Platform model formulation

In this section, we first formulate our game platform model with Big Data input flow pattern and Blockchain software structure. Then, we provide real-world examples of our platform model in cloud-computing with multi-supercomputer centres, MIMO wireless channels, and Internet of Energy.

2.1. The platform with big data input flow pattern

The platform has V service pools (indexed by a set of positive integers $\mathcal{V} \equiv \{1, ..., V\}$) and J queues in parallel (indexed by $j \in \mathcal{J} \equiv \{1, ..., J\}$ and corresponding to J users) as shown in Figure 1. Each pool (or called centre) is equipped with J_v number of flexible parallel-servers, where v is an integer in $\mathcal{V}$. Associated with the queues, there is an J-dimensional packet arrival process $A = \left\{A(t) = (A_1(t), ..., A_J(t))', t \geq 0\right\}$, where $A_j(t)$ with $j \in \mathcal{J}$ and $t \geq 0$ is the number of packets that arrive at the jth queue during $(0, t]$. Note that, here and elsewhere in the paper, the prime denotes the transpose of a vector or a matrix. The whole platform is assumed to be driven by a stationary FS-CTMC $\alpha = \{\alpha(t), t \in [0, \infty)\}$ with a finite state space $\mathcal{K}{\equiv}\{1, ..., K\}$. The generator matrix of $\alpha(\cdot)$ is denoted by $G = (g_{il})$ with $i, l \in \mathcal{K}$, and

$$g_{il} = \begin{cases} -\gamma(i) & \text{if} \quad i = l, \\ \gamma(i)q_{il} & \text{if} \quad i \neq l, \end{cases} \tag{2.1}$$

where $\gamma(i)$ is the holding rate for the chain staying in a state $i \in \mathcal{K}$ and $Q = (q_{il})$ is the transition matrix of its embedded discrete-time Markov chain (see, e.g. [31]). Furthermore, let τ_n for each nonnegative integer $n \in \{0, 1, ...\}$ be defined by

$$\tau_0 \equiv 0, \;\; \tau_n \equiv \inf\{t > \tau_{n-1}: \alpha(t) \neq \alpha(t^-)\} \tag{2.2}$$

In other words, τ_n is a random jump time of the Markovian process $\alpha(\cdot)$.

The real-world aim of establishing the platform is to effectively offer Big Data services to different users. Therefore, how to mathematically model the Big Data arrival streams accurately according to their three-dimensional statistical feature will be the key. For this purpose, we will use TSRRPs to model the random dynamics of Big Data arrival streams. More precisely, this new traffic model can be defined by unifying the concepts of doubly stochastic renewal process (see, e.g. [5]) and renewal reward process (see, e.g. [19], [12])

Definition 2.1. *A process $A_j(\cdot)$ with $j \in \mathcal{J} \equiv \{1, ..., J\}$ is called an TSRRP if $A_j(\tau_n + \cdot)$ for each $n \in \{0, 1, ...\}$ is the counting process corresponding to a (conditional) delayed renewal reward process with arrival rate $\lambda_j(\alpha(\tau_n))$ and mean reward $m_j(\alpha(\tau_n))$ associated with finite squared coefficients of variations $\alpha_j^2(\alpha(\tau_n))$ and $\zeta_j^2(\alpha(\tau_n))$ during time interval $[\tau_n, \tau_{n+1})$.*

Note that, the inter-arrival times for an TSRRP can be correlated via another stochastic process: the external random environment $\alpha(\cdot)$. Since the inter-arrival process during each time interval $[\tau_n, \tau_{n+1})$ is a (conditional) delayed renewal reward process, the inter-arrival times may not be i.i.d. even if $\alpha(\cdot)$ takes a fixed state $i \in \mathcal{K}$ during the time interval. Thus, the TSRRP here defined in Definition 2.1 is different from the correlated process for bursty arrivals assumed in some existing studies (see, e.g. Chapter 5.5 of [20]), where the inter-arrival sequence is supposed to be i.i.d. if $\alpha(\cdot)$ takes a fixed state $i \in \mathcal{K}$ during a time interval $[\tau_n, \tau_{n+1})$. Furthermore, in our newly introduced definition of TSRRP, some other new feature is also added, i.e. at each arrival time during $[\tau_n, \tau_{n+1})$, there is an associated reward. In our application, it is interpreted as a batch with massive random number of arrival data packets and hence it can be used to model the Big Data cluster movement in a network environment. Therefore, the arrival process $A_j(\cdot)$ for each $j \in \mathcal{J}$ is supposed to be an TSRRP. In addition, we let $\{u_j(k), k = 1, 2, \ldots\}$ be the sequence of times between the arrivals of the $(k-1)$th and the kth reward batches of packets at the jth queue. The corresponding batch reward is denoted by $w_j(k)$ and all the packets arrived with it are indexed in certain successive order. Then, we can define the renewal counting process associated with the inter-arrival time sequence $\{u_j(k), k = 1, 2, \ldots\}$ for each $j \in \mathcal{J}$ by

$$N_j(t) = \sup\left\{ n \geq 0\colon \sum_{k=1}^{n} u_j(k) \leq t \right\} \tag{2.3}$$

Hence, we can present the TSRRP $A_j(\cdot)$ via

$$A_j(t) = \sum_{k=1}^{N_j(t)} w_j(k) \tag{2.4}$$

Every packet (or called job) will get service in the game platform and then leave the system. Concerning a real big data service (e.g. in a communication system or in a cloud-computing-based video-on-demand service), the packet is a real data package to be transmitted over wireless channels or wireline links. However, in a virtue big data service, the packet is a short data message to request big data job services with expensive costs and high complexities (e.g. to efficiently query big databases or to run intelligent strategic planning engines). In the former case, the packet length should be measured in bits, and in the latter case, it may be measured in time. Therefore, with this classification of the packet length measurements, we let $\{v_j(k), k = 1, 2, \ldots\}$ be the sequence of successive arrived packet lengths at queue j, which is supposed to be a sequence of strictly positive i.i.d. random variables with average packet length $1/\mu_j \in (0, \infty)$ and squared coefficient of variation $\beta_j^2 \in (0, \infty)$. In addition, we assume that all inter-arrival and service time processes are mutually (conditionally) independent when the environmental state is fixed. For each $j \in \mathcal{J}$ and each nonnegative constant h, we use $S_j(\cdot)$ to represent the renewal counting process associated with $\{v_j(k), k = 1, 2, \ldots\}$, i.e.

$$S_j(h) = \sup\left\{ n \geq 0 : \sum_{k=1}^{n} v_j(k) \leq h \right\} \tag{2.5}$$

Let $Q_j(t)$ be the jth queue length with $j \in \mathcal{J}$ at each time $t \in [0, \infty)$ and $D_j(t)$ be the number of packet departures from the jth queue in $(0, t]$. Then, the queueing dynamics governing the evolving of data in and data out in the platform can be modelled by

$$Q_j(t) = Q_j(0) + A_j(t) - D_j(t) \tag{2.6}$$

where each queue is supposed to have an infinite storage capacity to buffer real or virtue data packets (jobs) arrived for a given user. Furthermore, let $T_j(t)$ denote the cumulative amount of service given to the jth queue up to time t, i.e.

$$T_j(t) = \int_0^t \Lambda_j(Q(s), \alpha(s))ds, \tag{2.7}$$

where Λ_j for each $s \in [0, \infty]$ and $j \in \mathcal{J}$ is the summation of all service rates allocated to the jth user at time s from all possible pools and servers. Note that, Λ_j is given in a feedback control form and depends on both the current queue length $Q(s)$ and the system state $\alpha(s)$ at a time s. Thus, if we use $S_j(t)$ to denote the total number of jobs (packets) that finishes service in the system by time t, we know that $D_j(t) = S_j(T_j(t))$. Finally, we let $W(t)$ and $W_j(t)$ denote the (expected) total workload in the system at time t and the one corresponding to user j at time t, i.e.

$$W(t) = \sum_{j=1}^{J} W_j(t), \qquad W_j(t) = \frac{Q_j(t)}{\mu_j} \tag{2.8}$$

In the sequel, we will use $W(t)$ and $Q(t)$ as performance measures and design a rate scheduling policy $\Lambda = (\Lambda, \ldots \Lambda_J)$ for different service pools and servers to all the users in order that the total workload $W(t)$ and its associated total cost are minimized while the queue lengths of different users are fairly balanced.

Note that, in this platform, the available resources here are generally transformed into service rates although they can be interpreted as other forms, e.g. power in an MIMO wireless channel. Furthermore, as in a cloud-computing system or as in an MIMO Channel, we suppose that the available resources from different pools and servers can be flexibly allocated and shared between the system and users, i.e. the platform operates under a concurrent resource occupancy service regime. However, comparing with the existing studies, our allocation regime is in a more generalized cooperative manner, i.e. both jobs-to-(pools, servers) and (pools, servers)-to-jobs can be flexibly assigned, which brings new complexity to our scheduling policy design and analysis.

2.2. Modelling software structure as Blockchain

Blockchain (see, e.g. [8]) is a distributed database system and its software architecture can be presented in Figure 2.

With the emergence of IBM 50-qubit and Google 72-qubit quantum computers, it is recently evolving to the so-called Quantum-Blockchain by replacing the standard cryptographic Hash functions to quantum cryptographic Hash functions by Rajan and Visser [32]. Blockchain is widely used in Bitcoin and Ethereum applications (see,

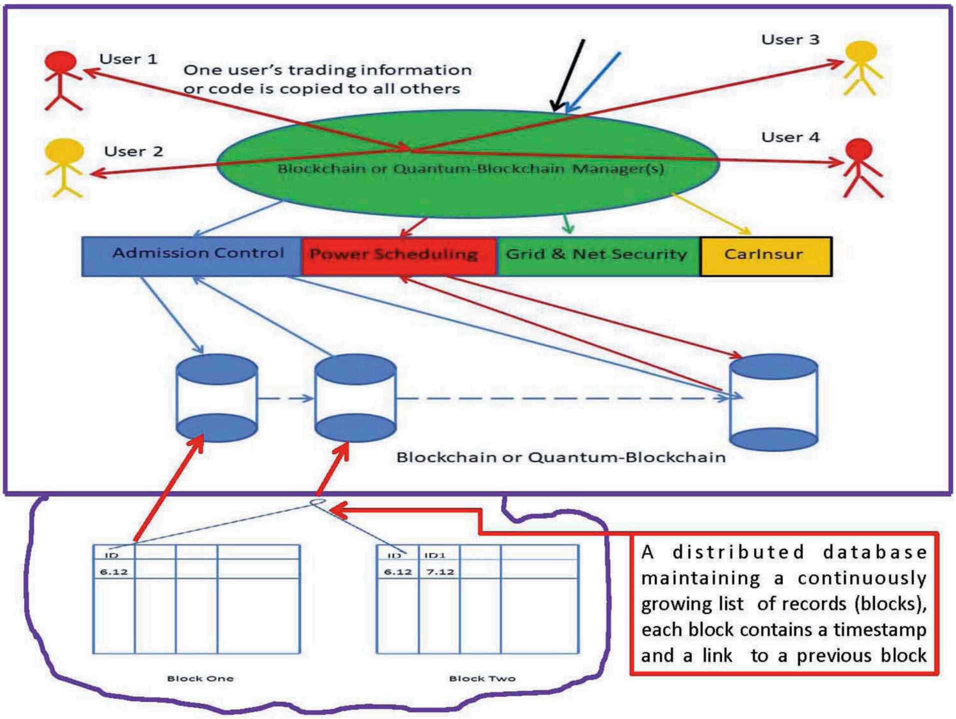
User 1
One user's trading information
or code is copied to all others
User 3
User 2
User 4
Blockchain or Quantum-Blockchain Manager(s)
Admission Control
Power Scheduling
Grid & Net Security
CarInsur
Blockchain or Quantum-Blockchain
ID
6.12
ID
ID1
6.12
7.12
Block One
Block Two
A distributed database maintaining a continuously growing list of records (blocks), each block contains a timestamp and a link to a previous block

Figure 2. A Blockchain or Quantum-Blockchain architecture.

e.g. [9] and [10]) supported with the functionalities of blockchain management, transaction generation, node communication, and block mining. Highly secured signature procedure and unchangeable data history make each transaction more safe than ever. Flexible node communications over the blockchains and smart contracts in conducting block mining make it possible for the blockchain management to be decentralized. Block mining and smart contracts can be realized by implementing policies via running various algorithms or optimization engines that are possibly aided with artificial intelligence. Therefore, owing to the consideration of system security, we are aimed to enrich the applications of blockchain to more areas, e.g. the area in cloud-computing, the area in communication networks, and the area in power and energy systems, in a unified way. In doing so, the key is about how to design efficient engines and implement them within the blockchains in realizing intelligent policies for user's admission control, resource scheduling, smart contracts, etc. as shown in Figure 2. In this paper, we will focus on designing scheduling algorithms in conducting resource allocations, e.g. bit or qubit rate allocation in a communication or quantum communication system, bitcoin mining in a blockchain, and power allocation in Internet of Energy. Note that the efficiency or optimization concerning an algorithm or a policy is in terms of the system delay, revenue, profit, cost, etc. We will model them through certain utility functions with respect to the performance measures of their internal data flow dynamics such as queue length processes and workload processes.

2.3. Real-world applications

In this subsection, we exactly map our game platform model into real-world systems including those of cloud-computing with multi-supercomputer centres, MIMO wireless channels, and Internet of Energy. The issues embedded in their corresponding Blockchains concerning resource allocations and performance modelling for our designed scheduling policies are also raised.

2.3.1. Cloud-computing with multi-supercomputer centres

Our game platform can be directly used to model the cloud-computing or quantum-cloud-computing service network with multi-supercomputer centres as shown in Figures 3–4 (where, the photos of supercomputers are enhanced from the one in Wikipedia [33]).

In this case, each service pool in our platform can be interpreted as a supercomputer or a service centre with multi-supercomputers. In this system, different users raise various service rate requirements in bits per second. However, the service capacity of each supercomputer is finite (e.g. a three user-shared capacity region is a convex set as shown in Figure 3). Furthermore, each user's data packet arrival process $A_i(\cdot)$ is random and the associated service requirements (e.g. the packet sizes) are also stochastically distributed. Therefore, how to use myopic game-theoretic scheduling policies to effectively and dynamically allocate service rates to different users according to some utility functions (e.g. the well-known proportionally fair and minimal potential ones) is

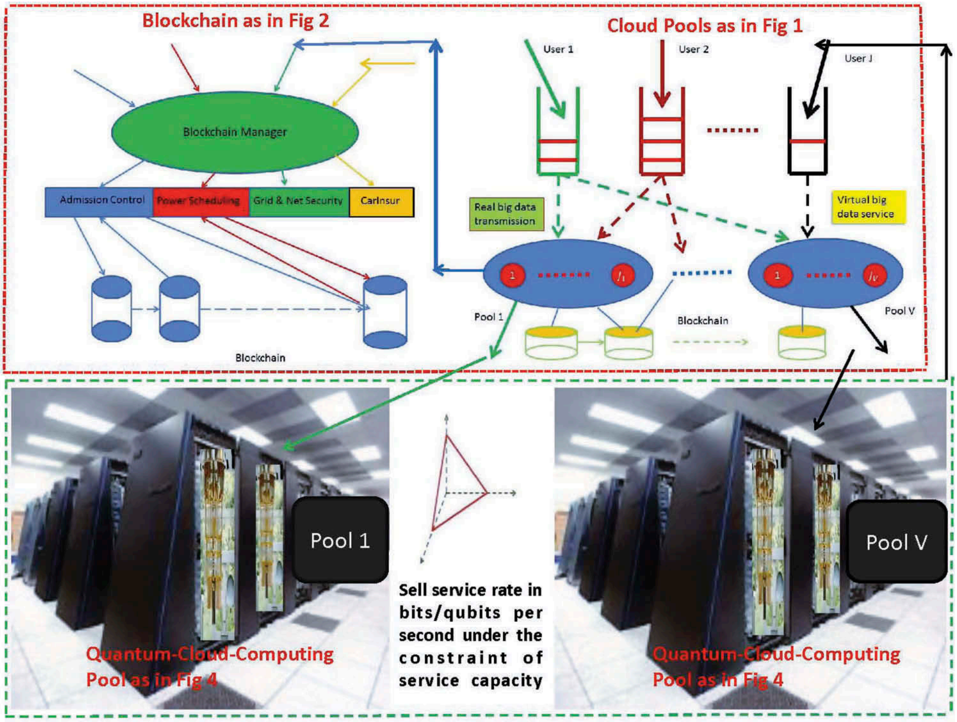

Figure 3. A game platform with parallel-queues and quantum-cloud-computing service pools.

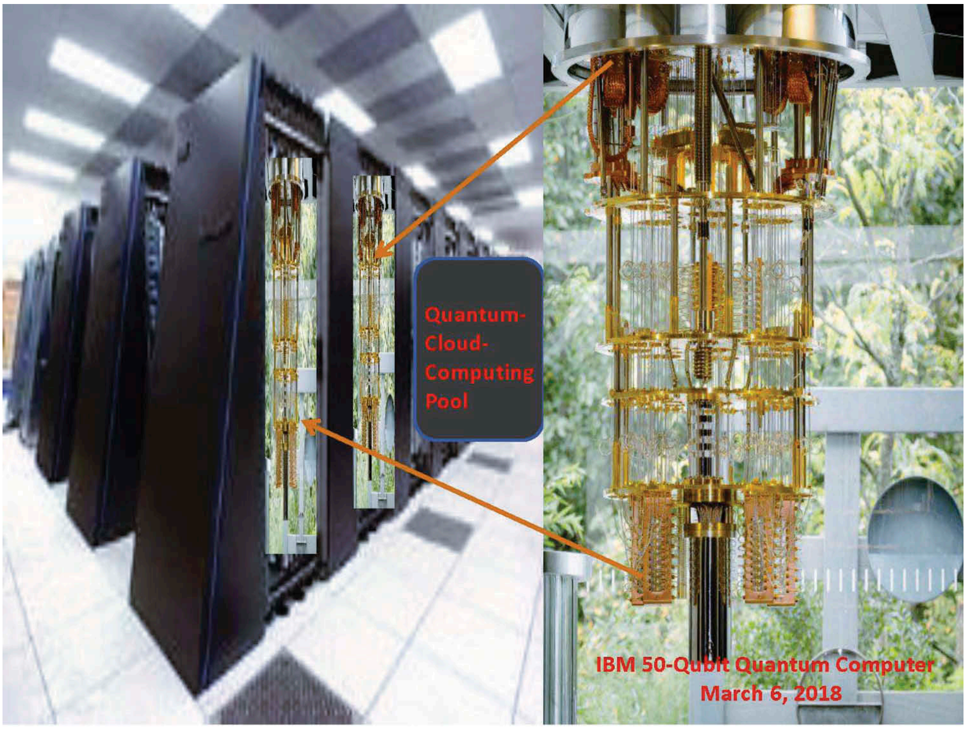

Figure 4. A quantum-cloud-computing service pool.

crucial, which will be discussed in the next section. More importantly, how to model the performance of this system's internal data flow dynamics under different rate scheduling polices will also be addressed.

2.3.2. MIMO wireless channels

MIMO technology takes a major role in the undergoing wireless systems (see, e.g. [5]) and the future quantum communications (see, e.g. [21] and [22]). In an MIMO wireless system, V transmission channels classified by different spectrums can be considered as V service pools in our game platform (see, e.g. Figures 5–6 for such an example).

To be clear, we use the cellular system as an illustrative example. In this system, base stations can cooperate among noise-free infinite-capacity links in the sense that the base stations can perform joint beamforming and/or power control, but there is a constraint on the total power that the base stations can share. Note that, we here do not make any distinction between a single-cell cellular system having multiple base-station antennas and the traditional cellular system with cooperating single-antenna base stations. Therefore, our wireless system can be considered as a base station having M antennas and J users (mobiles), each of which has N antennas. More precisely, the uplink channel can be modelled as an J-user MIMO multiple access channel (MAC) and the downlink channel can be modelled as an J-user MIMO broadcast channel (BC). Since the J-users can share the channel simultaneously, the channel can be considered as having J parallel-servers classified by different frequencies as displayed in Figure 5. Due to the

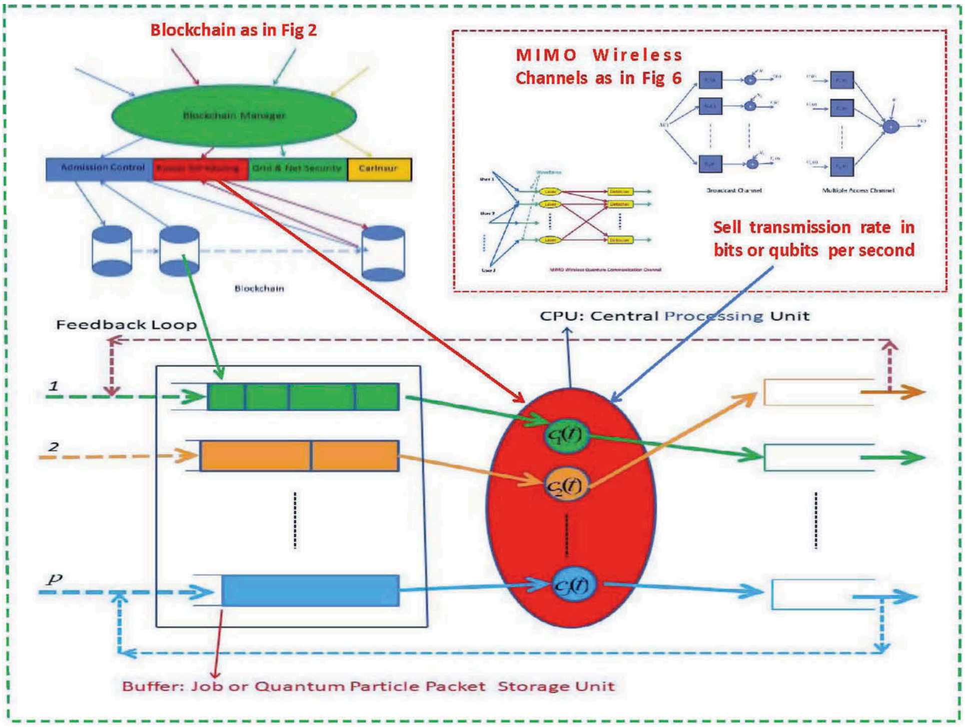
Blockchain as in Fig 2
Blockchain Manager
Admission Control
Grid & Net Security
Carinsur
Blockchain
MIMO Wireless Channels as in Fig 6
Sell transmission rate in bits or qubits per second
CPU: Central Processing Unit
Feedback Loop
1
2
p
$c_1(t)$
$c_2(t)$
$c_p(t)$
Buffer: Job or Quantum Particle Packet Storage Unit

Figure 5. A MIMO channel with parallel-queues and Blockchain.

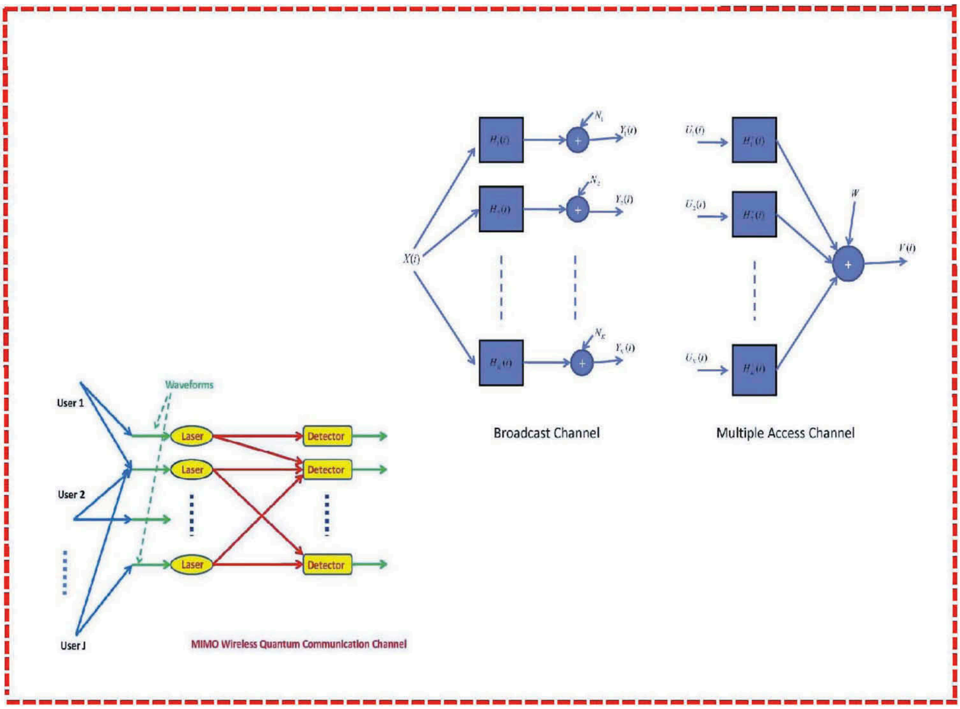

Figure 6. A MIMO wireless or quantum wireless channel.

Doppler's effect, the channel fading can be supposed to obey the stationary FS-CTMC $\alpha(\cdot)$ as defined previously. At the transmit end, arriving packets for each user are buffered before transmission and the rate of arrivals is a random process that switches with the FS-CTMC channel fading through admission control. Therefore, the processor-sharing queueing process $Q(\cdot)$ in Equation (2.6) and its associated workload process $W(\cdot)$ in Equation (2.8) can be used to model the channel dynamics of its internal data flows for both J-user MIMO MAC and J-user MIMO BC as shown in Figure 5. Owing to the total transmission power constraint, the channel capacity region can be calculated as a randomly evolving convex set process. How to use the myopic game-theoretic scheduling policy to effectively and dynamically allocate transmission rates (or equivalently transmission powers) to different users while establishing related performance models of their internal data flow dynamics under different scheduling policies will be our major concern. These will be detailed in the next section.

2.3.3. Internet of energy

Due to the fast development of today's technologies and the influence of other factors, the entire energy sector is to be restructured and turned into an intelligent and efficient supply system, i.e. the Internet of Energy (see, e.g. [34]). More precisely, it has an integrated dynamic network infrastructure and is based on standard and inter-operable communication protocols that interconnect the energy network with the Internet. It allows units of energy locally generated, stored, and forwarded to be dispatched when and where it is needed. The related information/data follows the energy flows thus implementing the necessary information exchange together with the energy transfer. In the work by Dai [35], the author presented a design about the Internet of Energy partially as shown in Figure 7.

In this integrated system, the energy usage demand (arrival) process for each user i can be described by $A_i(t)$ as defined previously. However, in this case, each arrival packet is a virtue data packet to the cloud-computing service centres. It is a service request to indicate the user's resource consuming requirements and initiate to run which smart engines within the designed blockchain, e.g. the energy scheduling or oilcoin mining algorithm. After running the smart engines, the physical energy network will provide real service to the user. Note that, the communications among the users, the power grid, and the cloud-computing service centres can be realized through the fifth generation (5G) or the future MIMO wireless channels. Therefore, our system designed in Figure 7 can also be considered as an example of Generalized Internet of Things. Furthermore, the newly added information concerning the latest oilcoin service release in the country of Venezuela is adapted from Paraskova [36]. This oilcoin is the first lawful one in the world. Here, there are three folds for us to introduce this oilcoin blockchain into our Internet of Energy. First, this oilcoin can be directly mined and traded through our game platform; second, the associated blockchain can be considered as an illustration concerning the mechanism design of dynamic decision-making, pricing, and online payments; third, it can encourage the invention and lawful endorsement of other digital coin for new energy system, e.g. new energy car system. The related resource scheduling algorithms and their corresponding performance of this system's internal data flow dynamics will be detailed in the next section.

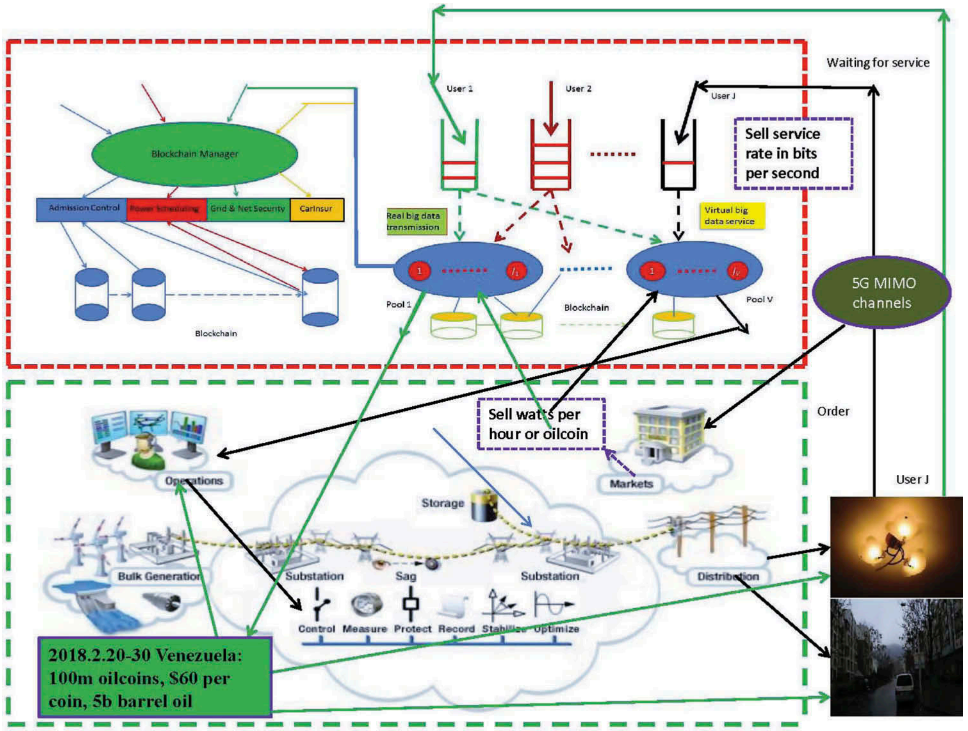

Figure 7. Virtue data packet service interactions between game platform and real energy system.

3. Performance modelling via RDRS models for scheduling policies

Although TSRRPs can accurately model Big Data arrival streams, it is difficult to directly analyse the corresponding physical queueing model in Equation (2.6) or the physical workload model in Equation (2.8) due to the non-Markovian nature of TSRRPs. Therefore, we turn to find the approximating models for $Q(t)$ and $W(t)$ in Equation (2.6) and Equation (2.8), respectively. More precisely, we model the performance measures of random dynamics under different scheduling policies by RDRS models with four folds. First, we state our main claim of performance modelling by considering our queueing system under the asymptotic regime where it is heavily loaded, i.e. under the so-called heavy traffic condition (or called load balance condition) that will be detailed in the processes of model justifications. Second, we identify and design two scheduling policies such that their performance can be modelled by the RDRS models: a myopic game-theoretical scheduling policy and an alternative scheduling policy. Third, we conduct model justifications via diffusion approximations. Fourth, we present a simulation case study to illustrate the effectiveness of RDRS modelling technique (see, e.g. the simulation results displayed in Figures 8–10 and their interpretations presented in Subsection 3.4).

3.1. Main claim and RDRS models

For each $t \geq 0$ and $j \in \mathcal{J}$, we define two sequences of diffusion-scaled processes $\hat{Q}^r(\cdot)$ and $\hat{W}^r(\cdot)$ by

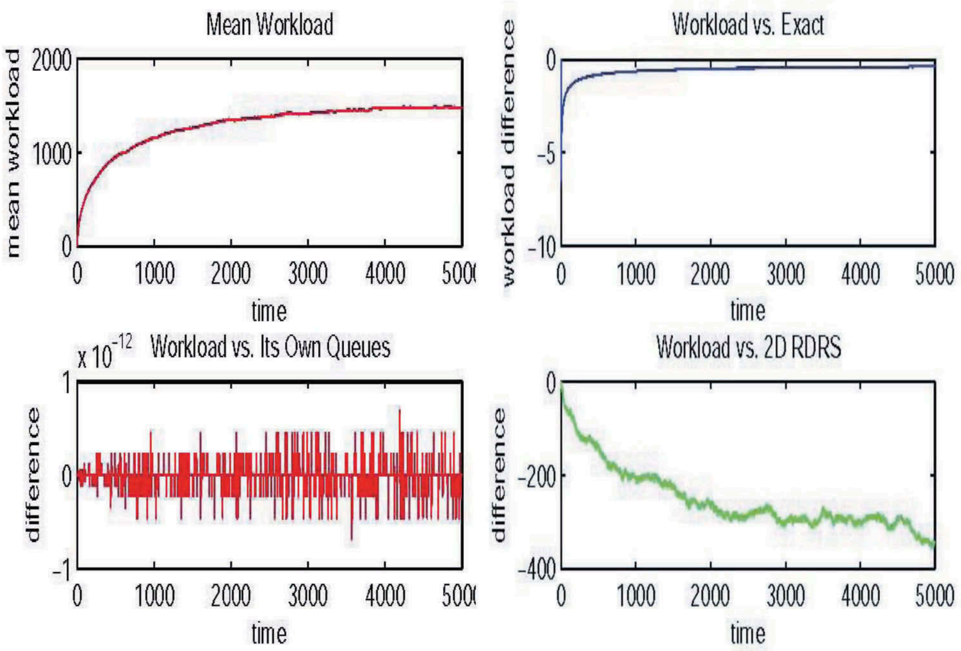

Figure 8. In this simulation, the number of simulation iterative times is $N = 6000$, the simulation time interval is $[0, T]$ with $T = 800$ or 40, which is further divided into $n = 5000$ subintervals as explained in Subsection 3.4. Other values of simulation parameters introduced in Definition 3.1 and Subsubsection 3.2.1 are as follows: $\lambda_1 = 10/3$, $\lambda_2 = 5$, $m_1 = 3$, $m_2 = 1$, $\mu_1 = 1/10$, $\mu_2 = 1/20$, $\alpha_1 = 10$, $\alpha_2 = 20$, $\beta_1 = 10$, $\beta_2 = 20$, $\zeta_1 = 1$, $\zeta_2 = 2$, $\rho_1 = \rho_2 = 1000$, $c_1^2 = c_2^1 = 1500$, $\theta_1 = -1$, $\theta_2 = -1.2$.

$$\hat{Q}_j^r(t) \equiv \frac{Q_j^r(r^2 t)}{r}, \quad \hat{W}^r(t) \equiv \frac{W^r(r^2 t)}{r} \tag{3.1}$$

where $\{r, r \in \mathcal{R}\}$ is supposed to be a strictly increasing sequence of positive real numbers and tends to infinity. Then, we can state our main claim as follows.

Claim 3.1 *Under the so-called heavy traffic condition (or called load balance condition) that is given in Section 4, the sequence of 2-tuple scaled processes in Equation (3.1) associated with the myopic game-theoretical scheduling policy and the alternative scheduling policy, which are designed in the next subsection, converges jointly in distribution, i.e.*

$$(\hat{Q}^r(\cdot), \hat{W}^r(\cdot)) \Rightarrow (\hat{Q}(\cdot), \hat{W}(\cdot)) \quad along \quad r \in \mathcal{R}, \tag{3.2.}$$

where, either $\hat{W}(\cdot))$ or $\hat{Q}(\cdot)$ is an RDRS model. Furthermore, for the myopic game-theoretical scheduling policy at each time t, the limit queue length $\hat{Q}(\cdot)$ is an asymptotic Pareto minimal-dual-cost Nash equilibrium process globally over $[0, \infty)$.

Note that, under different scheduling policies, the exact presentations of the corresponding RDRS models can be different and we will identify them explicitly in the

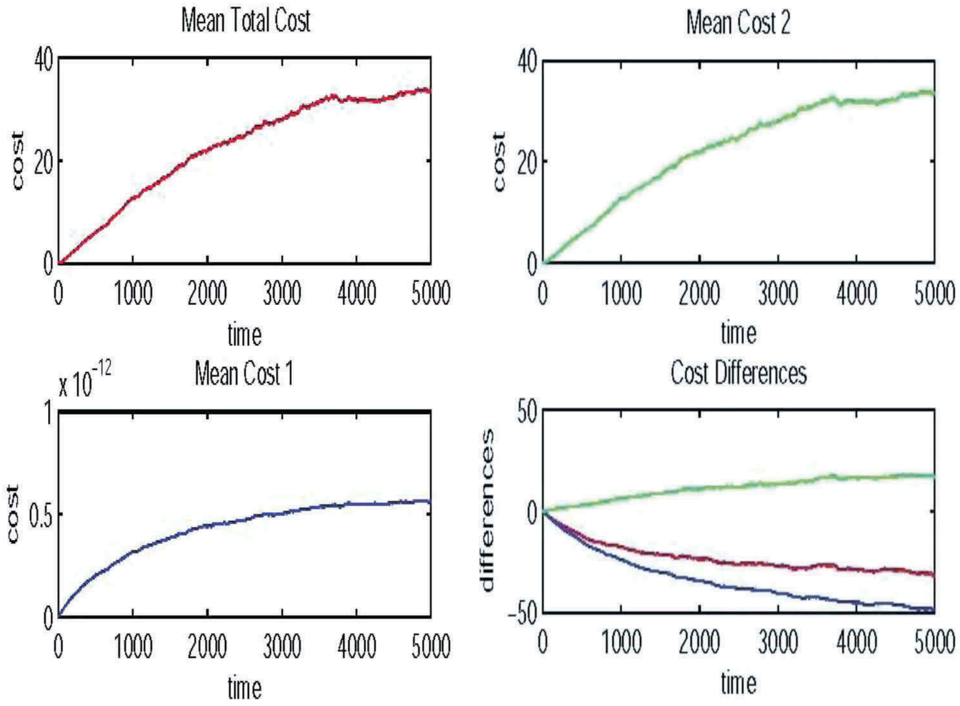

Figure 9. In this simulation, the number of simulation iterative times is $N = 6000$, the simulation time interval is $[0, T]$ with $T = 800$ or 40, which is further divided into $n = 5000$ subintervals as explained in Subsection 3.4. Other values of simulation parameters introduced in Definition 3.1 and Subsubsection 3.2.1 are as follows: $\lambda_1 = 10/3$, $\lambda_2 = 5$, $m_1 = 3$, $m_2 = 1$, $\mu_1 = 1/10$, $\mu_2 = 1/20$, $\alpha_1 = 10$, $\alpha_2 = 20$, $\beta_1 = 10$, $\beta_2 = 20$, $\zeta_1 = 1$, $\zeta_2 = 2$, $\rho_1 = \rho_2 = 1000$, $c_1^2 = c_2^1 = 1500$, $\theta_1 = -1$, $\theta_2 = -1.2$.

subsequent subsection of RDRS model identifications. Here, we first state the general definition of an RDRS model as follows.

Definition 3.1. *A u-dimensional stochastic process $\hat{Z}(\cdot)$ with $u \in \mathcal{J}$ is called an RDRS with oblique reflection if it can be uniquely represented as*

$$\hat{Z}(t) = \hat{X}(t) + \int_0^t R(\alpha(s))d\hat{Y}(s) \geq 0 \tag{3.3}$$

where

$$d\hat{X}(t) = b(\alpha(t))dt + \sigma^E dH^E(t) + \sigma^S dH^S(t) \tag{3.4}$$

Furthermore, $b(\alpha(t)) = (b_1(\alpha(t)), ..., b_u(\alpha(t))'$ is a u-dimensional vector, σ^E and σ^S are $u \times J$ matrices, and $R(\alpha(t))$ for each $t \in R_+$ is a $u \times u$ matrix. In addition, $(\hat{Z}(\cdot), \hat{Y}(\cdot))$ is continuous a.s. and is a solution of Equation (3.3) with the properties for each $j \in \{1, ..., u\}$,

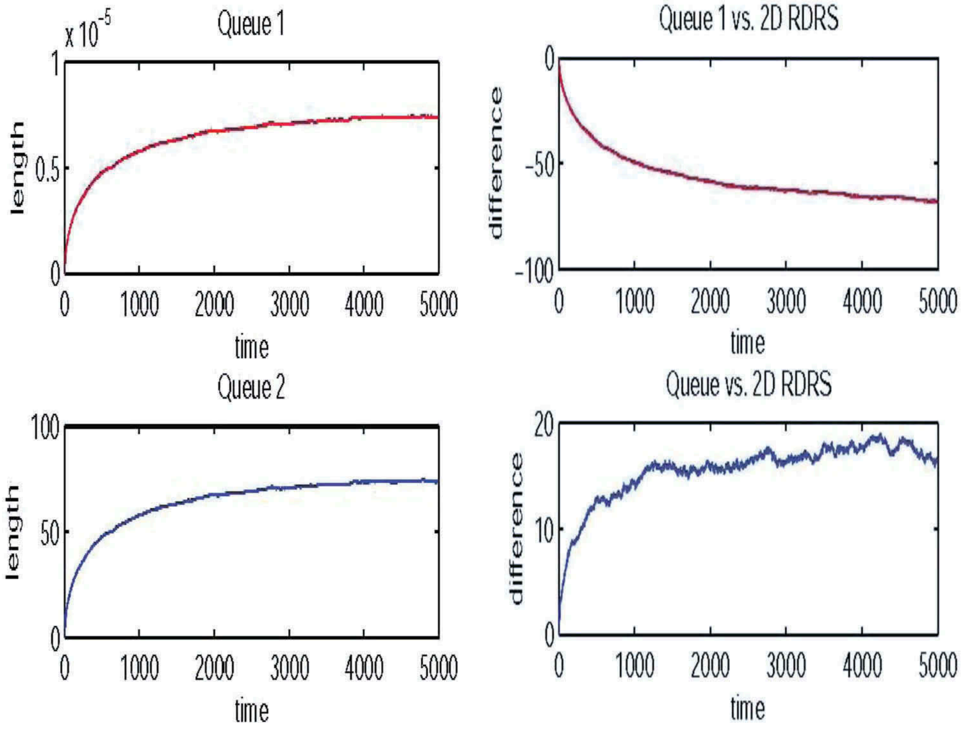

Figure 10. In this simulation, the number of simulation iterative times is $N = 6000$, the simulation time interval is $[0, T]$ with $T = 800$ or 40, which is further divided into $n = 5000$ subintervals as explained in Subsection 3.4. Other values of simulation parameters introduced in Definition 3.1 and Subsubsection 3.2.1 are as follows: $\lambda_1 = 10/3$, $\lambda_2 = 5$, $m_1 = 3$, $m_2 = 1$, $\mu_1 = 1/10$, $\mu_2 = 1/20$, $\alpha_1 = 10$, $\alpha_2 = 20$, $\beta_1 = 10$, $\beta_2 = 20$, $\zeta_1 = 1$, $\zeta_2 = 2$, $\rho_1 = \rho_2 = 1000$, $c_1^2 = c_2^1 = 1500$, $\theta_1 = -1$, $\theta_2 = -1.2$.

(1) $\hat{Y}_j(0) = 0$;
(2) *Each component* $\hat{Y}_j(\cdot)$ *of* $\hat{Y}(\cdot) = (\hat{Y}_1(\cdot), \ldots, \hat{Y}_u(\cdot))'$ *is non-decreasing;*
(3) *Each component* $\hat{Y}_j(\cdot)$ *can increase only at a time* $t \in [0, \infty)$ *that* $\hat{Z}_j(t) = 0$, i.e.

$$\int_0^\infty \hat{Z}_j(t) d\hat{Y}_j(t) = 0.$$

In addition, a solution to the RDRS in Equations (3.3)–(3.4) is called a strong solution if it is in the pathwise sense and is called a weak solution if it is in the sense of distribution.

Note that, in Definition 3.1, the processes $B^E(\cdot)$ and $B^S(\cdot)$ are, respectively, two J-dimensional standard Brownian motions, which are independent each other. For each state $i \in \mathcal{K}$, the nominal arrival rate vector $\lambda(i)$, the mean reward vector $m(i)$, the nominal throughput vector $\rho(i)$, and a constant parameter vector $\theta(i)$ are defined as follows:

$$\lambda(i) = (\lambda_1(i), \ldots, \lambda_J(i))', \tag{3.5}$$

$$m(i) = (m_1(i), ..., m_J(i))', \tag{3.6}$$

$$\rho(i) = \Big(\rho_1(i), ..., \rho_J(i)\Big), \tag{3.7}$$

$$\theta(i) = (\theta_1(i), ..., \theta_J(i))'. \tag{3.8}$$

Furthermore, the covariance matrices are defined as

$$\begin{aligned}\Gamma^E(i) &= \left(\Gamma^E_{kl}(i)\right)_{J\times J}\\ &\equiv \operatorname{diag}\left(\lambda_1(i)m_1^2(i)\zeta_1^2(i) + \lambda_1(i)m_1(i)\alpha_1^2,\right.\\ &\qquad \left. ..., \lambda_J(i)m_J^2(i)\zeta_J^2(i) + \lambda_J(i)m_J(i)\alpha_J^2\right),\end{aligned} \tag{3.9}$$

$$\begin{aligned}\Gamma^S(i) &= \left(\Gamma^S_{kl}(i)\right)_{J\times J}\\ &\equiv \operatorname{diag}\Big(\lambda_1(i)m_1(i)\beta_1^2, ..., \lambda_J(i)m_J(i)\beta_J^2\Big).\end{aligned} \tag{3.10}$$

In addition, the Itô‘s integrals in terms of the Brownian motions are defined as

$$H^e(t) = \left(H_1^e(t)', ..., H_J^e(t)\right) \quad \text{with} \quad e \in \{E, S\}, \tag{3.11}$$

$$H_j^e(t) = \int_0^t \sqrt{\Gamma^e_{jj}(\alpha(s))}dB_j^e(s). \tag{3.12}$$

Remark 3.1. *In comparing with the RBMs widely used and studied in queueing and financial literature, our RDRS model introduced in Definition 3.1 exhibits some new feature*, i.e. *it is a Markovian-modulated reflecting diffusion process. In the case of a constant environment (e.g. a quasi-static channel in a wireless system), it reduces to an RBM.*

3.2. Scheduling policies

In this subsection, we identify and design a myopic game-theoretical scheduling policy and an alternative scheduling policy for the purpose as mentioned in the previous subsection. To be simple, we begin with an illustrative example.

3.2.1. An illustrative scheduling policy example

In this subsubsection, we consider a single-pool system with two-users and hence will omit all the related pool index v for simplicity. More precisely, in Figure 11, we take $V = 1$ and $J = 2$. Furthermore, we assume that the state space of the FS-CTMC $\alpha(t)$ defined in Subsection 2.1 consists only of a single state 1, i.e. $\alpha(t) \equiv 1$ for all $t \in [0, \infty)$. In an MIMO wireless environment, such a case is corresponding to the so-called pseudo static channels (see, e.g. [5] and [18]).

The capacity region denoted by $\mathcal{R}$ is assumed to be a non-degenerate convex one confined by five boundary lines including the two ones on x-axis and y-axis as shown in Figure 11. The capacity upper bound of the region satisfies $c_1 + c_2 = 2000$. This region

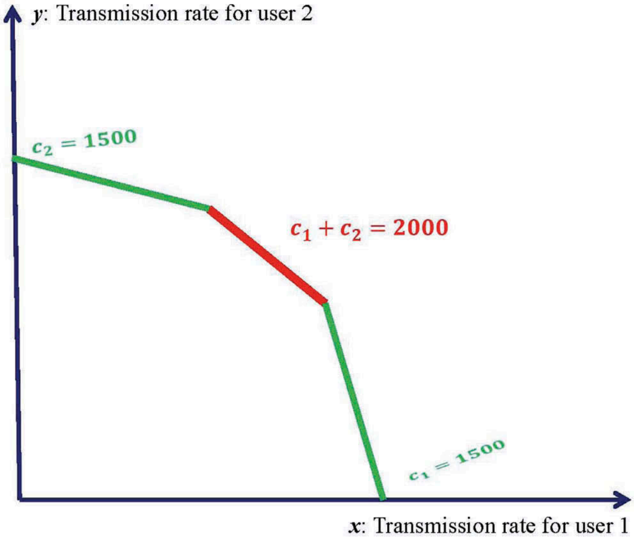
y: Transmission rate for user 2
$c_2 = 1500$
$c_1 + c_2 = 2000$
$c_1 = 1500$
x: Transmission rate for user 1

Figure 11. A 2-user capacity region.

is corresponding to a degenerate fixed MIMO wireless channel of the generally randomized one by Dai [5]. For each rate vector $c = (c_1, c_2) \in \mathcal{R}$, we take the utility functions for user 1 and user 2, respectively, by

$$U_1(q,c) = U_1(q_1,c_1) = q_1 \ln(c_1), \quad U_2(q,c) = U_2(q_2,c_2) = -\frac{q_2^2}{c_2^2}, \tag{3.13}$$

where $ln(\cdot)$ is the logarithm function with the base e. Furthermore, the vector $q = (q_1, q_2) \in [0,\infty) \times [0,\infty)$ is a given queue length of 2 users and it corresponds to the queue length process $Q(t)$ defined in Equation (2.6) at a particular time point. The utility functions U_1 and U_2 are called proportionally fair and minimal potential delay allocations, respectively, which are widely used in the design of communication protocols (see, e.g. [17]). In addition, what kind of utility functions should be used in real-world systems can frequently be negotiated and contracted among different users and the owner of our game platform. Here, the purpose for us to choose the specific forms of U_1 and U_2 is just for an illustration. Based on these utility functions, we can design our rate-scheduling policy at each time point $t \in [0,\infty)$ by a Pareto maximal-utility Nash equilibrium point to the non-zero-sum game problem

$$\max_{c \in \mathcal{R}} U_j(q,c) \quad \text{for each} \quad j \in \{0,1,2\} \quad \text{and a fixed} \quad q \in R_+^2, \tag{3.14}$$

where $U_0(q,c) = U_1(q,c) + U_2(q,c)$ and $R_+^2 = [0,\infty) \times [0,\infty)$. In other words, if $c^* = (c_1^*, c_2^*)$ is a solution to the game problem in Equation (3.14), we can conclude that

$$U_0(q,c^*) \geq U_0(q,c), \tag{3.15}$$

$$U_1(q,c^*) \geq U_1(q,c_{-1}^*) \quad \text{with} \quad c_{.-1}^* = (c_1, c_2^*), \tag{3.16}$$

$$U_2(q,c^*) \geq U_2(q,c^*_{-2}) \quad \text{with} \quad c^*_{\cdot-2} = (c^*_1, c_2). \tag{3.17}$$

Furthermore, it follows from the inequalities in Equations (3.15)–(3.17) that, if a game player's (i.e. a user's) rate service policy is unilaterally changed, his utility cannot be improved.

3.2.2. General capacity region

In our platform, the jobs in the jth queue for each $j \in \mathcal{J}$ can be served simultaneously by a random but at most V_j ($\leq V$) number of service pools at a given time point. It can be realized by processors-sharing techniques or through multiple users' and antennas' cooperation in MIMO wireless channels, i.e. base stations can perform joint beamforming and/or power control at a particular time period. Under this simultaneous service mechanism, the total service rate for the jth queue at the time point is the summation of the rates from all the pools possibly to serve the jth queue. For convenience, we index such pools by a subset $\mathcal{V}(j)$ of the set $\mathcal{V}$, i.e.

$$\mathcal{V}(j) \equiv \left\{v_{1j}, \ldots, v_{V_j j}\right\} \subseteq \mathcal{V}, \tag{3.18}$$

where v_{lj} for each $l \in \left\{1, \ldots, V_j\right\}$ indexes the v_{lj}th pool in $\mathcal{V}(j)$.

In the same way, a pool indexed by $v \in \mathcal{V}$ can possibly serve at most J_v number of job classes indexed by a subset $\mathcal{J}(v)$ of the set $\mathcal{J}$, i.e.

$$\mathcal{J}(v) \equiv \{j_{v1}, \ldots, j_{vJ_v}\} \subseteq \mathcal{J}, \tag{3.19}$$

where j_{vl} for each $l \in \{1, \ldots, J_v\}$ indexes the j_{vl}th job class in $\mathcal{J}(v)$. The pool v is equipped with J_v number of flexible parallel-servers with rate allocation vector

$$c_{v\cdot}(t) = (c_{j_{v1}}(t), \ldots, c_{j_{vJ_v}}(t))', \tag{3.20}$$

where $c_{j_{vl}}(t)$ for each $l \in \{1, \ldots, J_v\}$ is the assigned service rate to the j_{vl}th user at pool v and time t. In the sequel, we will also denote the rate $c_{j_{vl}}(t)$ by $c_{vj}(t)$ for an index $j \in \mathcal{J}(v)$ that corresponds to the $l \in \{1, \ldots, J_v\}$. The vector in Equation (3.20) takes values in a capacity region $\mathcal{R}_v(\alpha(t))$ driven by the FS-CTMC $\alpha = \{\alpha(t), t \in [0, \infty)\}$.

For each $i \in \mathcal{K}$ and $v \in \mathcal{V}$, the set $\mathcal{R}_v(i)$ is a convex region consisting of the origin and owns L_v $(>J_v)$ boundary pieces (see, e.g. the left graph of Figure 12 is an example by Dai [5], the right graph of Figure 12 is an example of Ye and Yao [17], and the detailed explanations for these two graphs are presented at the end of this subsubsection). In the region, each point is defined according to the corresponding users, i.e. $x = (x_{j_{v1}}, \ldots, x_{j_{vJ_v}})$. On the boundary of $\mathcal{R}_v(i)$ for each $i \in \mathcal{K}$, J_v of them are $(J_v - 1)$-dimensional linear facets along the coordinate axes. The other ones are located in the interior of $R^{J_v}_+$ and form the so-called *capacity surface* represented by $\mathcal{O}_v(i)$, which has $B_v = L_v - J_v\,(>0)$ linear or smooth curved facets $h_{vk}(c_{v\cdot}, i)$ on $R^{J_v}_+$ for $k \in U_v \equiv \{1, 2, \ldots, B_v\}$, i.e.

$$\mathcal{R}_v(i) \equiv \left\{c_{v\cdot} \in R^{J_v}_+ \colon\ h_{vk}(c_{v\cdot}, i) \leq 0,\ k \in \mathcal{U}_v\right\}. \tag{3.21}$$

If let $C_{U_v}(i)$ denote the sum capacity upper bound for $\mathcal{R}_v(i)$, the facet in the centre of $\mathcal{O}_v(i)$ is linear and is assumed to be a non-degenerate $(J_v - 1)$-dimensional region. More precisely, it can be expressed by

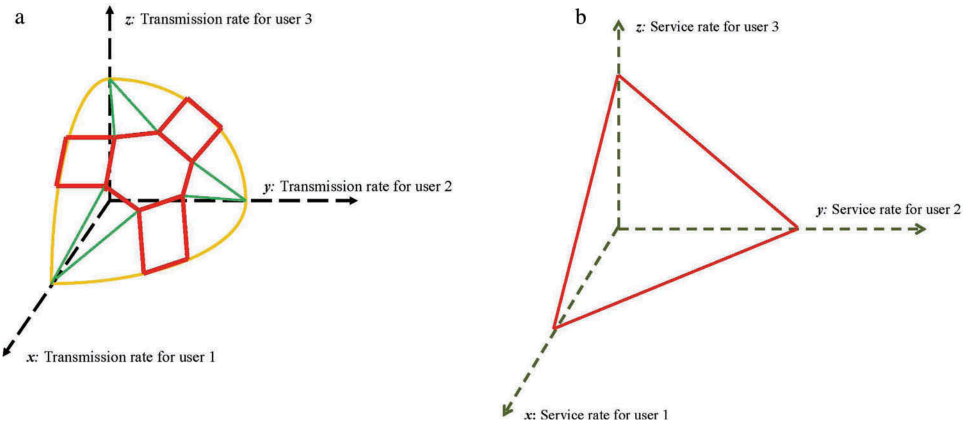
a
z: Transmission rate for user 3
y: Transmission rate for user 2
x: Transmission rate for user 1
b
z: Service rate for user 3
y: Service rate for user 2
x: Service rate for user 1

Figure 12. A 3-user capacity set in R_+^3 for a cooperative MIMO wireless channel in the left graph; a degenerate 3-user capacity set in R_+^3 for a cloud-processors-sharing system in the right graph.

$$h_{vk_{U_v}}(c_{v\cdot},i)=\sum_{j\in\mathcal{J}(v)}c_j-C_{U_v}(i), \tag{3.22}$$

where $k_{U_v}\in\mathcal{U}_v$ is the index corresponding to $C_{U_v}(i)$. Furthermore, we assume that any one of the J_v linear facets along the coordinate axes forms an (J_v-1)-user capacity region associated with a particular group of J_v-1 users when the queue corresponding to the other user is empty. By the same way, we can interpret the (J_v-l)-user capacity region for each $l\in\{2,\ldots,J_v-1\}$.

In the allocation of the service resources over the capacity regions to different users, we adopt the so-called head of line service discipline. In other words, the service goes to the packet at the head of the line for a serving queue where packets are stored in the order of their arrivals. The service rates are determined through a function of the environmental state and the number of packets in each of the queues. For each state $i\in\mathcal{K}$ and a given queue length vector $q=(q_1,\ldots,q_J)'$, let $\Lambda_{\cdot j}(q,i)$ for each $j\in\mathcal{J}$ denote the rate vector (in bps) of serving the jth queue at all its possible service pools, i.e.

$$\Lambda_{\cdot j}(q,i)=c_{\cdot j}^{\mathcal{Q}(q)}(i)=(c_{v_{1j}}^{\mathcal{Q}(q)}(i),\ldots,c_{v_{V_j}j}^{\mathcal{Q}(q)}(i)), \tag{3.23}$$

where

$$\mathcal{Q}(q)\equiv\{j\in\mathcal{J},q_j=0\}. \tag{3.24}$$

In the meanwhile, let $\Lambda_{v\cdot}(q,i)$ for each $v\in\mathcal{V}$ denote the rate vector for all the users possibly served at service pool v, i.e.

$$\Lambda_{v\cdot}(q,i)=c_{v\cdot}^{\mathcal{Q}(q)}(i)=(c_{j_{v1}}^{\mathcal{Q}(q)}(i),\ldots,c_{j_{vJ_v}}^{\mathcal{Q}(q)}(i)). \tag{3.25}$$

Obviously, if the pool index $v_{lj}\in\mathcal{V}(j)$ for an integer $l\in\{1,\ldots,V_j\}$ with $j\in\mathcal{J}$, we have that $c_{v_{lj}}^{\mathcal{Q}(q)}(i)=c_j^{\mathcal{Q}(q)}(i)$. Thus, for each $j\in\mathcal{J}$, the total rate used in Equation (2.7) can be expressed as

$$\Lambda_j(Q(s),\alpha(s)) = \sum_{v\in V(j)} c_{vj}^{\mathcal{Q}(Q(s))}(\alpha(s)). \tag{3.26}$$

Finally, we impose the convention that an empty queue should not be served. Thus, for each $v \in \mathcal{V}$ and $\mathcal{Q} \subseteq \mathcal{J}$ (e.g. a set as given by Equation (3.24)), we can define

$$c_{j_{vl}}^{\mathcal{Q}}(i) \equiv \begin{cases} =0 & \text{if} \quad j_{vl} \in \mathcal{Q} \quad \text{with} \quad l \in \{1,\ldots,J_v\}, \\ >0 & \text{if} \quad j_{vl} \notin \mathcal{Q} \quad \text{with} \quad l \in \{1,\ldots,J_v\}, \end{cases} \tag{3.27}$$

$$c_{vj}^{\mathcal{Q}}(i) \equiv c_{j_{vl}}^{\mathcal{Q}}(i) \quad \text{for some} \quad j \in \mathcal{J}(v) \quad \text{corresponding to each} \quad l \in \{1,\ldots,J_v\}, \tag{3.28}$$

$$F_{\mathcal{Q}}^{v}(i) \equiv \{x \in \mathcal{R}_v(i)\colon\ x_{j_{vl}} = 0 \quad \text{for all} \quad j_{vl} \in \mathcal{Q} \quad \text{with} \quad l \in \{1,\ldots,J_v\}\}. \tag{3.29}$$

Hence, for all $\mathcal{Q}$ such that $\emptyset \subseteq \ \mathcal{Q} \subseteq \mathcal{J}(v)$ corresponding to each $v \in \mathcal{V}$, if $c_{v\cdot}^{\mathcal{Q}}(i)$ is on the boundaries of the capacity region $\mathcal{R}_v(i)$, we have the following observation that

$$\sum_{j\in\mathcal{J}(v)} c_{vj}^{\emptyset}(i) \geq \sum_{j\in\mathcal{J}(v)} c_{vj}^{\mathcal{Q}}(i), \tag{3.30}$$

$$\sum_{j\in\mathcal{J}(v)\setminus\mathcal{Q}} c_{vj}^{\emptyset}(i) \leq \sum_{j\in\mathcal{J}(v)\setminus\mathcal{Q}} c_{vj}^{\mathcal{Q}}(i), \tag{3.31}$$

where $c_{v\cdot}^{\emptyset}(i) \in \mathcal{O}_v(i)$ and $\emptyset$ denote the empty set.

Typical examples of our capacity region include those for J-user MIMO multiple access uplink and broadcast downlink wireless channels, two-way or future quantum communication channels, and J-user cloud-processors-sharing centres or links (see, e.g. [24], [37], [5], [21], [22], [17]). In a cooperative wireless channel, the inequalities in Equations (30)–(31) are both strict, which lead to a capacity region (e.g. with three users) as displayed in the left graph of Figure 12. Note that, in this three-user case, there are 3 linear facets along the coordinate axes and 13 linear or smooth curved facets on the capacity surface. Furthermore, for such an MIMO channel, the capacity region reflects the cooperation property that the maximum of the sum of the rates is achieved only when all of the queues are non-empty. However, in a general cloud-processors-sharing service system, the equalities in Equations (3.30)–(3.31) may be both true, which leads to a degenerate capacity region as shown in the right graph of Figure 12.

3.2.3. A myopic game-theoretical scheduling policy

To dynamically realize the optimal and fair resource allocation, we design a strategy by a static Pareto maximal-utility Nash equilibrium policy myopically at each time point t to a non-zero-sum game problem for each state $i \in \mathcal{K}$ and a given queue length vector $q = (q_1,\ldots,q_J)'$. The Pareto optimality represents the full utilization of resources in the whole game system and the Nash equilibrium represents the fairness to all the users. More precisely, in this game, there are J users (players) corresponding to the J queues and each of them has his own utility function $U_{vj}(q_j, c_{vj})$ with $j \in \mathcal{J}(v)$ and $v \in \mathcal{V}(j)$. The utility functions may also depend on some additional parameters such as prices. Nevertheless, in the current paper, we assume that they are pre-negotiated and are given. Every user chooses a policy to maximize his own utility function at each service pool v while the summation of all the utility functions is also maximized. In other words, we can formulate the following generalized scheduling game problem by extending the one in Example 3.4,

$$\max_{c_{v\cdot}\in F^{v}_{\mathcal{Q}}(i),\, j\in\mathcal{J}(v)\backslash\mathcal{Q}(q)} U_{vj}(q,c)=U_{vj}(q,c^{*}(i)), \tag{3.32}$$

where the rate vector c is given by

$$c=((c_{j_{11}},\ldots,c_{j_{1J_1}}),\ldots,(c_{j_{V1}},\ldots,c_{j_{VJ_V}}))$$

and the objective functions are defined by

$$U_{00}(q,c)=\sum_{v\in\mathcal{V}(j)}\sum_{j\in\mathcal{J}(v)\backslash\mathcal{Q}(q)} U_{vj}(q_j,c_{vj}),$$

$$U_{0j}(q,c)=\sum_{v\in\mathcal{V}(j)} U_{vj}(q_j,c_{vj}) \quad \text{for each } \ j\in\mathcal{J}(v)\backslash\mathcal{Q}(q),$$

$$U_{vj}(q,c)=U_{vj}(q_j,c_{vj}) \quad \text{for each } \ j\in\mathcal{J}(v)\backslash\mathcal{Q}(q) \ \text{ and } \ v\in\mathcal{V}(j).$$

Note that, the total utility function $U_{00}(q,c)$ does not have to be a constant (e.g. zero). In other words, the game is not necessarily a zero-sum one. Thus, by unifying the concepts of Nash equilibrium and Pareto optimality in [25] and [26], we have the definition concerning a static Pareto maximal-utility Nash equilibrium policy myopically at a particular time point for the dynamic scheduling game as follows.

Definition 3.2. *For each state $i\in\mathcal{K}$ and a queue length vector $q\in R^{J}_{+}$, the rate vector*

$$c^{*}(i)\in F_{\mathcal{Q}(q)}(i)\equiv F^{1}_{\mathcal{Q}(q)}(i)\times\ldots\times F^{V}_{\mathcal{Q}(q)}(i)$$

is called a static Pareto maximal-utility Nash equilibrium policy to the non-zero-sum game problem in Equation (3.32) if it is also a maximal one to the sum of all the user's utility functions corresponding to the index in $\mathcal{J}(v)\backslash\mathcal{Q}(q)$ for each $v\in\mathcal{V}$ and no user will profit by unilaterally changing his own policy when all the other user's policy keep the same. Mathematically, for each $j\in\mathcal{J}(v)\backslash\mathcal{Q}(q)$ and any given $c(i)\in F_{\mathcal{Q}(q)}(i)$, we have that

$$U_{00}(q,c^{*}(i))\geq U_{00}(q,c(i)) \tag{3.33}$$

$$U_{vj}(q,c^{*}(i))\geq U_{vj}(q,c^{*}_{\cdot-j}(i)) \quad \textit{for} \ \ j\in\mathcal{J}(v)\backslash\mathcal{Q}(q), \ v\in\{0\}\cup\mathcal{V}(j) \tag{3.34}$$

$$c^{*}_{\cdot-j}(i)\equiv(c^{*}_{\cdot 1}(i),\ldots,c^{*}_{\cdot j-1}(i),c_{\cdot j}(i),c^{*}_{\cdot j+1}(i),\ldots,c^{*}_{\cdot J}(i)) \tag{3.35}$$

3.2.4. An alternative scheduling policy

In this subsubsection, we design an alternative scheduling policy for the purpose of performance comparisons. More precisely, we consider the case that the utility function is linear in terms of q and/or c, e.g.

$$U_{vj}(q_j, c_{vj}) = c_{vj}^{\mathcal{Q}(q)}(i) \quad \text{for} \quad j \in \mathcal{J}(v),\ v \in \mathcal{V}(j),\ i \in \mathcal{K}. \tag{3.36}$$

Therefore, the corresponding utility-maximal game in Equation (3.32) and/or its dual-cost-minimal game problem in Equation (3.41) are also linear ones. In this case, the Pareto optimal Nash equilibrium policies to these game problems may be not unique. To overcome the difficulty for this degenerate problem, we turn to select one of the Nash equilibrium points as our scheduling policy to reach certain reasonable system performance, and in the meanwhile, a multi-dimensional RDRS is established as its performance model for the corresponding limit queue length process. Note that, this simple scheduling policy will be cited as Algorithm II and the corresponding queue length process will be denoted by $Q^{II}(\cdot)$.

In communication networks and cloud-computing service practices, the prediction of customer's demands plays an important role (see, e.g. [18]). Once the demands are predicted, the relative traffic rate for each user $j \in \mathcal{J}$ is determined by a constant k_j, i.e.

$$k_j = \frac{b_j(i)}{b_1(i)} \quad \text{for all} \quad i \in \mathcal{K} \tag{3.37}$$

where $b_j(i)$ is the average bit arrival rate for $j \in \mathcal{J}$ when the channel is in state $i \in \mathcal{K}$, i.e.

$$b_j(i) = \frac{\lambda_j(i) m_j(i)}{\mu_j} \tag{3.38}$$

Then, for each $v \in \mathcal{V}(j)$ and $j \in \mathcal{J}$, the rate vector of serving the J queues is designed by

$$\Lambda_{v\cdot}(q, i) = \Lambda_{v\cdot}^{II}(q, i) = c_{v\cdot}^{\mathcal{Q}(q)}(i) \tag{3.39}$$

such that, for each $j, l \in \mathcal{J}(v) \backslash \mathcal{Q}(q) \cap \mathcal{J}(v)$ and $q \in R_+^J$,

$$\frac{c_{vj}^{\mathcal{Q}(q)}(i)}{k_j} = \frac{c_{vl}^{\mathcal{Q}(q)}(i)}{k_l} \tag{3.40}$$

Note that, for the given linear-utility functions in Equation (3.36), this designed policy $c_{v\cdot}^{\mathcal{Q}}$ for each $v \in \mathcal{V}$ and $i \in \mathcal{K}$ is located on the capacity surface $O_v(i)$. Furthermore, it is the unique Nash equilibrium policy to the utility-maximal game-based scheduling problem in Equation (3.32) subject to the constraint in Equation (3.40) for all $v \in \mathcal{V}$ and $i \in \mathcal{K}$. However, it may not be a Pareto optimal Nash equilibrium policy.

3.3. Identifications of RDRS models for different scheduling policies

In this subsection, we identify the exact expressions of RDRS models for the previously designed myopic game-theoretical scheduling policy and the alternative scheduling policy. Related simulation iterative procedures for these RDRS models are also proposed.

3.3.1. RDRS model under the game-theoretical scheduling policy

To begin with and to state our main theorem, we need to introduce another concept of the so-called static Pareto minimal-dual-cost Nash equilibrium policy myopically at a particular time point. In doing so, we formulate a minimal-dual-cost game problem corresponding to the maximal-utility game problem in Equation (3.32). More precisely, for each given $i \in \mathcal{K}$, a rate vector $c \in \mathcal{R}(i) \equiv \mathcal{R}_1(i) \times ... \times \mathcal{R}_V(i)$, and a parameter $w \geq 0$, the problem can be stated as follows:

$$\min_{q_j \in R_+, j \in \mathcal{C}(c) \cap \mathcal{J}(v)} C_{vj}(q,c) \tag{3.41}$$

subject to

$$\sum_{j \in \mathcal{C}(c)} \frac{q_j}{\mu_j} \geq w,$$

where the cost function $C_{vj}(q,c)$ for each $j \in \mathcal{J}(v)$ and $v \in \mathcal{V}(j)$ is defined by

$$C_{00}(q,c) = \sum_{v \in \mathcal{V}(j)} \sum_{j \in \mathcal{C}(c) \cap \mathcal{J}(v)} C_j(q_j, c_{vj}),$$

$$C_{0j}(q,c) = \sum_{v \in \mathcal{V}(j)} C_{vj}(q_j, c_{vj}),$$

$$C_{vj}(q,c) = C_{vj}(q_j, c_{vj}) = \frac{1}{\mu_j} \int_0^{q_j} \frac{\partial U_{vj}(u, c_{vj})}{\partial c_{vj}} du \quad \text{for} \quad j \in \mathcal{C}(c) \cap \mathcal{V}(v) \quad \text{and} \quad v \in \mathcal{V}(j),$$

and $\mathcal{C}(c)$ is an index set corresponding to the non-zero rates and non-empty queues, i.e.

$$\mathcal{C}(c) \equiv \{j\colon c_{\cdot j} \neq 0 \ \text{ componentwise with } \ j \in \mathcal{J}\}.$$

In other words, when the environment is in state $i \in \mathcal{K}$, we try to identify a queue state q corresponding to a given $c \in \mathcal{R}(i)$ and a given parameter $w \geq 0$ such that the individual user's dual-costs and the total dual-cost over the system are all minimized at the same time while the (average) workload meets or exceeds w. In addition, the total dual-cost function $U_{00}(q,c)$ does not have to be a constant. Then, we have the following definition.

Definition 3.3. *For each state $i \in \mathcal{K}$ and a rate vector $c(i) \in \mathcal{R}(i)$, the queue length vector $q^* \in R_+^J$ with $q_j^* = 0$ if $j \in \mathcal{J} \backslash \mathcal{C}(c)$ is called a static Pareto minimal-dual-cost Nash equilibrium policy to the non-zero-sum game problem in Equation (3.41) if it is also a minimal one to the sum of all the user's dual cost functions corresponding to the indices in $\mathcal{C}(c)$ and no user will profit by unilaterally changing his own policy when all the other user's policies keep the same. Mathematically, for each $j \in \mathcal{C}(c)$, $v \in \{0\} \cup \mathcal{V}$, and any given $q \in R_+^J$ with $q_j = 0$ if $j \in \mathcal{J} \backslash \mathcal{C}(c)$, we have that*

$$C_{00}(q^*, c(i)) \leq C_{00}(q, c(i)), \tag{3.42}$$

$$C_{vj}(q^*, c(i)) \leq C_{vj}(q_{-j}^*, c(i)), \tag{3.43}$$

$$q_{-j}^* \equiv (q_1^*, ..., q_{j-1}^*, q_j, q_{j+1}^*, ..., q_J^*). \tag{3.44}$$

Based on Definition 3.3 and the concept concerning the asymptotic optimality widely used in heavy traffic analysis (see, e.g. [17], [5]), we can develop some new concept about the asymptotic Pareto minimal-dual-cost Nash equilibrium policy as follows.

Definition 3.4. *Let $\hat{Q}^{G,r}(\cdot)$ and $\hat{W}^{G,r}(\cdot)$ denote the diffusion-scaled queue length and workload processes respectively under an arbitrarily feasible rate scheduling policy G. A process $\hat{Q}(\cdot)$ is called an asymptotic Pareto minimal-dual-cost Nash equilibrium policy globally over the whole time horizon if*

$$\liminf_{r\to\infty} C_{00}(\hat{Q}^{r,G}(t),\rho_j(\alpha(t))) \geq C_{00}(\hat{Q}(t),\rho_j(\alpha(t))), \tag{3.45}$$

$$\liminf_{r\to\infty} C_{vj}(\hat{Q}^{r,G}_{-j}(t),\rho_j(\alpha(t))) \geq C_{vj}(\hat{Q}(t),\rho_j(\alpha(t))) \tag{3.46}$$

for any $t \geq 0$, $j \in \mathcal{J}(v)$, $v \in \{0\} \cup \mathcal{V}(j)$, and

$$\hat{Q}^{r,G}_{-j}(t) = (\hat{Q}_1(t),\dots,\hat{Q}^{r,G}_j(t),\dots,\hat{Q}_J(t)) \tag{3.47}$$

Now, let $q^*(w,\rho(i))$ be the Pareto minimal-dual-cost Nash equilibrium policy to the game problem in Equation (3.41) with respect to each given w and $i \in \mathcal{K}$ at time t. Then, we can present our main theorem as follows.

Theorem 3.1. For the game scheduling policy determined by Equation (3.32) with $Q^r(0) = 0$ and conditions Equations (4.7)–(4.12) (that will be detailed in Section 4), the sequence of 2-tuple processes converges jointly in distribution, *i.e.*

$$(\hat{Q}^r(\cdot),\hat{W}^r(\cdot)) \Rightarrow (\hat{Q}(\cdot),\hat{W}(\cdot)) \quad along \quad r \in \mathcal{R} \tag{3.48}$$

Furthermore, the limit queue length $\hat{Q}(\cdot)$ and total workload $\hat{W}(\cdot)$ are related each other through

$$\hat{Q}(t) = q^*(\hat{W}(t),\rho(\alpha(t))) \tag{3.49}$$

where $\hat{W}(\cdot)$ is a 1-dimensional RDRS in strong sense with

$$b(i) = \theta_1(i)/\mu_1 + \cdots + \theta_J(i)/\mu_J \tag{3.50}$$

$$\sigma^E = \sigma^S = \left(1/\mu_1,\dots,1/\mu_J\right), \tag{3.51}$$

$$R(i) = 1 \tag{3.52}$$

In addition, there is a common supporting probability space, under which and with probability one, the limit queue length $\hat{Q}(\cdot)$ is an asymptotic Pareto minimal-dual-cost Nash equilibrium policy globally over time interval $[0,\infty)$. Finally, the limit workload $\hat{W}(\cdot)$ is also asymptotic minimal in the sense that

$$\liminf_{r\to\infty} \hat{W}^{r,G}(t) \geq \hat{W}(t) \tag{3.53}$$

The proof of Theorem 3.1 will be provided in Subsection 4.2. Instead, we here design a simulation procedure for the RDRS in the theorem. More precisely, for a constant $T\in[0,\infty)$, we divide the interval $[0,T]$ equally into n subintervals $\{[t_i,t_{i+1}], i\in\{0,1,\ldots,n-1\}\}$ with $t_0=0$, $t_n=T$, and $\Delta t_i=t_{i+1}-t_i=\frac{T}{n}$. Furthermore, let

$$\Delta F(t_i)\equiv F(t_i)-F(t_{i-1}) \tag{3.54}$$

for each process $F(\cdot)\in\left\{B^E(\cdot),B^S(\cdot),\hat{W}(\cdot),\hat{Y}(\cdot)\right\}$. Then, we can present our iterative simulation procedure as follows.

Simulation Procedure 3.1 *First of all, we endorse initial values to the related processes,*

$$\hat{W}(0)=\hat{Y}(0)=\hat{Q}(0)=0. \tag{3.55}$$

Then, for each $i\in\{1,\ldots,n\}$, the remaining procedure consists of the following 6 steps:

Step 1. Calculating the netput value $\hat{V}(t_i)$ at time t_i after obtaining the value $\hat{Y}(t_{i-1})$ of the regulating process $\hat{Y}(t)$ at time t_{i-1}, i.e.

$$\begin{aligned}\hat{V}(t_i)=\hat{W}(t_{i-1})+\hat{Y}(t_{i-1})+\hat{b}(\alpha(t_i))\Delta t_i+\hat{\sigma}^E(\alpha(t_i))\Gamma^E(\alpha(t_i))\Delta B^E(t_i)\\ +\hat{\sigma}^S(\alpha(t_i))\Gamma^S(\alpha(t_i))\Delta B^S(t_i).\end{aligned} \tag{3.56}$$

Step 2. Determining the incremental value $\Delta\hat{Y}(t_i)$ of the regulating process $\hat{Y}(t)$ over time interval $[t_{i-1},t_i]$ according to the sign of $\hat{V}(t_i)$, i.e.

$$\Delta\hat{Y}(t_i)=\begin{cases}0 & \text{if} \quad \hat{V}(t_i)>0,\\ -\hat{V}(t_i) & \text{if} \quad \hat{V}(t_i)\le 0.\end{cases} \tag{3.57}$$

Step 3. Computing the value $\hat{W}(t_i)$ of the workload process $\hat{W}(t)$ at time t_i, i.e.

$$\hat{W}(t_i)=\hat{V}(t_i)+\Delta\hat{Y}(t_i). \tag{3.58}$$

Step 4. Obtaining the value $\hat{Q}(t_i)$ of the queue length process $\hat{Q}(t)$ at time t_i through Pareto minimal-dual-cost Nash equilibrium point, i.e.

$$\hat{Q}(t_i)=q^*(\hat{W}(t_i),\rho(\alpha(t_i))). \tag{3.59}$$

Step 5. Calculating the cost values corresponding to different users at time t_i, i.e.

$$\hat{C}_k(t_i)=C_k(\hat{Q}_k(t_i),\rho_k(\alpha(t_i))),\ k\in\{1,\ldots,K\}. \tag{3.60}$$

Step 6. Letting i = i + 1 and repeating the procedure in Step 1 to Step 5.

3.3.2. RDRS model under the alternative scheduling policy

As a related study of the one presented in the previous subsubsection, we have the following corollary for the alternative scheduling policy.

Corollary 3.2 *The weak convergence holds for Algorithm II under the conditions in Equations (4.7)–(4.8) and (4.12) (that will be detailed in Section 4) if the inequalities in Equations (3.30)–(3.31) are both strict and $Q^{II,r}(0)=0$*, i.e.

$$\hat{\mathcal{Q}}^{II,r}(\cdot) \Rightarrow \hat{\mathcal{Q}}^{II}(\cdot) \quad along \quad r \in \mathcal{R}, \tag{3.61}$$

where $\hat{\mathcal{Q}}^{II}(\cdot)$ is an J-dimensional RDRS in strong sense with

$$b(i) = (\theta_1(i), ..., \theta_J(i))' \qquad for\ each \quad i \in \mathcal{K}, \tag{3.62}$$

$$\sigma^E = \sigma^S = I_{J\times J} \qquad (an \ \ J \times J \ \ unit\ matrix), \tag{3.63}$$

$$R(i) = \begin{pmatrix} 1 & R_{12}(i) & ... & R_{1J}(i) \\ R_{21}(i) & 1 & ... & R_{2J}(i) \\ . & . & ... & . \\ . & . & ... & . \\ . & . & ... & . \\ R_{J1}(i) & R_{J2}(i) & ... & 1 \end{pmatrix}, \tag{3.64}$$

$$R_{jk}(i) = \frac{c_j^{\emptyset}(i) - c_j^{\{k\}}(i)}{c_k(i)} \quad for\ k{\neq}j \quad and \ \ each \ \ i \in k, \ (3.65) \tag{3.65}$$

$$c_j^{\{k\}}(i) = \sum_{v\in\mathcal{V}(j)} c_{vj}^{\{k\}}(i) \qquad and \ \ c_{vj}^{\{k\}}(i) \ \ is\ defined\ in \ \ (3.28) \ . \tag{3.66}$$

Note that, the proof of Corollary 3.2 can be conducted by combining Theorem 3.1 and its proof and the techniques used by Bhardwaj et al. [18]. However, due to the length limitation of this paper, we omit its detail here. Instead, for the purpose of numerical comparisons to the simulation case study in Subsection 3.4, we here present a simulation procedure corresponding to the two-dimensional RDRS model in the corollary as follows.

Simulation Procedure 3.2 *First of all, we endorse initial values to the related processes,*

$$\hat{\mathcal{Q}}^{II}(0) = \hat{Y}^{II}(0) = 0. \tag{3.67}$$

Then, for each $i \in \{1, ..., n\}$, the remaining procedure consists of the following 6 steps:

Step 1. Calculating the netput vector value $\hat{V}^{II}(t_i)$ at time t_i after obtaining the vector value $\hat{Y}^{II}(t_{i-1})$ of the regulating vector process $\hat{Y}^{II}(t)$ at time t_{i-1}, i.e.

$$\begin{aligned} \hat{V}^{II}(t_i) = \hat{\mathcal{Q}}^{II}(t_{i-1}) + R(\alpha(t_i))\hat{Y}^{II}(t_{i-1}) + b(\alpha(t_i))\Delta t_i \\ +\sigma^E(\alpha(t_i))\Gamma^E(\alpha(t_i))\Delta B^E(t_i) + \sigma^S(\alpha(t_i))\Gamma^S(\alpha(t_i))\Delta B^S(t_i). \end{aligned} \tag{3.68}$$

Step 2. Determining the incremental vector value $\Delta\hat{Y}^{II}(t_i)$ of the regulating process $\hat{Y}^{II}(t)$ over time interval $[t_{i-1}, t_i]$ according to the sign of each component in the vector $\hat{V}^{II}(t_i)$, i.e.

$$\Delta\hat{Y}^{II}(t_i) = \begin{cases} 0 & if \quad \hat{V}^{II}(t_i) > 0, \\ -R^{-1}(\alpha(t_i))\hat{V}^{II}(t_i) & if \quad \hat{V}^{II}(t_i) \leq 0, \\ (0, -\hat{V}_2^{II}(t_i)/R_{22}(\alpha(t_i)))' & if \quad \hat{V}_1^{II}(t_i) > 0,\ \hat{V}_2^{II}(t_i) \leq 0, \\ (-\hat{V}_1^{II}(t_i)/R_{11}(\alpha(t_i)), 0)' & if \quad \hat{V}_1^{II}(t_i) \leq 0,\ \hat{V}_2^{II}(t_i) > 0. \end{cases} \tag{3.69}$$

Step 3. Obtaining the vector value $\hat{Q}^{II}(t_i)$ of queue length process $\hat{Q}^{II}(t)$ at time t_i, i.e.

$$\hat{Q}^{II}(t_i) = \hat{V}^{II}(t_i) + R(\alpha(t_i))\Delta\hat{Y}^{II}(t_i). \tag{3.70}$$

Step 4. Getting the value $\hat{W}^{II}(t_i)$ of the workload process $\hat{W}^{II}(t)$ at time t_i, i.e.

$$\hat{W}^{II}(t_i) = \frac{1}{\mu_1}\hat{Q}_1^{II}(t_i) + \frac{1}{\mu_2}Q_2(t_i). \tag{3.71}$$

Step 5. Calculating the cost values corresponding to different users at time t_i, i.e.

$$C_k(t_i) = C_k(Q_k^{II}(t_i), c_k(\alpha(t_i))),\ k \in \{1, 2\}. \tag{3.72}$$

Step 6. Letting i = i + 1 and repeating the procedure in Step 1 to Step 5.

3.4. A simulation case study via RDRS models

In this subsection, we consider a single-pool system with two-users as presented in subsubsection 3.2.1 and hence will omit all the related pool index v for simplicity. Note that, in a corresponding real-world system, the parameter vector q in Equation (3.14) is the randomly evolving queue length process $Q(t)$ in Equation (2.6). How to use the RDRS performance model in Definition 3.1 to evaluate the effectiveness of the myopic service policy designed in Equation (3.14) globally over the whole time horizon $[0, \infty)$ is our concern. To reach this goal, we first identify the associated dual-cost functions $C_j(q, c)$ as defined in Equation (3.41) with $j \in \{1, 2\}$ for corresponding $U_j(q, c)$ given in Equation (3.13). More precisely,

$$C_1(q_1, c_1) = \frac{1}{\mu_1}\int_0^{q_1} \frac{\partial U_1(u, c_1)}{\partial c_1} du \quad = \quad \frac{q_1^2}{2\mu_1 c_1}, \tag{3.73}$$

$$C_2(q_2, c_2) = \frac{1}{\mu_2}\int_0^{q_2} \frac{\partial U_2(u, c_2)}{\partial c_2} du \quad = \quad \frac{2q_2^3}{3\mu_2 c_2^3}, \tag{3.74}$$

where $\frac{1}{\mu_j}$ with $j \in \{1, 2\}$ are average packet lengths corresponding the two users as explained just after the equation in Equation (2.4). Then, we can formulate the corresponding minimal dual-cost non-zero-sum game problem as

$$\min_{q \in R_+^2} C_j(q, c) \quad \text{subject to} \quad \frac{q_1}{\mu_1} + \frac{q_2}{\mu_2} \geq w \tag{3.75}$$

for a fixed constant $w > 0$, a fixed $c \in \mathcal{R}$, and all $j \in \{0, 1, 2\}$ with $C_0(q, c) = C_1(q, c) + C_2(q, c)$. Note that, both $C_j(q_j, c_j)$ for $j \in \{1, 2\}$ are strictly increasing in terms of q_j. Thus, a Pareto minimal dual-cost Nash equilibrium point must be located on the line where the equality of the constraint inequality in Equation (3.75) holds (i.e. $q_1/\mu_1 + q_2/\mu_2 = w$). Then, we have that

$$q_j = \mu_j \left(w - \frac{q_{2-j+1}}{\mu_{2-j+1}} \right) \quad \text{with} \quad j \in \{1, 2\}. \tag{3.76}$$

Therefore, it follows from Equation (3.76) that

$$f(q_1) \equiv \sum_{j=1}^{2} C_j(q_j, c_j) = \frac{q_1^2}{2\mu_1 c_1} + \frac{2\mu_2^2}{3c_2^3} \left(w - \frac{q_1}{\mu_1} \right)^3. \tag{3.77}$$

Furthermore, we can get the minimal value of the function $f(q_1)$ at a point such that $\frac{\partial f(q_1)}{\partial q_1} = 0$, i.e. the unique Pareto minimal Nash equilibrium point $q^*(w) = (q_1^*, q_2^*)(w)$ to the problem in Equation (3.75) can be explicitly solved as

$$\begin{cases} q_1^*(w) & = \frac{1}{2}\left(\frac{2w}{\mu_1} + \frac{c_2^3}{2c_1\mu_2^2}\right)\mu_1^2 - \sqrt{\frac{1}{4}\left(\frac{2w}{\mu_1} + \frac{c_2^3}{2c_1\mu_2^2}\right)^2 \mu_1^4 - \mu_1^2 w^2}, \\ q_2^*(w) & = \mu_2\left(w - \frac{q_1^*}{\mu_1}\right), \end{cases} \tag{3.78}$$

which is the intersection point of the red and blue curves in the right graph of Figure 13.

In other words, we have

$$C_0(q^*, c) \leq C_0(q, c), \tag{3.79}$$

$$C_1(q^*, c) \leq C_1(q_{-1}^*, c) \quad \text{with} \quad q_{-1}^* = (q_1, q_2^*), \tag{3.80}$$

$$C_2(q^*, c) \leq C_2(q_{-2}^*, c) \quad \text{with} \quad q_{-2}^* = (q_1^*, q_2). \tag{3.81}$$

From the inequalities in Equations (3.79)–(3.81), one can see that, if a dual-cost game player j with $j \in \{1, 2\}$ unilaterally changes his bid's policy q_j, his cost cannot be

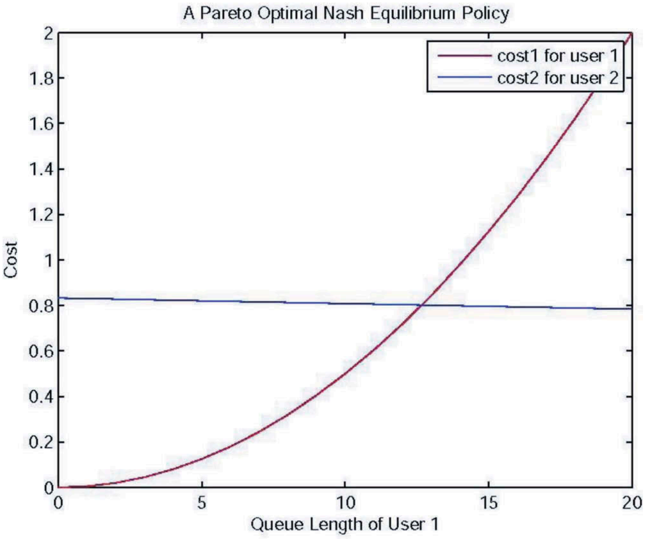

Figure 13. A Pareto optimal Nash equilibrium policy.

reduced. Furthermore, these properties can help us to prove the effectiveness of the designed policy by the Nash-equilibrium point to the game problem in Equation (3.14) through the corresponding one to the dual-cost game problem in Equation (3.75). More precisely, in the following Model Justification Part II (Subsection 3.2.3), we have the claim that the physical workload $W(t)$ in Equation (2.8) for this single pool case can indeed be modelled by an RDRS in Definition 3.1 when the nominal load rate (certain average rate of the random rate Λ in Equation (2.7)) closes to the Nash-equilibrium point c^* (the service capacity limit). Furthermore, as stated in Claim 3.1 and justified in Theorem 3.1, $W(t)$ is asymptotically minimal at any time $t \in [0, \infty)$ almost surely along any sample path under some supporting probability space. In the meanwhile, the queue length process $Q(t)$ in Equation (2.6) is also fairly minimized in the sense that it is the asymptotically minimal dual-cost Nash equilibrium point process given by $Q(t) = q^*(W(t))$, where $q^*(w)$ is given in Equation (3.78).

Note that, for this example, it follows from Theorem 3.1 that the coefficients of the one-dimensional RDRS under our game-based scheduling policy for the physical workload process $\hat{W}$ can be denoted by

$$\hat{b} = \frac{\theta_1}{\mu_1} + \frac{\theta_2}{\mu_2}, \ \hat{\sigma}^E = \hat{\sigma}^S = \left(\frac{1}{\mu_1}, \frac{1}{\mu_2}\right), \ \hat{R} = 1. \tag{3.82}$$

Furthermore, by extending the discussion in page 15 of the work by Harrison [38] and considering each $w, t \in [0, \infty)$, we can derive the distribution for $\hat{W}(t)$ as follows:

$$F(t, w) = P\{\hat{W}(t) \le w\} = \Phi\left\{\frac{w - \hat{b}t}{\hat{\sigma}t^{1/2}}\right\} - \exp\left(\frac{2\hat{b}w}{\hat{\sigma}^2}\right)\Phi\left\{\frac{-w - \hat{b}t}{\hat{\sigma}t^{1/2}}\right\}, \tag{3.83}$$

where Φ is the standard normal distribution, and

$$\hat{\sigma} = \sqrt{\left(\sum_{j=1}^{2} \hat{\sigma}_j^E \sqrt{\Gamma_{jj}^E}\right)^2 + \left(\sum_{j=1}^{2} \hat{\sigma}_j^S \sqrt{\Gamma_{jj}^S}\right)^2}. \tag{3.84}$$

Then, by the one-dimensional RDRS corresponding to Equation (3.82) and the Pareto minimal Nash equilibrium policy q^* in Equation (3.78), we can derive the iterative formula (called Simulation Algorithm 3.1 in Subsection 3.2.3) for our workload-based simulation. Note that, all the processes related to the workload processes will be covered with a ‘hat’. Similarly, in this example, the coefficients of the corresponding two-dimensional RDRS under a static proportional allocation strategy designed in the following Model Justification Part II (Subsection 4.2) for the limit queue length process $\hat{Q}^{II}$ can be denoted by

$$b = (\theta_1, \theta_2)', \ \sigma^S = \sigma^E = I_{2\times 2}, \ R = \begin{pmatrix} 1 & R_{12} \\ R_{21} & 1 \end{pmatrix}, \tag{3.85}$$

$$R_{jk} = \frac{\rho_j - c_j^{\{k\}}}{\rho_k} \quad \text{for} \ \ k \neq j \ \ \text{and} \ \ k, j \in \{1, 2\}. \tag{3.86}$$

Then, by the two-dimensional RDRS corresponding to Equations (3.85)–(3.86) and an alternative scheduling policy designed in Subsection 3.2.4, we can derive the iterative formula (called Simulation Algorithm 3.2 in Subsection 3.2.4) for our queue-length-based simulation.

Now, based on the derived formulas, simulation algorithms, and given parameters shown in Figures 8–10, we have the corresponding simulation results for our case study as summarized in the graphs of the figures. In this example, the number N of simulation iterative times is 6000. The simulation time interval is $[0, T]$ with $T = 800$ for all the graphs in Figures 8–10 except the first one in the right column of Figure 8, where $T = 40$. Both the intervals $[0, 800]$ and $[0, 40]$ are divided into $n = 5000$ subintervals. Furthermore, all the 'simulated means' used in the graphs except the first one in the right column of Figure 8 are in the average sense, e.g. the simulated mean workload is given by

$$E\left[\hat{W}(t_i)\right] = \frac{1}{N}\sum_{j=1}^{N}\hat{W}(\omega_j, t_i), \tag{3.87}$$

where ω_j denotes the used jth sample paths of $\hat{W}(\cdot)$. Nevertheless, the simulated mean workload in the first graph of the right column of Figure 8 is corresponding to the distribution in Equation (3.83) for some positive number $\bar{N}$, i.e.

$$E\left[\hat{W}(t_i)\right] = \frac{1}{\bar{N}}\sum_{j=1}^{\bar{N}}\hat{W}(\omega_j, t_i)dF(t_i, \hat{W}(\omega_j, t_i)). \tag{3.88}$$

Note that, in Figure 8, we give the performance evaluation and comparisons with respect to the workload process $\hat{W}(\cdot)$. The red curve displayed in the first graph of the left column is the simulated mean workload function with respect to time point t_i. The result presented in the second graph of this column shows the difference between the simulated mean workload and the corresponding one obtained through the Nash equilibrium policy in Equation (3.78), i.e.

$$E[\hat{W}(t_i)] - E[\hat{W}^*(t_i)]. \tag{3.89}$$

From the graph, we can see that these two workload processes are quite consistent. The blue curve displayed in the first graph of the right column is the comparison result between the simulated mean workload over the sampling interval $[0, 40]$ in its state space by the formula in Equation (3.88) and the corresponding exact one obtained through the distribution given in Equations (3.83)–(3.84). The purpose of this comparison is to illustrate the effectiveness of our generated random numbers from the designed iterative procedures for RDRSs and RBMs. From the graph, we can see that the simulated workload is quite consistent with its exact distribution given in Equations (3.83)–(3.84) even in the simulation 'warming-up' non-stationary period. The green curve displayed in the second graph of the right column is the difference between the simulated mean workload and the corresponding one obtained through the linear-utility-scheduling policy, which illustrates that our game-based scheduling policy outperforms this linear one.

Furthermore, in Figure 9, we deliver the performance evaluation and comparisons in terms of the dual-cost processes obtained through the functions given in Equations (3.73)–(3.74). The red, blue, and green curves displayed in the first and second graphs of the left column and in the first graph of the right column are the corresponding mean total, user 1, and user 2 dual-cost functions. The associated curves displayed in the second graph of the right column are the comparison results between the dual-costs obtained through our game-based scheduling policy and those corresponding to the alternative one designed in Model Justification Part II (Subsection 4.2). These comparison results indicate that our game-based scheduling policy outperforms the one designed in Model Justification Part II (Subsection 4.2) for the given target utility functions.

Finally, in Figure 10, we present the performance of the mean total queue length and those for two different users under our game-based scheduling policy and their comparisons with the one designed in Model Justification Part II (Subsection 4.2).

4. RDRS modelling justification

In this section, we provide the mathematical proof concerning the RDRS modelling justification presented in Theorem 3.1. To begin with, we first impose some required conditions and assumptions.

4.1. The required conditions

The utility functions can be either simply taken as the well-known proportionally fair and minimal potential delay allocations as used in Equation (3.13) for Example 3.4 or generally taken such that the existence of a Pareto maximal-utility Nash equilibrium policy to the game problem in Equation (3.32) is guaranteed. More precisely, we can suppose that $U_{vj}(q_j, c_{vj})$ for each $j \in \mathcal{J}(v)$ and $v \in \mathcal{V}(j)$ is defined on R^J_+. It is second-order differentiable and satisfies

$$U_{vj}(0, c_{vj}) = 0, \tag{4.1}$$

$$U_{vj}(q_j, c_{vj}) = \Phi_{vj}(q_j)\Psi_v(c_{vj}) \ \text{ is strictly increasing and concave in } \ c_{vj} \ \text{ for } \ q_j > 0, \tag{4.2}$$

$$\Psi_v(v_j c_{vj}) = \Psi_v(v_j)\Psi_v(c_{vj}) \ \ or \ \ \Psi_v(v_j c_{vj}) = \Psi_v(v_j) + \Psi_v(c_{vj}) \ \text{ for constant } \ v_j \geq 0, \tag{4.3}$$

$$\frac{\partial U_{vj}(q_j, c_{vj})}{\partial c_{vj}} \ \text{ is strictly increasing in } \ q_j \geq 0, \tag{4.4}$$

$$\frac{\partial U_{vj}(0, c_{vj})}{\partial c_{vj}} = 0 \ \text{ and } \ \lim_{q_j \to \infty} \frac{\partial U_{vj}(q_j, c_{vj})}{\partial c_{vj}} = +\infty \ \text{ for each } \ c_{vj} > 0. \tag{4.5}$$

Furthermore, we assume that $\{U_{vj}(q_j, c_{vj}), j \in \mathcal{J}(v), v \in \mathcal{V}(j)\}$ satisfies the so-called radial homogeneity condition, i.e. for any scalar $a > 0$, each $q > 0$, $i \in \mathcal{K}$, and $v \in \mathcal{V}$, its Pareto maximal utility Nash equilibrium point for the game has the radial homogeneity,

$$c_{vj}(aq, i) = c_{vj}(q, i). \tag{4.6}$$

In addition, we introduce a sequence of independent Markov processes indexed by $r \in \mathcal{R}$, i.e. $\{\alpha^r(\cdot), r \in \mathcal{R}\}$. These systems all have the same basic structure as described in the last section except the arrival rates $\lambda_j^r(i)$ and the holding time rates $\gamma^r(i)$ for all $i \in \mathcal{K}$, which may vary with $r \in \mathcal{R}$. Here, we assume that they satisfy the heavy traffic condition

$$r\Big(\lambda_j^r(i) - \lambda_j(i)\Big)m_j(i) \to \theta_j(i) \ \text{ as } \ r \to \infty, \ \ \gamma^r(i) = \frac{\gamma(i)}{r^2} \tag{4.7}$$

for each $j \in \mathcal{J}$. Note that, $\theta_j(i) \in R$ is some constant for each $i \in \mathcal{K}$ and $j \in \mathcal{J}$, which can be chosen optimally in certain environment (see, e.g. [12]). In addition, we suppose that the nominal arrival rate $\lambda_j(i)$ is given by

$$\lambda_j(i)m_j(i) \equiv \mu_j\rho_j(i). \tag{4.8}$$

In practice, this nominal arrival rate is corresponding to the number of links allowed in a data service system, which can be realized by the technique of admission control (see, e.g. [39] and therein). Furthermore, $\rho_j(i)$ for each $j \in \mathcal{J}$ in Equation (4.8) is the nominal throughput determined by

$$\rho_j(i) = \sum_{v \in \mathcal{V}(j)} \rho_{vj}(i) \ \text{ and } \ \rho_{vj}(i) = v_{vj}\bar{\rho}_{vj}(i) \tag{4.9}$$

with $\rho_{v\cdot}(i) \in \mathcal{O}_v(i)$. In addition, $v_{v\cdot}$ and $\bar{\rho}_{v\cdot}(i)$ are an J_v-dimensional constant vector and a reference service rate vector, respectively, at service pool v, satisfying

$$\sum_{j \in \mathcal{J}(v)} v_j = J_v, \ v_j \geq 0 \ \text{ are constants for all } \ j \in \mathcal{J}(v), \tag{4.10}$$

$$\sum_{j \in \mathcal{J}(v)} \bar{\rho}_{vj}(i) = C_{U_v}(i) \ \text{ and } \ \bar{\rho}_{v1}(i) = \bar{\rho}_{vj}(i) \ \text{ for all } \ j \in \mathcal{J}(v). \tag{4.11}$$

Remark 4.1 *By Equation (3.22), $\bar{\rho}_{v\cdot}(i)$ for each $i \in \mathcal{K}$ and $v \in \mathcal{V}(j)$ can indeed be selected, which satisfy the second condition in Equation (4.11). Thus, the nominal throughput $\rho(i)$ in Equation (4.8) can be determined. One simple example that satisfies these conditions is to take $v_{vj} = 1$ for all $j \in \mathcal{J}(v)$ and $v \in \mathcal{V}(j)$. Then, the conditions in Equations (4.8)–(4.11) mean that the system manager hopes to maximally and fairly allocate capacity to all users. Furthermore, the system design parameters $\lambda_j(i)$ for all $j \in \mathcal{J}$ and each $i \in \mathcal{K}$ can be determined by Equation (4.8).*

Now, we suppose that the inter-arrival time corresponding to the kth arriving job batch to the system indexed by $r \in \mathcal{R}$ is given by

$$u_j^r(k,i)=\frac{\hat{u}_j(k)}{\lambda_j^r(i)} \quad \text{for each } \ j\in\mathcal{J},\ k\in\{1,2,\ldots\},\ i\in\mathcal{K}, \tag{4.12}$$

where the $\hat{u}_j(k)$ does not depend on r and i. Furthermore, it has mean one and finite squared coefficient of variation α_j^2. In addition, the number of packets, $w_j(k)$, and the packet length $v_j(k)$ are assumed not to change with r.

From the heavy traffic condition in Equation (4.7) for the rth environmental state process $\alpha^r(\cdot)$ with $r\in\mathcal{R}$, we know that $\alpha^r(r^2\cdot)$ and $\alpha(\cdot)$ are equal each other in distribution since they own the same generator matrix (see, e.g. the definition in pages 384–388 of Resnick [31]). Therefore, in the sense of distribution, all of the systems indexed by $r\in\mathcal{R}$ in Equation (3.1) share the same random environment over any time interval $[0,t]$.

4.2. Proof of theorem 3.1

First, from the second condition in Equation (4.7), we know that the processes $\alpha^r(r^2\cdot)$ for each $r\in\mathcal{R}$ and $\alpha(\cdot)$ are equal in distribution. Thus, without loss of generality, we can suppose that

$$\alpha^r(r^2t)=\alpha(t) \quad \text{for each } \ r\in\mathcal{R} \quad \text{and} \quad t\in[0,\infty). \tag{4.13}$$

Then, for each $j\in\mathcal{J}$, $r\in\mathcal{R}$ and by the radial homogeneity of $\Lambda(q,i)$ of the policy in Equation (4.6), we can define the fluid and diffusion scaled processes as follows:

$$E_j^r(\cdot)\equiv A_j^r(r^2\cdot), \tag{4.14}$$

$$\bar{T}_j^r(\cdot)\equiv\int_0^{\cdot}\Lambda_j\big(\bar{\mathcal{Q}}^r(s),\alpha(s)\big)ds=\frac{1}{r^2}T_j^r(r^2\cdot), \tag{4.15}$$

$$\bar{\mathcal{Q}}_j^r(t)\equiv\frac{1}{r^2}\mathcal{Q}_j^r(r^2t), \tag{4.16}$$

$$\bar{E}_j^r(t)\equiv\frac{1}{r^2}E_j^r(t), \tag{4.17}$$

$$\bar{S}_j^r(t)\equiv\frac{1}{r^2}S_j^r(r^2t). \tag{4.18}$$

Thus, by Equation (2.6), Equation (4.13), and the assumptions among arrival and service processes, we know that

$$\hat{\mathcal{Q}}_j^r(\cdot)=\frac{1}{r}E_j^r(\cdot)-\frac{1}{r}S_j^r(\bar{T}_j^r(\cdot)). \tag{4.19}$$

Furthermore, let

$$\hat{E}^r(\cdot)=(\hat{E}_1^r(\cdot),\ldots,\hat{E}_J^r(\cdot))' \ \text{ with } \ \hat{E}_j^r(\cdot)=\frac{1}{r}\Big(A_j^r(r^2\cdot)-r^2\bar{\lambda}_j^r(\cdot)\Big), \tag{4.20}$$

$$\hat{S}^r(\cdot) = (\hat{S}_1^r(\cdot), ..., \hat{S}_J^r(\cdot))' \quad \text{with} \quad \hat{S}_j^r(\cdot) = \frac{1}{r}\left(S_j(r^2\cdot) - \mu_j r^2\cdot\right) \tag{4.21}$$

for each $j \in \mathcal{J}$ with

$$\begin{aligned} \bar{\lambda}_j^r(\cdot) &\equiv \int_0^{\cdot} m_j(\alpha(s))\lambda_j^r(\alpha(s))ds \\ &= \int_0^{\cdot} m_j(\alpha^r(r^2 s))\lambda_j^r(\alpha^r(r^2 s))ds \\ &= \tfrac{1}{r^2}\int_0^{r^2\cdot} m_j(\alpha^r(s))\lambda_j^r(\alpha^r(s))ds, \end{aligned} \tag{4.22}$$

and define

$$\bar{\lambda}^r(\cdot) = \left(\bar{\lambda}_1^r(\cdot), ..., \bar{\lambda}_J^r(\cdot)\right). \tag{4.23}$$

In addition, we use $\bar{Q}^r(\cdot)$, $\bar{E}^r(\cdot)$, $\bar{S}^r(\cdot)$, and $\bar{T}^r(\cdot)$ to denote the corresponding vector processes. Finally, as a fundamental tool in our subsequent justification, we prove the following generalized functional central limit theorem for TSRRPs, which also has the potential to be applied in more fields.

Lemma 4.1. *For the diffusion-scaled process in Equation (4.20), the following convergence in distribution holds, i.e. as $r \to \infty$,*

$$\hat{E}^r(\cdot) \Rightarrow H^E(\cdot). \tag{4.24}$$

PROOF. If the jump size related to each reward is bounded, the proof of the current lemma is a direct generalization of the one for Lemma 6 by Dai [5]. However, if the jump size is a unbounded random variable, the generalization needs some additional work. Nevertheless, due to the length of this proof, it will be provided in the Appendix of this paper. □

Next, corresponding to the processes in Equations (4.14)–(4.19), we define the following fluid limit related processes,

$$\bar{Q}_j(t) = \bar{Q}_j(0) + \bar{\lambda}_j(t, \zeta_t(\cdot)) - \mu_j \bar{T}_j(t) \quad \text{for each} \quad j \in \mathcal{J} \tag{4.25}$$

$$\bar{\lambda}(t) = \left(\bar{\lambda}_1(t), ..., \bar{\lambda}_J(t)\right), \ \bar{\lambda}_j(t) \equiv \int_0^t \lambda_j(\alpha(s))ds, \tag{4.26}$$

$$\bar{T}_j(t) = \int_0^t \bar{\Lambda}_j(\bar{\mathcal{Q}}(s), \alpha(s))ds, \tag{4.27}$$

$$\bar{\Lambda}_j(q, i) = \begin{cases} \Lambda_j(q, i) & \text{if} \quad q_j > 0, \\ \rho_j(i) & \text{if} \quad q_j = 0, \end{cases} \quad \text{for each} \quad i \in \mathcal{K}. \tag{4.28}$$

Then, we have the associated lemma concerning the weak convergence to a random fluid limit process under our scheduling game policy.

Lemma 4.2. *If $\bar{Q}^r(0) \Rightarrow \bar{Q}(0)$ along $r \in \mathcal{R}$, the joint convergence in distribution along a subsequence of $\mathcal{R}$ holds under the conditions required by Theorem 3.1,*

$$\left(\bar{E}^r(\cdot), \bar{S}^r(\cdot), \bar{T}^r(\cdot), \bar{\mathcal{Q}}^r(\cdot)\right) \Rightarrow \left(\bar{E}(\cdot), \bar{S}(\cdot), \bar{T}(\cdot), \bar{\mathcal{Q}}(\cdot)\right). \tag{4.29}$$

Furthermore, if $\bar{Q}(0) = (0)$, the convergence is true along the whole $\mathcal{R}$ and the limit satisfies

$$\bar{E}(\cdot) = \bar{\lambda}(\cdot), \;\; \bar{S}(\cdot) = \mu(\cdot), \;\; \bar{T}(\cdot) = \bar{c}(\cdot), \;\; \bar{\mathcal{Q}}(\cdot) = 0, \tag{4.30}$$

where $\bar{\lambda}(\cdot)$ is defined in Equation (4.26), $\mu(\cdot) \equiv (\mu_1, ..., \mu_J)' \cdot$, and $\bar{c}(\cdot)$ is defined by

$$\bar{c}(t) = (\bar{c}_1(t), ..., \bar{c}_J(t)) \;\; \textit{and} \;\; \bar{c}_j(t) \equiv \int_0^t \rho_j(\alpha(s))ds \;\; \textit{for each} \;\; j \in \mathcal{J}. \tag{4.31}$$

Proof. It follows from the proof of Lemma 7 by Dai [5] that we only need to prove the claim $\bar{Q}(\cdot) = 0$ in Equation (4.30) to be true for our current purpose. In fact, for each $i \in \mathcal{K}$, we define

$$\psi(q, i) \equiv \sum_{v \in \mathcal{V}} \psi_v(q, i) = \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{J}(v)} C_{vj}(q_j, \rho_{vj}(i)). \tag{4.32}$$

By the proof by Dai [5], we know that all the limits in Equation (4.30) are absolutely continuous and differentiable at almost all $t \in (0, \infty)$; in other words, almost every $t \in (0, \infty)$ is a *regular point* of these limits. Thus, for each regular time $t \geq 0$ of $\bar{\mathcal{Q}}(t)$ over time interval (τ_{n-1}, τ_n) with a given $n \in \{1, 2, ...\}$, we have

$$\begin{aligned}
&\frac{d\psi(\bar{Q}(t), \alpha(t))}{dt} \\
&= \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{J}(v)} \left(\frac{d\bar{Q}_j(t)}{dt} \frac{\partial C_{vj}(\bar{Q}_j(t), \rho_{vj}(\alpha(t)))}{\partial \bar{Q}_j(t)} + \frac{d\rho_{vj}(\alpha(t))}{dt} \frac{\partial C_{vj}(\bar{Q}_j(t), \rho_{vj}(\alpha(t)))}{\partial \rho_{vj}(\alpha(t))} \right) \\
&= \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{J}(v)} \left(\rho_{vj}(\alpha(t)) - \Lambda_{vj}(\bar{Q}(t), \alpha(t)) \right) \frac{\partial U_{vj}(\bar{Q}_j(t), \rho_{vj}(\alpha(t)))}{\partial \rho_{vj}(\alpha(t))} I_{\{\bar{Q}_j^I(t) > 0\}} \leq 0,
\end{aligned} \tag{4.33}$$

where the second equality follows from the concavity of the utility functions and the fact that $\Lambda_{vj}(\bar{Q}(t), \alpha(t))$ is the Pareto maximal Nash equilibrium policy to the utility-maximal game problem in Equation (3.32). Therefore, for any given $n \in \{0, 1, 2, ...\}$ and each $t \in [\tau_n, \tau_{n+1})$,

$$
\begin{aligned}
0 &\le \psi(\bar{Q}(t), \alpha(t) \\
&\le \psi(\bar{Q}(\tau_n), \alpha(\tau_n)) \\
&= \sum_{v\in\mathcal{V}} \sum_{j\in\mathcal{J}(v)} \frac{1}{\mu_j} \int_0^{\bar{Q}_j(\tau_n)} \frac{\partial U_{vj}(u, \rho_{vj}(\alpha(\tau_n)))}{\partial C_{vj}} du \\
&= \sum_{v\in\mathcal{V}} \left(\frac{d\psi_v(\bar{\rho}_{v1}(\alpha(\tau_n)))}{dc_{v1}}\right) \left(\frac{d\psi_v(\bar{\rho}_{v1}(\alpha(\tau_{n-1})))}{dc_{v1}}\right)^{-1} \psi_v(\bar{Q}(\tau_n), \alpha(\tau_{n-1})) \ldots \\
&\le \sum_{v\in\mathcal{V}} \left(\frac{d\psi_v(\bar{\rho}_{v1}(\alpha(\tau_n)))}{dc_{v1}}\right) \left(\frac{d\psi_v(\bar{\rho}_{v1}(\alpha(\tau_0)))}{dc_{v1}}\right)^{-1} \psi_v(\bar{Q}(0), \alpha(0)) \\
&\le \kappa \psi(\bar{Q}(0), \alpha(0)),
\end{aligned} \tag{4.34}
$$

where κ is a positive constant given by

$$
\kappa = \max_{v\in\mathcal{V}} \max_{i,j\in\mathcal{K}} \left(\frac{d\psi_v(\bar{\rho}_{v1}(i))}{dc_{v1}}\right) \left(\frac{d\psi_v(\bar{\rho}_{v1}(j))}{dc_{v1}}\right)^{-1}.
$$

Then, it follows from the fact in Equation (4.34) that $\bar{Q}^I(t) = 0$ for all $t \ge 0$. Hence, we complete the proof of the lemma. □

Next, we have the following three lemmas concerning the relationship between the utility-maximal and its dual cost-minimal games.

Lemma 4.3. *Under the conditions in Equations (4.1)–(4.6), if $\Lambda(q, i) \in F_{\mathcal{Q}}(i)$ for each $i \in \mathcal{K}$ is a given Pareto optimal Nash equilibrium policy to the utility-maximal game problem in Equation (3.32) and $\{q^l, l \in \mathcal{R}\}$ is a sequence of queue states, which satisfies $q^l \to q \in R_+^J$ as $l \to \infty$, then, for each $j \in \mathcal{J} \backslash \mathcal{Q}(q)$ and $v \in \mathcal{V}(j)$, we have,*

$$
\Lambda_{vj}(q^l, i) \to \Lambda_{vj}(q, i) \quad as \quad l \to \infty. \tag{4.35}
$$

Proof. Note that, our Pareto maximal Nash equilibrium policy to the utility-maximal game problem in Equation (3.32) is also an optimal solution to the corresponding utility-maximization problem as by Dai [5]. Thus, our claim can be proved by applying the radial homogeneity assumption in Equation (6) and the similar way as used in proving Lemma 3 of Dai [5]. □

Lemma 4.4. *For each environmental state $i \in \mathcal{K}$, the following two claims are true.*

Claim 1. *For a given queue state $q \in R_+^J$ and a scheduling policy $\Lambda(q, i) = c^*(i) \in F_{\mathcal{Q}(q)}(i)$, if $c^*(i) = \rho(i)$ is the Pareto maximal Nash equilibrium policy to the utility-maximal non-zero-sum game problem in Equation (3.32), then $q^* = q$ must be the Pareto minimal Nash equilibrium policy to the dual cost-minimal non-zero-sum game problem in Equation (3.41) with $c(i) = c^*(i)$ as parameters of the cost functions and with $w = \sum_{j\in\mathcal{C}(c^*(i))} q_j^*/\mu_j$ in the constraints, i.e. $q^*(w, \rho(i)) = q^*$.*

Claim 2. *Suppose that q^* is the Pareto minimal Nash equilibrium policy to the dual cost-minimal non-zero-sum problem in Equation (3.41) with $w > 0$ and $\Lambda(q^*, i) = \rho(i)$ as parameters of the cost functions, i.e. $q^*(w, \rho(i)) = q^*$. Then, the claim that $q^* > 0$ is true.*

Furthermore, $\Lambda^*(q^*, i) = \Lambda(q^*, i)$ must be the Pareto maximal Nash equilibrium policy to the utility-maximal non-zero-sum game problem in Equation (3.32) with $q = q^*$ as a parameter of the utility functions.

Proof. First, we prove the result in Claim 1 to be true. Without loss of generality, we suppose that $q > 0$. Then, for each $v \in \mathcal{V}$, it follows from the KKT optimality conditions (see, e.g. [40]) that the Pareto optimal Nash equilibrium policy to the utility-maximal game problem in Equation (3.32) can be determined by the solution to the system of equations

$$\begin{cases} c_{vj}\left(\frac{\partial U_{vj}(q_j, c_{vj})}{\partial c_{vj}} + \sum_{k=1}^{B_v} \eta_{vk} \frac{\partial h_{vk}(c_{v\cdot}, i)}{\partial c_{vj}}\right) & = 0 \quad \text{for each } \ j \in \mathcal{J}(v), \\ \eta_{vk} h_{vk}(c_{v\cdot}, i) & = 0 \quad \text{for each } \ k \in \mathcal{U}_v, \end{cases} \tag{4.36}$$

where B_v and $\mathcal{U}_v$ are defined in Equation (3.21), $\eta_{vk} \geq 0$ for all $k \in \mathcal{U}_v$ are the Lagrangian multipliers, and $h_{vk}(c_{v\cdot}, i)$ for each $k \in \mathcal{U}_v$ and $i \in K$ is defined in Equation (3.21). By the same way, the Pareto optimal Nash equilibrium policy to the dual cost-minimal non-zero-sum game problem (Equation (3.41)) can be obtained by the solution to the system of equations

$$\begin{cases} q_j\left(\frac{\partial C_{vj}(q_j, c_{vj})}{\partial q_j} + \frac{\theta_v}{\mu_j}\right) & = 0 \quad \text{for each } \ j \in \mathcal{J}(v), \\ \theta_v\left(w - \sum_{j \in \mathcal{J}(v)} \frac{q_j}{\mu_j}\right) & = 0, \end{cases} \tag{4.37}$$

where $\theta_v \geq 0$ is the Lagrangian multiplier. Furthermore, it follows from the definition of the cost function in Equation (3.41) that

$$\frac{\partial C_{vj}(q_j, c_{vj})}{\partial q_j} = \frac{1}{\mu_j} \frac{\partial U_{vj}(q_j, c_{vj})}{\partial c_{vj}}. \tag{4.38}$$

In addition, by the condition in Equation (4.2), we know that $U_{vj}(q, c_v)$ for all $j \in \{0\} \cup \mathcal{J}(v)$ are strictly concave in c_v for each $q > 0$. Thus, $\{c^*_{v\cdot}(i) = \rho_{v\cdot}(i)$ for all $v \in \mathcal{V}\}$ is the unique Pareto optimal Nash equilibrium to the utility-maximal game problem in Equation (3.32) for the given $q > 0$ as a parameter of the utility functions, which satisfies Equation (4.36). Hence, if we take

$$\theta_v = -\sum_{k=1}^{B} \eta_{vk} \frac{\partial h_{vk}(\rho_{v\cdot}(i), i)}{\partial c_{vj}},$$

it follows from Equation (4.36) and Equation (4.38) that Equation (4.37) holds. From the condition in Equation (4.5), we know that $C_{vj}(q, c_{v\cdot})$ for all $j \in \{0, 1, ..., J\}$ and $v \in V(j)$ are strictly convex in q for each $c_{v\cdot} > 0$. Therefore, the dual cost-minimal game problem in Equation (3.41) has the unique Pareto optimal Nash equilibrium policy $q^* = q$ when $c_{v\cdot} = c^*_{v\cdot}(i) = \rho_{v\cdot}(i)$ for each $v \in \mathcal{V}$ is as a parameter of the cost functions and $w = \sum_{j=1}^{J} q^*_j / \mu_j$ is in the constraints.

Second, we prove the result in Claim 2 to be true. It follows from the conditions in Equations (4.4)–(4.5) and the relationship between the utility and cost functions in Equation (3.41), we know that $C_{vj}(q, \rho_{v\cdot}(i))$ for all $j \in \{0, 1, ..., J\}$ and $v \in \mathcal{V}(j)$ are strictly convex in q. Thus, q^* is the unique Pareto optimal Nash equilibrium policy to the dual cost-minimal game problem (Equation (3.41)) with $w > 0$ and

$\Lambda_{v\cdot}(q^*, i) = \rho_{v\cdot}(i)$. Then, we can prove that $q^* > 0$ by showing a contradiction. In fact, without loss of generality, we assume that there exists some index $m \in \mathcal{J}$ with $m < J_v$ such that

$$\mathcal{Q}(q^*) \cap \mathcal{J}(v) = \{k_1, \ldots, k_m\} \text{ with } k_1 \neq j_{v1} \text{ and } k_m = j_{vJ_v}. \tag{4.39}$$

Thus, we can construct a 2-dimensional line for some constant $\epsilon \geq w$,

$$P_1 : \frac{q_{j_{v1}}}{\mu_{j_{v1}}} + \frac{q_{j_{J_v}}}{\mu_{j_{J_v}}} + \sum_{j \neq j_{v1}, j_{vJ_v}, j \in \mathcal{J}(v)} \frac{q_j^*}{\mu_j} = \epsilon \geq w \tag{4.40}$$

such that it passes through the point q^*, where we have used the corresponding index method as in $\mathcal{J}(v)$ for q and μ. Now, it follows from the relationship between the utility and cost functions given in Equation (3.41) that the function $f_v(q_{j_{v1}}, \rho_{v\cdot}(i))$ $(= C_{v0}(q, \rho_{v\cdot}(i)))$ for a given $v \in \mathcal{V}$ with the constraint P_1 for all $q = (q_{v1}, q_{v2}^*, \ldots, q_{v(J_v-1)}^*, q_{vJ_v})' \in \mathcal{R}_+^{J_v}$ is of the following derivative function in $q_{v1} \in \mathcal{R}_1^+$:

$$\begin{aligned} \frac{\partial f_v(q_{v1}, \rho_{v\cdot}(i))}{\partial q_{v1}} &= \frac{1}{\mu_{v1}} \frac{\partial U_{v1}(q_{v1}, \rho_{v1}(i))}{\partial c_{v1}} \\ &\quad - \frac{1}{\mu_{v1}} \frac{\partial U_{vJ_v}((\epsilon - \frac{q_{v1}}{\mu_{v1}} - \sum_{j \neq j_{v1}, j_{vJ_v}, j \in \mathcal{J}(v)} \frac{q_{vj}^*}{\mu_{vj}})\mu_{vj_{J_v}}, \rho_{vJ_v}(i))}{\partial c_{vJ_v}}. \end{aligned} \tag{4.41}$$

Furthermore, it is strictly increasing in $q_{v1} \in \mathcal{R}_1^+$ from Equation (4.4). Thus, it follows from Equations (4.41) and (4.5) that

$$\frac{\partial f_v(0, \rho_{v\cdot}(i))}{\partial q_{v1}} = -\frac{1}{\mu_{v1}} \frac{\partial U_{vJ_v}((\epsilon - \sum_{j \neq j_{v1}, j_{j_{J_v}}, j \in \mathcal{J}(v)} \frac{q_j^*}{\mu_j})\mu_{vJ_v}, \rho_{vJ_v}(i))}{\partial c_{vJ_v}} < 0, \tag{4.42}$$

$$\frac{\partial f_v(q_{v1}^*, \rho_{v\cdot}(i))}{\partial q_{v1}} = \frac{1}{\mu_{v1}} \frac{\partial U_{v1}(q_{v1}^*, \rho_{v1}(i))}{\partial c_{v1}} > 0. \tag{4.43}$$

Then, by Equations (4.42) and (4.43), we know that there is a $\tilde{q}_{v1} \in (0, q_{v1}^*)$ such that

$$\frac{\partial f_v(\tilde{q}_{v1}, \rho_{v\cdot}(i))}{\partial q_{v1}} = 0, \tag{4.44}$$

which implies that on the curve $f(q, \rho(i))$ with $q = (q_1, q_2^*, \ldots, q_{J-1}^*, q_J)' \in R_+^J$, there exists a minimal point $\tilde{q} \in \mathcal{R}_+^J$ with $\tilde{q} = (\tilde{q}_1, q_2^*, \ldots, q_{J-1}^*, \tilde{q}_J)'$ such that

$$C_{v0}(\tilde{q}, \rho_{v\cdot}(i)) < C_{v0}(q^*, \rho_{v\cdot}(i))$$

$$\tilde{q}_J = \left(\epsilon - \frac{\tilde{q}_{v1}}{\mu_{v1}} - \sum_{j \neq j_{v1}, j_{vJ_v}, j \in \mathcal{J}(v)} \frac{q_j^*}{\mu_j} \right) \mu_{vJ_v}.$$

This contradicts the assumption that q^* is the Pareto optimal Nash equilibrium policy to the dual cost-minimal game problem in Equation (3.41). Thus, we can conclude that $q^* > 0$.

Hence, if q^* is the Pareto optimal Nash equilibrium policy to the dual cost-minimal game problem in Equation (3.41) with $c = \Lambda(q^*, i) = \rho(i)$ for each $v \in \mathcal{V}$ as a parameter of the cost functions, we see that the equalities in Equation (4.37) hold with $q = q^*$ and $c_{v\cdot} = \Lambda_{v\cdot}(q^*, i) = \rho_{v\cdot}(i)$. Thus, we can take $\eta_{vk_{U_v}} = \theta_v$ and $\eta_{vk} = 0$ when $k \neq k_{U_v}$ in Equation (4.36) since $q^* > 0$ and $\rho_{v\cdot}(i)$ is on the curve $h_{vk_{U_v}}(c_{v\cdot}, i) = 0$. Therefore, $\Lambda^*_{v\cdot}(q^*, i) = \Lambda_v(q^*, i) = \rho_{v\cdot}(i)$ for each $i \in \mathcal{K}$ and $v \in \mathcal{V}$ is the Pareto optimal Nash equilibrium policy to the utility-maximal game problem in Equation (3.26) with $q = q^*$ as a parameter of the utility functions. □

Now, let $\parallel \cdot \parallel$ denote the norm of a vector $q \in R^J_+$ in the sense that $\parallel q \parallel = \sum_{j=1}^J |q_j|$. Then, we have the following lemma.

Lemma 4.5. *For the dual cost-minimal non-zero-sum game problem in Equation (3.41) associated with each state $i \in \mathcal{K}$, the following two claims are true.*

Claim 1. *It has a unique Pareto optimal Nash equilibrium policy $q^*(w, \rho(i))$ when $c_v = \rho_v(i)$ for each $i \in \mathcal{K}$ and $v \in \mathcal{V}$ is as a parameter of the cost functions. Furthermore, $q^*(w, \rho(i))$ is continuous in terms of w.*

Claim 2. *If, for any given constant $\epsilon > 0$, there exists another constant $\sigma > 0$ that depends only on ϵ, such that for any $q \in \mathcal{V}(\epsilon, \sigma, i)$ with*

$$\mathcal{V}(\epsilon, \sigma, i) \equiv \left\{ q \in R^J_+ : \|q - q^*(w, \rho(i))\| \leq \sigma \;\; \textit{and} \;\; w = \sum_{j=1}^J \frac{1}{\mu_j} q_j \;\; \geq \epsilon \right\}, \tag{4.45}$$

then we have

$$\sum_{j=1}^J \Lambda_j(q, i) = \sum_{j=1}^J \rho_j(i). \tag{4.46}$$

Proof. Proof of Lemma 4.5. Note that it follows from the condition in Equation (4.5) that $C_{vj}(q, c)$ for $j \in \{0, 1, ..., J\}$ and $v \in \mathcal{V}(j)$ is strictly convex in q for each $c_v > 0$. Thus, the dual cost-minimal game problem in Equation (3.41) has a unique Pareto optimal Nash equilibrium policy $q^* = q$ when $c_v = \rho_v(i)$ for each $v \in \mathcal{V}$ is as a parameter of the cost functions. Furthermore, the rest of the proof for this lemma follows from the one for Lemma 5 by Dai [5]. □

Finally, by applying Lemmas 4.1–4.5 in the current paper to the proof for Theorem 1 in [5], we can complete the proof for Theorem 3.1 in the current paper. □

5. Conclusion

In this paper, we have developed a generic game platform with multiple intelligent cloud-computing pools and parallel-queues for resources-competing users. It can be used to model various real-world systems including cloud-computing with multi-supercomputer centres, MIMO wireless channels, and Internet of Energy. Inside the platform, the software structure is modelled as Blockchain. All the users are associated with

Big Data arrival streams whose random dynamics is modelled by TSRRPs. Each user may be served simultaneously by multiple pools while each pool with parallel-servers may also serve multi-users at the same time via running smart policies in the Blockchain, e.g. solving a Nash equilibrium point myopically at each fixed time to a game-theoretic scheduling problem. To illustrate the effectiveness of our game platform, we model the performance measures of its internal data flow dynamics (i.e. queue length and workload processes) as RDRSs under our designed scheduling policies. By these RDRS models, we can prove our myopic game-theoretic policy to be an asymptotic Pareto minimal-dual-cost Nash equilibrium one globally over the whole time horizon to a randomly evolving dynamic game problem. In the meanwhile, we also develop iterative schemes for simulating our multi-dimensional RDRS models with the support of numerical comparisons to illustrate the effectiveness of the RDRS models and our game-theoretic policy. More applications and smart engines will be introduced and invented into our game platform.

Appendix: Proof of Lemma 4.1

First of all, we remark that the proof for the current lemma is a generalization from the one for Lemma 6 by Dai [5]. The major difference between these two proofs is that the jump size at each time τ_n with $n \in \{1, 2, \ldots\}$ for the renewal reward process in the current proof may be a unbounded random variable while the corresponding jump size at time τ_n for the discussion based on a renewal process is a bounded one by Dai [5].

More precisely, it follows from the heavy traffic condition (Equation (4.7)), the functional central limit theorem for renewal reward process (see, e.g. Theorem 7.4.1 in page 239 of Whitt [19]), the random change of time lemma (see, e.g. page 151 of Billingsley [41]), the Lemma 8.4 in Dai and Dai [42] that

$$\begin{aligned}
&\Big(\hat{E}^r(\tau_n + t) - \hat{E}^r(\tau_n)\Big) I_{\{0 \le t < \sigma_n\}} \\
&\quad = \frac{1}{r}\big(A^r(r^2(\tau_n + t)) - A^r(r^2\tau_n)\big) I_{\{0 \le t < \sigma_n\}} - r\big(\bar{\lambda}^r(\tau_n + t) - \bar{\lambda}^r(\tau_n)\big) I_{\{0 \le t < \sigma_n\}} \\
&\quad = \frac{1}{r}\tilde{A}^r(r^2(t - \phi_n/r^2) I_{\{0 \le t < \sigma_n\}}) + \frac{1}{r} w(\tau_n, \alpha(\tau_n)) - r\big(\bar{\lambda}^r(\tau_n + t) - \bar{\lambda}^r(\tau_n)\big) I_{\{0 \le t < \sigma_n\}} \\
&\quad \Rightarrow \big(\Gamma^E(\alpha(\tau_n))\big)^{\frac{1}{2}} I_{\{0 \le t < \sigma_n\}} B^E(t) \\
&\quad =^d \big(H^E(\tau_n + t) - H^E(\tau_n)\big) I_{\{0 \le t < \sigma_n\}}
\end{aligned} \tag{5.1}$$

for each $n \in \{0, 1, \ldots\}$ as $r \to \infty$, where $\sigma_n = \tau_{n+1} - \tau_n$ is an exponentially distributed random variable independent of all other random events concerned since $\alpha(\cdot)$ is an FS-CTMC, and

$$w(\tau_n, \alpha(\tau_n)) = (w_1(\tau_n, \alpha(\tau_n)), \ldots, w_J(\tau_n, \alpha(\tau_n)))'$$

is the random reward vector at time τ_n. Furthermore, $\tilde{A}^r(\cdot)$ is a renewal reward process with

$$\tilde{A}^r(r^2(\cdot - \phi_n/r^2)) = (\tilde{A}^r_1(r^2(\cdot - \phi^1_n/r^2)), \ldots, \tilde{A}^r_J(r^2(\cdot - \phi^J_n/r^2)))',$$

where $\phi_n = (\phi^1_n, \ldots, \phi^J_n)'$ is an J-dimensional random vector whose jth component ϕ^j_n for each $j \in \{1, \ldots, J\}$ denotes the remaining arrival time beginning at τ_n for a packet batch to the jth queue with rate $\lambda_j^r(\alpha(\tau_n))$ switched from $\lambda_j^r(\alpha(\tau_{n-1}))$ at time τ_n. For convenience, we now rewrite Equation (1) in the following way over each $[\tau_n, \tau_{n+1})$ as $r \to \infty$:

$$\begin{aligned}\tilde{E}^{r,n}(\cdot) &\equiv \hat{E}^r(\tau_n + \cdot) - \hat{E}^r(\tau_n)\\ &\Rightarrow H^E(\tau_n + \cdot) - H^E(\tau_n)\\ &\equiv \tilde{H}^{E,n}(\cdot).\end{aligned} \tag{5.2}$$

Next, to prove the claim in the lemma, we establish the relative compactness for $\hat{E}^r(\cdot)$ with $r \in \mathcal{R}$. In doing so, we define the modulus of continuity in terms of a function $x(\cdot)$: $[0, \infty) \to R^d$ with some integer $d > 0$ for each given $T > 0$ and $\delta > 0$ as follows:

$$\varpi(x, \delta, T) \equiv \inf_{t_l} \max_{l} \text{ Osc } (x, [t_{l-1}, t_l)), \tag{5.3}$$

where the infimum takes over the finite sets $\{t_l\}$ of points satisfying $0 = t_0 < t_1 < \ldots < t_m = T$ and $t_l - t_{l-1} > \delta$ for $l = 1, \ldots, m$, and

$$\text{Osc } (x, [t_{l-1}, t_l]) = \sup_{t_1 \leq s \leq t \leq t_2} \| x(t) - x(s) \|_2 \tag{5.4}$$

with $\| \cdot \|_2$ denoting the Euclidean norm in R^d. Thus, it follows from Corollary 7.4 in page 129 of Ethier and Kurtz [43] that the justification of the relative compactness is equivalent to proving the following two conditions: First, for each $\eta > 0$ and rational $t \geq 0$, there exists a constant $c(\eta, t)$ such that

$$\lim_{r \to \infty} inf\, P\left\{ \left\| \hat{E}^r(t) \right\|_2 \leq c(\eta, t) \right\} \geq 1 - \eta; \tag{5.5}$$

Second, for each $\eta > 0$ and $T > 0$, there exists a $\delta > 0$ such that

$$\lim_{r \to \infty} \sup P\left\{ \varpi(\hat{E}^r, \delta, T) \geq \eta \right\} \leq \eta. \tag{5.6}$$

To show the condition in Equation (5.5) to be true, we define $N(t) \equiv \max\{n, \tau_n \leq t\}$ for each $t \in (0, \infty)$. Thus, for each rational $t > 0$, take a $T > 0$ such that $t \in (0, T]$ and introduce a sequence of events:

$$\mathcal{S}_l \equiv \{\omega: N(T, \omega) \leq l\} \text{ for each } l \in \{1, 2, \ldots, \}. \tag{5.7}$$

Since $\alpha(\cdot)$ has at most finitely many jumps a.s. over $[0, T]$, we know that the sequence of probabilities $P\{S_l\}$ increases monotonously to the unity as $l \to \infty$. Therefore, for the given $\eta > 0$, there exists some large enough $L > 0$ such that

$$P\{\mathcal{S}_L\} \geq 1 - \frac{\eta}{2}. \tag{5.8}$$

Furthermore, by Equation (5.2) and Remark 7.3 in page 129 of Ethier and Kurtz [43], we know that $\tilde{E}^r(\cdot)$ satisfies the following compact containment condition, i.e. for each $\eta > 0$ and $T > 0$, there is a constant $K_n > 0$ for each $n \in \{0, 1, \ldots,\}$ such that

$$\inf_r P\{\mathcal{T}^{r,n}\} \geq 1 - \frac{\eta}{2(L+2)}, \tag{5.9}$$

where

$$\mathcal{T}^{r,n} \equiv \{\omega: \| \tilde{E}^r(t)\|_2 \leq K_n, t \in [0,T] \cap [0,\sigma_n)\}.$$

Now, for each $n \in \{1,2,...,\}$ and $t \in [\tau_{N(t)}, \tau_{N(t)+1})$, we have the following observations

$$\hat{E}^r(t) = \hat{E}^r(\tau_{N(t)}) + \tilde{E}^{r,N(t)}(t-\tau_{N(t)}), \tag{5.10}$$

$$\hat{E}^r(\tau_n) - \hat{E}^r(\tau_n^-) = \frac{w(\tau_n, \alpha(\tau_n))}{r}. \tag{5.11}$$

Then, for any $t_1, t_2 \in [0,T]$ and along each sample path, it follows from Equations (5.10)–(5.11) that

$$\begin{aligned}\text{Osc}\left(\hat{E}^r, [t_1,t_2]\right) \leq \sum_{n=0}^{N(t_2)} \text{Osc}\left(\tilde{E}^{r,n}, [t_1-\tau_n, t_2-\tau_n] \cap [0,\sigma_n)\right) \\ + \Xi^r(T)(N(t_2)-N(t_1)),\end{aligned} \tag{5.12}$$

where Ξ^r is defined by

$$\Xi^r(T) \equiv \frac{1}{r}\left(\max_{1\leq j\leq N_{ji}(r^2T),\, i\in\mathcal{K}, j\in\mathcal{J}} w(k,j,i)\right), \tag{5.13}$$

and $w(k,j,i)$ in Equation (5.13) is the kth batch size corresponding to the jth user when the environment mode is in state $i \in \mathcal{K}$. Furthermore, for each $t \in [0,T]$,

$$N_{ji}(t) = \sup\left\{n \geq 0: \sum_{k=1}^{n} w(k,j,i) \leq t\right\}.$$

Note that, by the claim proved in Lemma 8.4 of [42], we know that

$$\Xi^r(T) \to 0 \qquad a.s. \quad as \quad r \to \infty, \tag{5.14}$$

which implies that there is a constant $K_{L+1} > 0$ such that

$$\inf_r P\{\mathcal{T}^{r,L+1}\} \geq 1 - \frac{\eta}{2(L+2)}, \tag{5.15}$$

where

$$\mathcal{T}^{r,L+1} \equiv \{\omega: \| \Xi^r(T)\|_2 \leq K_{L+1}\}.$$

Thus, it follows from Equation (5.12) that, along each sample path in $\mathcal{S}_L \cap \mathcal{T}^{r,n} \cap \mathcal{T}^{r,L+1}$ for $r \in \{1,2,...\}$ and $n \in \{0,1,...,L+1\}$,

$$\begin{aligned}\left\|\hat{E}^r(t)\right\|_2 &\leq \left\|\hat{E}^r(0)\right\|_2 + \text{Osc}\left(\hat{E}^r, [0,t]\right) \\ &\leq 2\sum_{n=0}^{L+1} \sup_{t\in[0,T]\cap[0,\sigma_n)} \left\|\tilde{E}^{r,n}(t)\right\|_2,\end{aligned} \tag{5.16}$$

where

$$\tilde{E}^{r,L+1}(t) \equiv \Xi^r(T) \quad \text{for all} \quad t \in [0, T].$$

Hence, for the above arbitrarily given $\eta > 0$, each rational $t \in [0, T]$, and sufficiently large $r \in \mathcal{R}$, we know that

$$\begin{aligned} P\left\{ \left\| \hat{E}^r(t) \right\|_2 \leq 2^{L+1} \sum_{n=0}^{L+1} K_n \right\} & \\ \geq P\left\{ \left\{ \left\| \hat{E}^r(t) \right\|_2 \leq 2^{L+1} \sum_{n=0}^{L+1} K_n \right\} \bigcap \mathcal{S}_L \right\} & \\ \geq P\{\mathcal{S}_L\} - \sum_{n=0}^{L+1} P\left\{ \left\{ \| \tilde{E}^{r,n}(t) \|_2 > K_n \right\} \bigcap \mathcal{S}_L \text{ for some } t \in [0, T] \cap [0, \sigma_n) \right\} & \\ > 1 - \eta, & \end{aligned} \tag{5.17}$$

where, $[0, T] \cap [0, \sigma_{L+1})$ is defined to be $[0, T]$. Furthermore, the second inequality in Equation (5.17) follows from Equation (5.16) and the last inequality in Equation (5.17) follows from Equations (5.8) and (5.9). Thus, we have shown that the condition in Equation (5.5) holds.

Next, we prove the condition in Equation (5.6) to be true. It follows from Equations (5.2) and (5.14) that, for each $\eta > 0$ and $T > 0$, there exists a $\delta_n > 0$ for each $n \in \{0, 1, ..., L + 1\}$ such that

$$\limsup_{r \to \infty} P\left\{ \varpi(\tilde{E}^{r,n}, \delta_n, [0, T] \cap [0, \sigma_n)) \geq \frac{\eta}{L+2} \right\} \leq \frac{\eta}{L+2}. \tag{5.18}$$

Furthermore, if we take $\delta = \min\{\delta_0, ..., \delta_{L+1}\} > 0$, it follows from Equations (5.3) and (5.12) and 1.9 in page 326 of Jacod and Shiryaev [44] that

$$\begin{aligned} \varpi(\hat{E}^r, \delta, T) &\leq \sum_{n=0}^{L+1} \varpi(\tilde{E}^{r,n}, \delta, [0, T] \cap [0, \sigma_n)) \\ &\leq \sum_{n=0}^{L+1} \varpi(\tilde{E}^{r,n}, \delta_n, [0, T] \cap [0, \sigma_n)) \end{aligned} \tag{5.19}$$

along each sample path in S_L for each $r \in \{1, 2, ...,\}$. Thus, for each sufficiently large $r \in \mathcal{R}$, it follows from Equations (5.18)–(5.19) that

$$P\left\{ \varpi(\hat{E}^r, \delta, T) \geq 2^{L+1} \frac{\eta}{L+2} \right\} \leq \sum_{n=0}^{L+1} P\left\{ \varpi(\tilde{E}^{r,n}, \delta_n, [0, T] \cap [0, \sigma_n)) \geq \frac{\eta}{L+2} \right\} \leq \eta,$$

or in other words, the condition in Equation (5.6) holds.

Hence, by the proved claims in Equations (5.5) and (5.6), we know that $\hat{E}^r(\cdot)$ is relatively compact for $r \in \mathcal{R}$. Furthermore, we consider any subsequence $\mathcal{R}_1 \subseteq \mathcal{R}$ such that

$$\hat{E}^r(\cdot) \Rightarrow \hat{E}(\cdot) \quad \text{(a process to be identified) along} \quad r \in \mathcal{R}_1. \tag{5.20}$$

Then, by the Skorohod representation theorem (see, e.g. Theorem 3.1.8 in page 102 of Ethier and Kurtz [43]) and the random change of time lemma (see, e.g. page 151 of Billingsley [41]), we can conclude that

$$\left(\hat{E}^r(\cdot)I_{\{\cdot\leq\tau_{n+1}\}}, \hat{E}^r(\cdot)I_{\{\cdot\leq\tau_n\}}\right) \Rightarrow \left(\hat{E}(\cdot)I_{\{\cdot\leq\tau_{n+1}\}}, \hat{E}(\cdot)I_{\{\cdot\leq\tau_n\}}\right).$$

for each $n \in \{0, 1, \ldots,\}$ along $r \in \mathcal{R}_1$. Thus, by the method of induction in terms of $n \in \{0, 1, \ldots,\}$, Equation (5.2), and the continuous-mapping theorem (see, e.g. Theorem 3.4.1 in page 85 of Whitt [19]), we know that the limit in Equation (5.20) along $r \in \mathcal{R}_1$ is $H^E(\cdot)$. In addition, we can conclude that $\hat{E}^r \Rightarrow H^E(\cdot)$ along $r \in \mathcal{R}$ since $\mathcal{R}_1$ is arbitrarily chosen. Finally, by the independence assumptions, we know that the claim in Lemma 4.1 holds.

Disclosure statement

No potential conflict of interest was reported by the author.

Funding

The author gratefully acknowledges that the project is funded by National Natural Science Foundation of China with Grant No. [11771006], Grant No. [10971249], and Grant No. [11371010].

References

[1] K. Schwab, *The Fourth Industrial Revolution*, World Economic Forum, Cologny, Switzerland, 2016.

[2] N. Dedic and C. Stanier, *Towards Differentiating Business Intelligence, Big Data, Data Analytics and Knowledge Discovery*, Vol. 285, Heidelberg: Springer International Publishing, Berlin, 2017.

[3] C. Snijders, U. Matzat, and U.D. Reips, *'Big Data': Big gaps of knowledge in the field of Internet*, Int. J. Internet Sci. 7 (2012), pp. 1–5.

[4] G. Santucci The Internet of things: Between the revolution of the Internet and the metamorphosis of objects, *European Commission Community Research and Development Information Service*, October 23 2016.

[5] W. Dai, *Optimal rate scheduling via utility-maximization for J-user MIMO Markov fading wireless channels with cooperation*, Oper. Res. 61 (6) (2013), pp. 1450–1462 (with 26 page online e-companion (Supplemental)). doi:10.1287/opre.2013.1224.

[6] H. Viswanathan and K. Kumaran, *Rate scheduling in multiple antenna downlink wireless systems*, IEEE Trans. Commun. 53 (4) (2005), pp. 645–655. doi:10.1109/TCOMM.2005.844961.

[7] H.S. Wang and N. Moayeri, *Finite-state Markov channel – A useful model for radio communication channels*, IEEE Trans. Vehicular Technol. 44 (1) (1995), pp. 163–171. doi:10.1109/25.350282.

[8] M. Iansiti and K.R. Lakehani, *The truth about Blockchain*, Harv. Bus. Rev. (January 2017).

[9] S. Nakamoto, *A Peer-To-Peer Electronic Cash System*, 2013.

[10] V. Buterin, *Ethereum: A next-generation smart contract and decentralized application platform*, (2013). Available at http://ethereum.org/ethereum.html

[11] P. Schueffel, *Taming the Beast: A Scientific Definition of Fintech*, J. Innovation Manag. 4 (2016), pp. 45.

[12] W. Dai and Q. Jiang, *Stochastic optimal control of ATO systems with bat arrivals via diffusion approximation*, Probab. Eng. Informat. Sci. 21 (3) (2007), pp. 477–495. doi:10.1017/S0269964807000095.

[13] Economist, Data, data everywhere, *The Economist*, 25 February 2010 (2011).

[14] A. De Mauro, M. Greco, and M. Grimaldi, *A formal definition of big data based on its essential features*, Lib. Rev. 65 (2016), pp. 122–135. doi:10.1108/LR-06-2015-0061.

[15] W. Fischer and K.M. Hellstern, *The Markov-modulated Poisson process (MMPP) cookbook*, Perform. Eval. 18 (1992), pp. 149–171. doi:10.1016/0166-5316(93)90035-S.

[16] A. Cetinkaya and T. Hayakawa, *Stability of switched stochastic dynamical system driven by Brownian motion and Markov modulated compound Poisson process*, 2011 American Control Conference, San Francisco, CA, 2011.

[17] H. Ye and D.D. Yao, *Heavy traffic optimality of a stochastic network under utility-maximizing resource control*, Oper. Res. 56 (2) (2008), pp. 453–470. doi:10.1287/opre.1070.0455.

[18] S. Bhardwaj, R.J. Williams, and A.S. Acampora, *On the performance of a two-user MIMO downlink system in heavy traffic*, IEEE Trans. Inf. Theory 53 (5) (2007), pp. 1851–1859. doi:10.1109/TIT.2007.894662.

[19] W. Whitt, *Stochastic-Processes Limits*, Springer, New York, 2002.

[20] H.J. Kushner, *Heavy Traffic Analysis of Controlled Queueing and Communication Networks*, Springer-Verlag, New York, 2001.

[21] J.H. Shapiro, et al., *Quantum Computation and Communication - Optical and Quantum Communications-20*, RLE Prog. Rep. 145 (2003), pp. 20–21.

[22] A.M. Childs, D.W. Leung, and H.K. Lo, *Two-way quantum communication channels*, Int. J. Quantum Inf. 4 (1) (2005), pp. 63–83. doi:10.1142/S0219749906001621.

[23] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Inc, Chichester, 1991.

[24] A. Goldsmith, S.A. Jafar, N. Jindal, and N. Vishwanath, *Capacity limits of MIMO Channels*, IEEE J Sel Areas Commun. 21 (5) (2003), pp. 684–702. doi:10.1109/JSAC.2003.810294.

[25] J.F. Nash, *Equilibrium Points in N-person Games*, Proc. Natl. Acad. Sci. 36 (36) (1950), pp. 48C9. doi:10.1073/pnas.36.1.48.

[26] J.R. Rosen, *Existence and uniqueness of equilibrium points for concave N-person games*, Econometrics. 33 (3) (1965), pp. 520–534. doi:10.2307/1911749.

[27] J.M. Harrison, *A broader view of Brownian networks*, Ann. Appl. Probab. 13 (3) (2003), pp. 1119–1150. doi:10.1214/aoap/1060202837.

[28] S. Asmussen, P. Glynn, and J. Pitman, *Discretization error in simulation of one-dimensional reflection Brownian motion*, Ann. Appl. Probab. 5 (4) (1995), pp. 875–996. doi:10.1214/aoap/1177004597.

[29] M. Mousavi and P.W. Glynn, *Exact simulation of non-stationary reflected Brownian motion*, (2013). Available at SSRN http://ssrn.com/abstract=2373347

[30] R. Atar, A. Mandelbaum, and M. Reiman, *Scheduling a multi-class queue with many exponential servers: Asymptotic optimality in heavy-traffic*, Ann. Appl. Probab. 14 (3) (2004), pp. 1084–1134. doi:10.1214/105051604000000233.

[31] S.I. Resnick, *Adventures in Stochastic Processes*, Birkhäuser, Boston, 1992.

[32] D. Rajan and M. Visser. *Quantum Blockchain using entanglement in time*, (2018) Available at https://arxiv.org/abs/1804.05979.

[33] Wikipedia, *Supercomputer*, (2018). Available at https://en.wikipedia.org/wiki/Supercomputer.

[34] Artemis, *Internet of energy for electric mobility*, (2018). Available at http://www.artemis-ioe.eu/.

[35] W. Dai, *Internet of energy with multiple cloud-computing service centers and artificial intelligence*. Invited Plenary Talk at IEEE 7th International Conference on Power and Energy Systems, Toronto, Canada, 2017.

[36] T. Paraskova. *OilCoin: The world's first compliant cryptocurrency*, (2017). Available at https://oilprice.com/Energy/Crude-Oil/OilCoin-Worlds-First-Compliant-Cryptocurrency.html.

[37] N. Jindal, S. Vishwanath, and A. Goldsmith, *On the duality of Gaussian multiple-access and broadcast channels*, IEEE Trans. Inf. Theory. 50 (5) (2004), pp. 768–783. doi:10.1109/TIT.2004.826646.

[38] J.M. Harrison, *Brownian Motion and Stochastic Flow Systems*, John Wiley & Sons, New York, 1985.

[39] W. Dai, *Optimal control with monotonicity constraints for a parallel-server loss channel serving multi-class jobs*, Math. Comput. Model. Dyn.L Syst. 20 (3) (2014), pp. 284–315. doi:10.1080/13873954.2013.831359.

[40] D.G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed., Addison-Wesley Publishing Company, Reading, MA, 1984.

[41] P. Billingsley, *Convergence of Probability Measures*, 2nd ed., John Wiley & Sons, New York, 1999.

[42] J.G. Dai and W. Dai, *A heavy traffic limit theorem for a class of open queueing networks with finite buffers*, Queueing Syst. 32 (1–3) (1999), pp. 5–40. doi:10.1023/A:1019178802391.

[43] S.N. Ethier and T.G. Kurtz, *Markov Processes: Characterization and Convergence*, John Wiley & Sons Inc, New York, 1986.

[44] J. Jacod and A.N. Shiryaev, *Limit Theorems for Stochastic Processes*, 2nd ed., Springer-Verlag, Berlin, 2003.