JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **17**, Number **4**, July **2021**

doi:10.3934/jimo.2020049

pp. **1845–1861**

A STOCHASTIC MODEL AND SOCIAL OPTIMIZATION OF A BLOCKCHAIN SYSTEM BASED ON A GENERAL LIMITED BATCH SERVICE QUEUE

Wenjuan Zhao and Shunfu Jin*

School of Information Science and Engineering, Yanshan University
Qinhuangdao 066004, China

Wuyi Yue

Department of Intelligence and Informatics, Konan University
Kobe 658-8501, Japan

(Communicated by Shoji Kasahara)

Abstract. Blockchain is well known as a database technology supporting digital currencies such as Bitcoin, Ether and Ripple. For the purpose of maximizing the overall revenue of the blockchain system, we propose a pricing policy to impose on transactions. Regarding the mining process as a vacation, and the block-verification process as a service, we establish a type of non-exhaustive queueing model with a limited batch service and a possible zero-transaction service. By selecting the beginning instant of a block-verification process as a Markov point and using the method of a generating function, we obtain the stationary probability distribution for the number of transactions in the system at the Markov points and analyze the elapsed time for the mining cycle. Based on the model analysis results, we derive the average latency of transactions and demonstrate how the average latency of transactions changes in relation to the arrival rate of transactions. With a reward-cost structure, we construct an individual benefit function and a social benefit function. By improving the Grasshopper Optimization Algorithm (GOA), we search for the Nash equilibrium and the socially optimal arrival rates of transactions. Numerical results show that the Nash equilibrium arrival rate of transactions is always higher than the socially optimal arrival rate of transactions for a given mining parameter and a specific block capacity. For this, we propose a pricing policy that forces the transactions to accept the socially optimal arrival rate and maximize the overall revenue of the blockchain system, including all transactions and miners.

1. **Introduction.** Blockchain is a decentralized distributed ledger that does not allow deletion of data [15]. Compared with traditional accounting techniques, blockchain has many obvious advantages, such as irreversibility, anonymity and autonomy [10]. Since the terminology of blockchain was first presented by Nakamoto in [8], blockchain has enjoyed sustained development. In recent years, considerable efforts have been devoted to the study of the blockchain system.

2020 *Mathematics Subject Classification.* Primary: 60K15, 60K25; Secondary: 68M14.

Key words and phrases. Blockchain, limited batch service, Markov chain, Nash equilibrium, social optimization, intelligent optimization algorithm, pricing policy.

* Corresponding author: Shunfu Jin.

One of the central topics in the field of the blockchain system research is application technology. In [17], Zhao et al. proposed a lightweight backup and efficient recovery scheme for keys of health blockchain in body sensor networks (BSNs). The main advantage of the proposed scheme is its ability to secure private physiological data on the health blockchain. In [9], Novo presented a new architecture that is a generic, transparent and has a manageable access control system for the Internet of Things (IoT) based on blockchain technology. The architecture addressed the scalability problem of managing access to numerous constrained IoT devices and compelled the constrained networks to simultaneously connect to the blockchain network using specific nodes. In [13], with the help of the blockchain technology, Muhamed et al. proposed a global higher education credit platform, named EduCTX. The establishment of this platform was the first step toward a higher education system with transparency and technical advancement. However, one important issue that has been overlooked by the afore-mentioned research is how to evaluate and improve the performance of the blockchain system.

From the view point of performance evaluation, some works have been carried out on the blockchain system based on queueing theory. In [4, 5], Kasahara et al. established a single-server queue with a batch service and a priority mechanism based on the Bitcoin system. By using the method of a supplementary variable, they derived the average transaction-confirmation time. With numerical experiment results, they quantitatively evaluated the effects of the block size on the transaction-confirmation time. However, in the above-mentioned research on the blockchain system, neither the process for solving the puzzle-like problem, nor the implementation of the coinbase was taken into account.

In [6], Li et al. built a Markovian batch-service queueing system with two different service stages and derived the stationary probability vector of the system. They obtained formulas for the average number of transactions in the queue, the average number of transactions in a block, and the average confirmation time of transactions. Unfortunately, the coinbase transaction was also omitted in this model, and the model analysis was short on generality due to the assumption of an exponentially distributed service time.

It is noteworthy that in [14], Vlasiou investigated the Lindley-type equation by considering a system consisting of one server and two service points. According to the relation between the waiting times of successive customers, she derived a closed-form expression for the steady-state limiting distribution of the waiting time with sufficient assumptions. Lindley-type equation provides a potentially promising method to be applied to the model analysis of non-exhaustive queues and performance evaluation of the blockchain system.

It is a meaningful challenge to establish a system model that is more accordant with a practical blockchain system. Compared with our previous work [18], there is a substantial and appropriate extension in this paper. In our previous work, we investigate the average number of transactions and the average latency of transactions in the blockchain system. In this paper, based on the working flow of a mining cycle, we show how to model the blockchain system and how to evaluate the system performance. In addition, we present the formulation of a pricing policy with a reasonable remittance fee to socially optimize the blockchain system, as well as provide numerical results to verify the correctness of the modeling approach and the rationality of the pricing policy.

The main contributions of this paper are as follows.

(1) Regarding the mining process as a vacation and the block-verification process as a service, we establish a type of non-exhaustive queueing model with a limited batch service and a possible zero-transaction service. By relaxing the assumption of exponential distribution for service time, our proposed model is more general in practice.
(2) By selecting the beginning instant of a block-verification process as a Markov point and calculating the elapsed time for a mining cycle, we analyze the proposed model. Compared with the regeneration cycle method widely used in analyzing a non-exhaustive service vacation model, the analysis method employed in this paper is more concise and efficient.
(3) With the enhanced GOA algorithm, we give the Nash equilibrium and the socially optimal arrival rates of transactions, and then we present an appropriate remittance fee charged to transactions for maximizing the overall revenue of the blockchain system. This is the first work on the Nash equilibrium of transactions in the blockchain system based on queueing theory.

The rest of this paper is organized as follows. In Section 2, based on the mining cycle in the blockchain system, we establish a type of non-exhaustive queueing model with a limited batch service and a possible zero-transaction service. In Section 3, we carry out an analysis of the system model and derive the average latency of transactions. In Section 4, with numerical results, we present a pricing policy forcing the transactions to accept the socially optimal arrival rate. In Section 5, we summarize the conclusions.

2. **Blockchain system and mathematical model.**

In this section, we discuss the mining cycle in the blockchain system. Accordingly, we establish a type of non-exhaustive queueing model with a limited batch service and a possible zero-transaction service.

2.1. **Mining cycle in the blockchain system.**

Transactions, nodes and blocks are the basic components of a blockchain system. A blockchain system originates from the generation of a Genesis Block. The volunteer nodes, which temporally store authenticated transactions in a Transaction Memory Pool (Tx Mem Pool), are called miners. Miners compete for the right to add a new block to the blockchain by solving a puzzle-like problem. This competition process is called the mining process. Obviously, the time taken to complete the mining process increases with the level of difficulty of the puzzle-like problem. To measure the time duration of a mining process, we introduce a mining parameter. The mechanism of Proof of Work (PoW) is adopted to realize the determination of competition results. Finding the solution to the puzzle-like problem means the generation of a new block. The first miner that finds the solution to the puzzle-like problem is called the winning miner, whereas the other miners are called non-winning miners. If a newly generated block is empty, the empty block is directly connected to the blockchain without verification. Otherwise, the winning miner broadcasts the newly generated block to all the non-winning miners. Non-winning miners receiving the broadcast message verify the newly generated block and feedback validation results to the winning miner. Once the validation succeeds, the newly generated block will be connected to the blockchain. And then the blockchain system synchronizes the blockchain and updates all Tx Mem Pools. For a block with transactions, the block-verification process includes the validation process and the connection process. The winning miner receives rewards with the coinbase, a special transaction structured by the nodes

to reward the winning miner for the contribution, and the remittance fee charged to transactions in the newly generated block. For an empty block, the connection process is just the block-verification process and the winning miner receives only one reward with the coinbase.

A mining process and the subsequent block-verification process combine to constitute a mining cycle. The operation of a mining cycle is collectively performed by multiple nodes distributed in different places to avoid the possibility of a winning miner being controlled or bribed. All the nodes in the blockchain are equal, and any of the nodes can be selected as a winning miner. All the nodes in the blockchain are independent with each other and keep a complete blockchain to ensure the security of data. The working flow of a mining cycle [2, 7] is illustrated in Fig. 1.

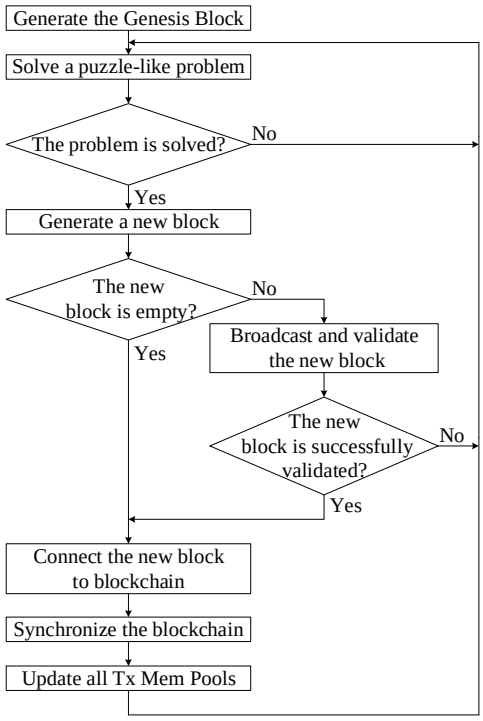


FIGURE 1. The working flow of a mining cycle.

Based on the working flow of a mining cycle illustrated in Fig. 1, we note that the average latency of transactions is dependent of a variety of factors, such as the block capacity, the mining parameter, the arrival rate of transactions, and so on. In this paper, the latency of a transaction refers to the time duration from the instant when a transaction arrives at Tx Mem Pools to the instant that this transaction is connected to the blockchain. In order to quantitatively evaluate the response performance of a blockchain system, we need to establish a mathematical model to more realistically capture the stochastic behaviors of the transactions.

2.2. **Mathematical model.** Note that the size of a block is limited and the block can not be confirmed during the mining process in a blockchain system. A non-exhaustive queue [5] is naturally suitable to capture the working flow of a mining

cycle. We also note that the block-verification process includes the validation process and the connection process. The validation process consists of the validation process of transactions and the validation process of the basic block. When the number of transactions in the newly generated block gets bigger, the validation process of transactions will last longer.

In this paper, we limit our research to the blockchian system with light transaction traffic, and neglect the influence for the number of transactions on the time duration of the block-verification process. Considering that transactions in the newly generated block are simultaneously validated and a newly generated block is possibly empty, we establish a type of non-exhaustive queueing model with a limited batch service and a possible zero-transaction service. In this queueing model, the mining process is regarded as a vacation, and the block-verification process is regarded as a service.

In this queueing model, once a newly generated block is connected to the blockchain, i.e. a block-verification period ends, a mining process will start, no matter whether or not there are transactions waiting in the system. Moreover, when a puzzle-like problem is solved, i.e. a mining process ends, if the newly generated block is empty, a zero-transaction will start; otherwise, a normal service period will start.

A winning miner will be selected from all the potential miners distributed over the whole blockchain system. Therefore, the consensus algorithm based selection process makes it difficult for a minority of nodes to control the whole system. In order to capture the features of decentralization and distribution in the blockchain system, we build a queueing system model to regard the whole blockchain system as a server pool. In each mining cycle, one of the miners will be promoted to be a wining miner who will act as a server in this queueing model.

We assume that the transaction arrivals follow a Poisson process with the parameter λ ($\lambda > 0$).

We assume that the time duration V for a mining process is an independent and identically distributed (i.i.d) random variable and follows a general distribution with a distribution function $V(t)$. The Laplace-Stieltjes Transform (LST) $V^*(s)$, the mean value $E[V]$ and the second moment $E[V^2]$ of the time duration V for a mining process are given as follows:

$$V^*(s) = \int_0^\infty e^{-st}\,\mathrm{d}V(t), \quad E[V] = \frac{1}{\theta} = \int_0^\infty t\,\mathrm{d}V(t), \quad E[V^2] = \int_0^\infty t^2\,\mathrm{d}V(t)$$

where θ is the mining parameter.

We assume that the time duration S for a block-verification process is i.i.d random variable and follows a general distribution with a distribution function $S(t)$. The LST $S^*(s)$, the mean value $E[S]$ and the second moment $E[S^2]$ of the time duration S for a block-verification process are given as follows:

$$S^*(s) = \int_0^\infty e^{-st}\,\mathrm{d}S(t), \quad E[S] = \frac{1}{\mu} = \int_0^\infty t\,\mathrm{d}S(t), \quad E[S^2] = \int_0^\infty t^2\,\mathrm{d}S(t)$$

where μ is the block-verification rate.

Let b be the block capacity. We can intuitively explain the stability condition of the system as follows: the expected number of transactions arriving during a mining cycle must be smaller than b. That is,

$$\lambda(E[S] + E[V]) < b.$$

3. **Model analysis.** In this section, we first employ the method of using an embedded Markov chain to calculate the probability generating function $Q(z)$ for the number of transactions in the system at the Markov points, and then we derive the average latency of transactions by analyzing the elapsed time for a mining cycle.

3.1. **The probability generating function $Q(z)$.** For a queueing system with general service, the numbers of transactions at instants t, $t \geq 0$ do not have the Markov property. We employ the method of the embedded Markov chain to analyze the queueing model.

The operation of the queueing model with general limited batch service is illustrated in Fig. 2.

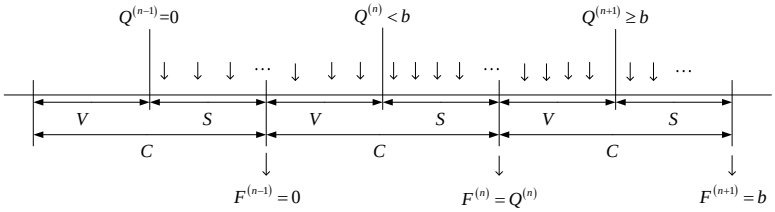


FIGURE 2. Operation of the queueing model with general limited batch service.

In this figure, V represents the time duration of a mining process, S represents the time duration of a block-verification process, C represents the time duration of a mining cycle, $F^{(n)}$ represents the number of transactions getting service during the nth block-verification process, and $Q^{(n)}$ represents the number of transactions in the system at the beginning instant of the nth block-verification process.

We note that the numbers of transactions in the system at beginning instants of the block-verification processes constitute a Markov chain $\{Q^{(n)}, n \geqslant 1\}$. These instants are called the embedded Markov points. The transition probability of this Markov chain is given as follows:

$$\begin{aligned} p_{jk} &= P\left\{Q^{(n+1)} = k \mid Q^{(n)} = j\right\} \\ &= \begin{cases} \displaystyle\int_0^\infty \frac{(\lambda t)^k}{k!} e^{-\lambda t}\, \mathrm{d}S * V(t), & j < b \\ \displaystyle\int_0^\infty \frac{(\lambda t)^{k-j+b}}{(k-j+b)!} e^{-\lambda t}\, \mathrm{d}S * V(t), & b \leq j < b+k \\ 0, & j \geq b+k \end{cases} \end{aligned} \tag{1}$$

where $S * V(t)$ is the distribution function for the convolution of S with V.

Let q_k be the probability for the number of transactions at the Markov points being equal to k in the steady state, i.e.,

$$q_k = \lim_{n\to\infty} P\left\{Q^{(n)} = k\right\}, \quad k \geq 0.$$

It follows that

$$\begin{aligned} q_k = &\sum_{j=0}^{b-1} q_j \int_0^\infty \frac{(\lambda t)^k}{k!} e^{-\lambda t}\, \mathrm{d}S * V(t) \\ &+ \sum_{j=b}^{k+b} q_j \int_0^\infty \frac{(\lambda t)^{k-j+b}}{(k-j+b)!} e^{-\lambda t}\, \mathrm{d}S * V(t), \quad k \geq 0. \end{aligned} \tag{2}$$

The probability generating function $Q(z)$ for the number of transactions at the regeneration point in the steady state is given as follows:

$$\begin{aligned} Q(z) = &\sum_{k=0}^{\infty} z^k \sum_{j=0}^{b-1} q_j \int_0^\infty \frac{(\lambda t)^k}{k!} e^{-\lambda t}\, \mathrm{d}S * V(t) \\ &+ \sum_{k=0}^{\infty} z^k \sum_{j=b}^{k+b} q_j \int_0^\infty \frac{(\lambda t)^{k-j+b}}{(k-j+b)!} e^{-\lambda t}\, \mathrm{d}S * V(t). \end{aligned} \tag{3}$$

Note that the time duration V of a mining process depends on the difficulty level of the puzzle-like problem, whereas the time duration S of a block-verification process is relevant to the validation process and the connection process of the newly generated block. Therefore, we assume that the time duration V of a mining process is independent of the time duration S of a block-verification process. Based on this assumption, Eq. (3) can be simplified as follows:

$$Q(z) = \frac{1}{z^b} S^*(\lambda(1-z)) V^*(\lambda(1-z)) \left(\sum_{j=0}^{b-1} q_j z^b + Q(z) - Q_b(z) \right) \tag{4}$$

where

$$Q_b(z) = \sum_{k=0}^{b-1} q_k z^k.$$

Simplifying Eq. (4), we get

$$Q(z) = \frac{S^*(\lambda(1-z)) V^*(\lambda(1-z)) (Q_b(1) z^b - Q_b(z))}{z^b - S^*(\lambda(1-z)) V^*(\lambda(1-z))}. \tag{5}$$

To determine $Q(z)$, we need to compute the coefficients $\{q_0, q_1, \ldots, q_{b-1}\}$ of $Q_b(z)$.

In the denominator on the right-hand side (r.h.s.) of Eq. (5), we introduce notations as follows:

$$f(z) = z^b, \quad g(z) = -S^*(\lambda(1-z)) V^*(\lambda(1-z)).$$

Using Rouche's theorem [3] and Lagrange's theorem [1], it can be proved that $|f(z)| > |g(z)|$ on the circle $|z| = 1 + \varepsilon$ for $\varepsilon > 0$, and that $f(z)$ and $f(z) + g(z)$ have the same number of zeros inside $|z| = 1 + \varepsilon$. Therefore, the denominator on the r.h.s. of Eq. (5) has b roots inside $|z| = 1 + \varepsilon$. One of these roots is $z = 1$, and the other $b - 1$ roots are given as follows:

$$z_r = \sum_{n=1}^{\infty} \frac{e^{\frac{2\pi r n}{b} i}}{n!} \frac{d^{n-1}}{dz^{n-1}} (S^*(\lambda(1-z)) V^*(\lambda(1-z)))^{n/b} \Big|_{z=0}, \quad r = 1, 2, \ldots, b-1 \tag{6}$$

where $i = \sqrt{-1}$.

Since $Q(z)$ is analytic in $|z| \leq 1$, the numerator on the r.h.s. of Eq. (5) must also be zero at $z = z_r$ for $r = 1, 2, \ldots, b-1$. Therefore, we have $b-1$ equations as

$$\sum_{k=1}^{b-1} q_k z_r^b - \sum_{k=1}^{b-1} q_k z_r^k = 0, \quad r = 1, 2, \ldots, b-1. \tag{7}$$

Letting $z \to 1$ and using L'Hopital rule in Eq. (5), we get

$$1 = \frac{bQ_b(1) - Q_b^{'}(1)}{b - \lambda E[S] - \lambda E[V]} \tag{8}$$

where $Q_b^{'}(1)$ is the first derivative of $Q_b(z)$ at $z = 1$.

Rearranging Eq. (8) yields

$$\sum_{k=1}^{b-1} (b-k) q_k = b - \lambda E[S] - \lambda E[V]. \tag{9}$$

Based on Eqs. (7) and (9), we numerically compute the coefficients $\{q_0, q_1, \ldots, q_{b-1}\}$ of $Q_b(z)$. Furthermore, we obtain the probability generating function $Q(z)$ for the number of transactions at the Markov points in the steady state.

3.2. **Average latency $E[T]$ of transactions.** A mining process and the subsequent block-verification process combine to constitute a mining cycle. Let C be the time duration of a mining cycle. Based on the assumption that the time duration V of a mining process and the time duration S of a block-verification process are independent of each other, the LST $C^*(s)$ and the mean value $E[C]$ for the time duration C of a mining cycle are given as follows:

$$C^*(s) = S^*(s)V^*(s), \quad E[C] = E[S] + E[V].$$

The probability generating function $A_C(z)$ for the number of transactions arriving during a mining cycle is given as follows:

$$A_C(z) = S^*(\lambda(1-z))V^*(\lambda(1-z)). \tag{10}$$

Let D be the elapsed time for a mining cycle. Referencing [11], the probability density function $h(t)$ for the elapsed time D of a mining cycle is given as follows:

$$h(t) = \frac{1}{E[C]}(1 - C(t))$$

where $C(t)$ is the distribution function for the time duration C of a mining cycle.

The probability generating function $A_D(z)$ for the number of transactions arriving during the elapsed time D of a mining cycle is given as follows:

$$\begin{aligned} A_D(z) &= \sum_{i=0}^{\infty} z^i \int_0^{\infty} \frac{(\lambda t)^i}{i!} e^{-\lambda t} h(t) dt \\ &= \frac{1 - C^*(\lambda(1-z))}{\lambda(1-z)E[C]}. \end{aligned} \tag{11}$$

The probability generating function $L_s(z)$ for the number of transactions to be verified at the beginning instant of a mining cycle is given as follows:

$$\begin{aligned} L_s(z) &= \frac{Q(z)}{V^*(\lambda(1-z))} \\ &= \frac{S^*(\lambda(1-z))(Q_b(1)z^b - Q_b(z))}{z^b - C^*(\lambda(1-z))}. \end{aligned} \tag{12}$$

We note that the number of transactions at any moment within a mining cycle is the sum of the number of transactions at the beginning instant of a mining cycle and the number of transactions arriving during the elapsed time of the same mining cycle. The probability generating function $L(z)$ for the number of transactions at any moment is obtained as follows:

$$L(z) = L_s(z)A_D(z). \tag{13}$$

Substituting Eqs. (11) and (12) into Eq. (13) gives

$$L(z) = \frac{S^*(\lambda(1-z))(Q_b(1)z^b - Q_b(z))}{z^b - C^*(\lambda(1-z))} \times \frac{1 - C^*(\lambda(1-z))}{\lambda(1-z)E[C]}. \tag{14}$$

Taking the derivative of z, letting $z \to 1$ and using L'Hopital rule in Eq. (14), the average number $E[L]$ of transactions in the blockchain system is given as follows:

$$E[L] = \lambda E[S] + \frac{\lambda E[C^2]}{2E[C]} + \frac{b(b-1)(Q_b(1)-1) - Q_b^{''}(1) + \lambda^2 E[C^2]}{2(b - \lambda E[C])} \tag{15}$$

where $Q_b^{''}(1)$ is the second derivative of $Q_b(z)$ at $z = 1$ and $E[C^2]$ is the second moment of the time duration C for a mining cycle.

Following Little's law [16], the average latency $E[T]$ of a transaction is then given as follows:

$$E[T] = E[S] + \frac{E[C^2]}{2E[C]} - \frac{b(b-1)(1-Q_b(1)) + Q_b^{''}(1)}{2\lambda(b - \lambda E[C])} + \frac{\lambda E[C^2]}{2(b - \lambda E[C])}. \tag{16}$$

4. **Remittance fee charged to transactions.** In this section, we firstly investigate the Nash equilibrium and the socially optimal arrival rates of transactions. Then, we present a pricing policy that forces the transactions to accept the socially optimal arrival rate and maximizes the overall revenue of the blockchain system.

4.1. **Nash equilibrium arrival rate of transactions.** In this paper, we investigate the Nash equilibrium arrival rate from the view point of transactions. Let R be the reward of a transaction from a completed service, and β be the cost to a transaction for staying in the system. In order to guarantee that the blockchain system is stable, we formulate the maximal value λ_{max} for the arrival rate λ of transactions as follows:

$$\lambda_{max} = \frac{b}{E[S] + E[V]}. \tag{17}$$

We define the individual net benefit function U_I of a transaction as follows:

$$U_I(\lambda) = R - \beta E[T] \tag{18}$$

where $E[T]$ is the average latency of transactions given in Eq. (16).

In our model, a vacation period is certainly followed by a service period. We note that at most b transactions can be served during one service period, and all these transactions are processed simultaneously. When the arrival rate λ of transactions increases, the average latency of transactions will increase. Based on Eq. (18), we say that the individual benefit $U_I(\lambda)$ of a transaction is a decreasing function about the arrival rate λ of transactions. Provided the net benefit $U_I(\lambda)$ is positive, the arrival rate λ of transactions will be as high as possible. If there is at least one solution for the inequality $U_I \geq 0$ within the closed interval $[0, \lambda_{max}]$, the maximal value of the solutions is the Nash equilibrium arrival rate λ_e of transactions.

Otherwise, the Nash equilibrium arrival rate of transactions is $\lambda_e = 0$. We discuss the Nash equilibrium arrival rate as follows:

(1) $U_I(\lambda_{max}) \geq 0$ indicates that all the transactions join the blockchain system to get service, and the net benefit will be non-negative. Therefore, $\lambda_e = \lambda_{max}$ is the Nash equilibrium arrival rate, and no other Nash equilibrium arrival rate exists.
(2) $U_I(0) \leq 0$ indicates that no other transactions join the blockchain system, and the net benefit of a transaction that joins the blockchain system is negative. Therefore, $\lambda_e = 0$ is the Nash equilibrium arrival rate, and no other Nash equilibrium arrival rate exists.
(3) $U_I(0) > 0$ & $U_I(\lambda_{max}) < 0$ indicates that there is a unique Nash equilibrium arrival rate $0 < \lambda_e < \lambda_{max}$, where the Nash equilibrium arrival rate λ_e can be obtained by solving $U_I(\lambda_e) = 0$.

From the analysis in Section 3, we note that it is difficult to present the average latency $E[T]$ of a transaction in a closed form. Hence, it is difficult to give the Nash equilibrium arrival rate λ_e of transactions in a closed form.

Referencing [6], we assume that the time duration S for a block-verification process follows an exponential distribution with a block-verification rate μ and the time duration V for a mining process follows an exponential distribution with a mining parameter θ. With the assumption, Eq. (16) can be modified as follows:

$$E[T] = \frac{1}{\mu} + \frac{\mu^2 + \mu\theta + \theta^2}{\mu\theta(\mu + \theta)} + \frac{2\lambda^2(\mu^2 + \mu\theta + \theta^2) - \mu^2\theta^2(b(b-1)(1 - Q_b(1)) + Q_b''(1))}{2\lambda\mu\theta(\mu\theta b - \lambda\mu - \lambda\theta)}. \tag{19}$$

By setting the block-verification rate $\mu = 2$, the mining parameter $\theta = 0.5,\ 1.0,\ 1.5$, the reward $R = 15$, the cost $\beta = 1.5$ and the block capacity $b = 40,\ 80$ as an example, we carry out experiments to show the change trend for the individual benefit U_I of a transaction in relation to the arrival rate λ of transactions in Fig. 3.

In Fig. 3, we find that with the parameters set above, all the individual benefits U_I show decreasing trends as the arrival rate λ of transactions increases. We also find that all the individual benefits U_I go through $U_I = 0$, i.e., there are always values of λ_e subject to $U_I = 0$.

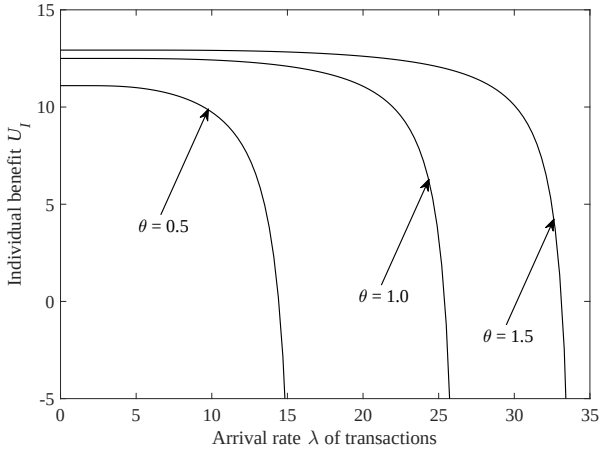
4.2. **Socially optimal arrival rate of transactions.** When designing a blockchain system, we should consider the Nash equilibrium arrival rate as well as the socially optimal arrival rate for transactions. By aggregating the individual benefits of all transactions and the coinbase reward ψ of the winning miner, we define the social benefit function U_S for a blockchain system as follows:

$$U_S(\lambda) = \lambda(R - \beta E[T]) + \frac{\psi}{E[C]}. \tag{20}$$

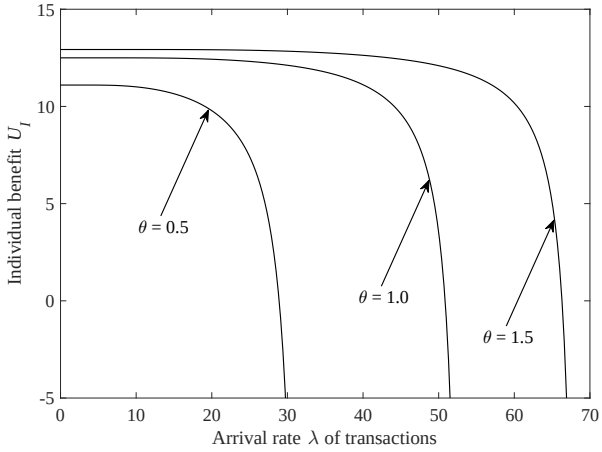
By maximizing the social benefit, the socially optimal arrival rate λ^* of transactions is given as follows:

$$\lambda^* = \underset{\lambda \in [0, \lambda_{max}]}{\arg\max}\ U_S(\lambda). \tag{21}$$

Using the same parameters as in Fig. 3 and setting the coinbase reward $\psi = 3$, we show how the social net benefit $U_S(\lambda)$ changes with respect to the arrival rate λ of transactions in Fig. 4.



(a) $b = 40$

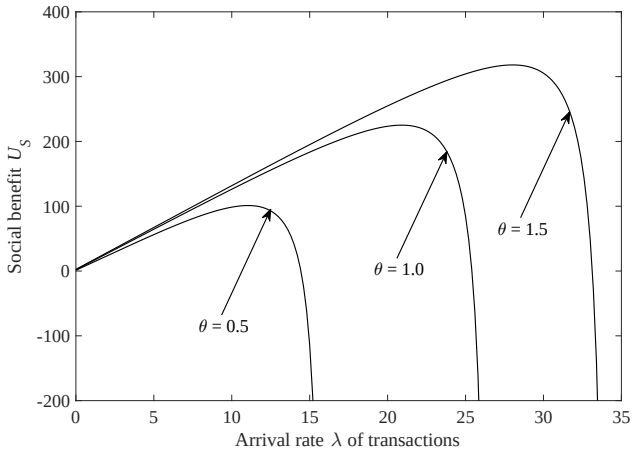


(b) $b = 80$

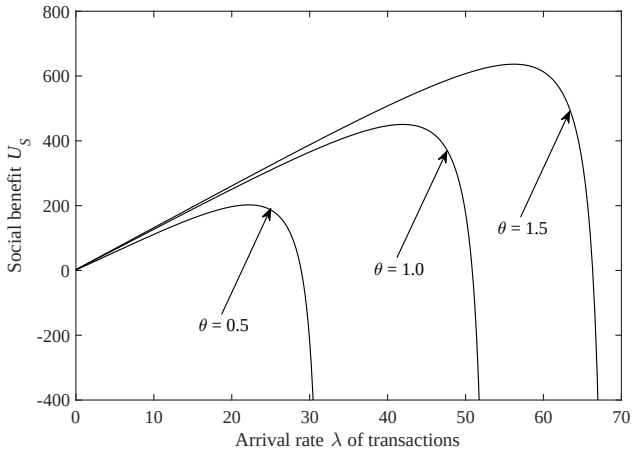
FIGURE 3. Change trend of the individual benefit $U_I(\lambda)$ of a transaction.

In Fig. 4, we find that as the arrival rate λ of transactions increases, the social benefit $U_S(\lambda)$ of transactions shows an upper convex trend. With this trend, there is always a socially optimal arrival rate λ^* of transactions and a maximal social benefit $U_S(\lambda^*)$.

In this system model, the mathematical expressions for the Nash equilibrium arrival rate λ_e and the socially optimal arrival rate λ^* of transactions are difficult to give in closed-forms. Moreover, the strict monotonicity of the social benefit is difficult to discuss and the strict differentiability of the social benefit is difficult to prove. The social optimization problem of a blockchain system involves complicated nonlinear equations and nonlinear optimization problems. In this case, when solving the optimization problem involved in the blockchain system, intelligent optimization algorithms are more efficient than conventional optimization methods, such as the



(a) $b = 40$



(b) $b = 80$

FIGURE 4. Change trend of social benefit $U_S(\lambda)$ of transactions.

steepest descent method or Newton's method. We therefore present a GOA based intelligent searching algorithm to obtain the Nash equilibrium arrival rate λ_e and the socially optimal arrival rate λ^*.

Grasshopper Optimization Algorithm (GOA) was proposed based on the behaviour of grasshopper swarms in nature by Shahrzad Saremi et al. [12]. GOA simulates the repulsion and attraction that is obliged between grasshoppers to explore the search space and exploit promising regions for solving optimization problems. In GOA, the population of grasshoppers are firstly initialized. And then, the fitness of each grasshopper is calculated and the best grasshopper (agent) is evaluated.

However, the drawbacks of GOA are slow convergence speed and poor search accuracy. For this, we propose an enhanced GOA by employing a chaos function

to initialize the population of grasshoppers, and we replace linear adaptation with cosine adaptation to update the decreasing coefficient.

The working flow of the enhanced GOA proposed in this paper is given in Fig. 5.

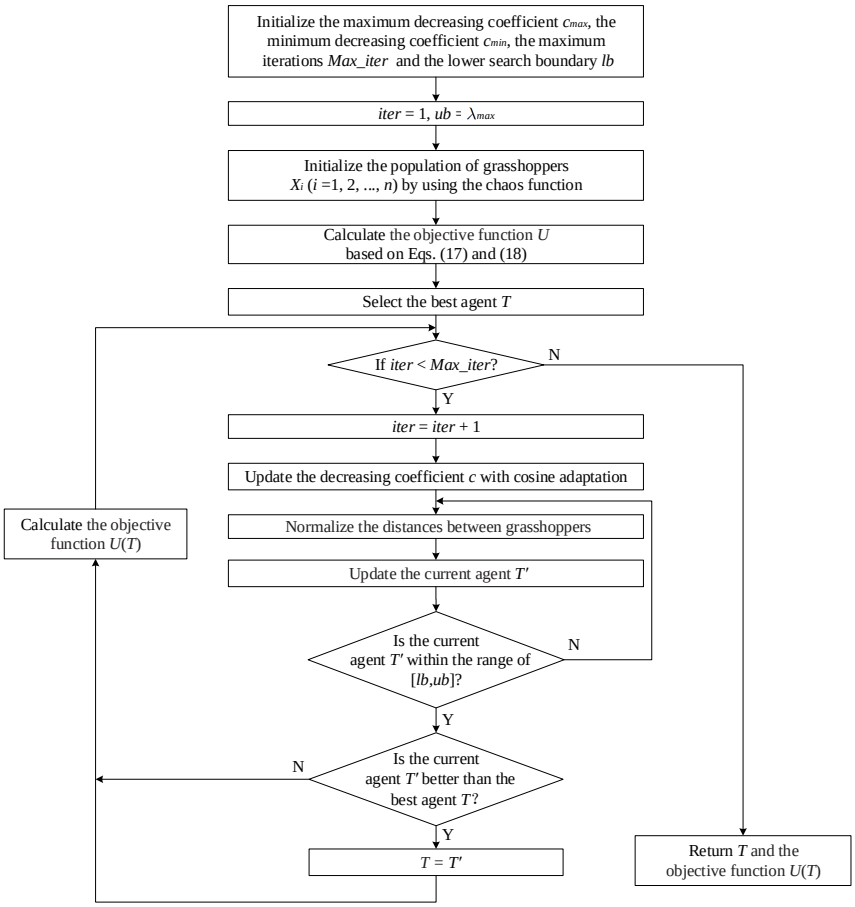
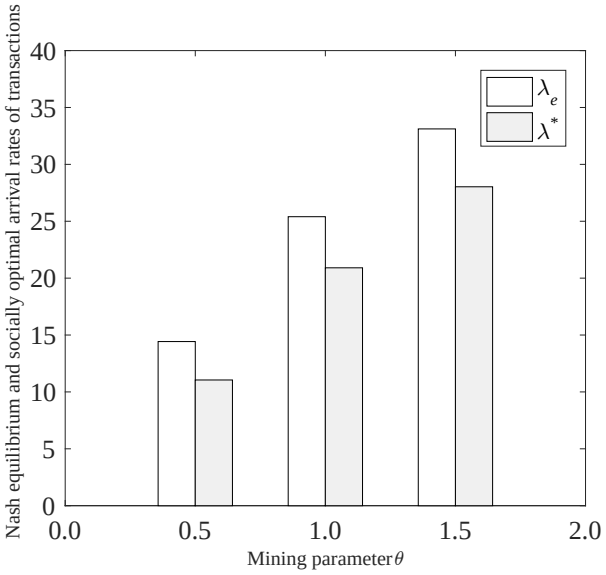


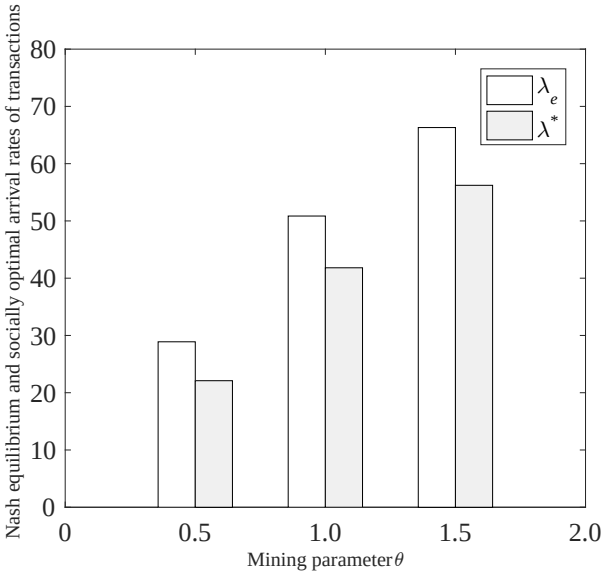
FIGURE 5. The working flow of the enhanced GOA.

4.3. **Pricing policy.** By using the same parameters as in Fig. 4, and setting the number $n = 50$ of grasshoppers, the maximum iterations $Max\_iter = 100$, the maximum decreasing coefficient $c_{max} = 1$, the minimum decreasing coefficient $c_{min} = 0.00004$, and the lower search boundary $lb = 0.0001$, we carry out experiments. The upper search boundary is equal to the maximal arrival rate λ_{max} of transactions. The value of maximal arrival rate λ_{max} of transactions is obtained by Eq. (17). The numerical results of the Nash equilibrium arrival rate λ_e of transactions and the socially optimal arrival rate λ^* of transactions with different block capacity b and different mining parameter θ are demonstrated in Fig. 6.

From Fig. 6, we observe that the Nash equilibrium arrival rate λ_e of transactions is always higher than the socially optimal arrival rate λ^* of transactions. Namely, more transactions choose to join the blockchain system under the Nash equilibrium



(a) $b = 40$



(b) $b = 80$

FIGURE 6. Nash equilibrium and socially optimal arrival rates of transactions.

arrival rate, but this will deteriorate the social benefit. In order to oblige transactions to adopt the socially optimal arrival rate, we charge an appropriate remittance fee f to transactions who join the blockchain system.

With the remittance fee f charged to transactions, the individual benefit function $U_I^f(\lambda)$ of transactions is modified as follows:

$$U_I^f(\lambda) = R - f - CE[T]. \tag{22}$$

Accordingly, the social benefit $U_S^f(\lambda)$ of transactions is given as follows:

$$\begin{aligned} U_S^f(\lambda) &= \lambda(R - f - \beta E[T]) + \lambda f + \frac{\psi}{E[C]} \\ &= \lambda(R - \beta E[T]) + \frac{\psi}{E[C]}. \end{aligned} \tag{23}$$

We note that $U_S^f(\lambda)$ is equal to $U_S(\lambda)$. That is to say, the remittance fee f charged to transactions has no effect on the social benefit. This is because that the remittance fee is just transferred from transactions to miners.

By substituting the socially optimal arrival rate λ^* of transactions into Eq. (22) and letting $U_I^f(\lambda) = 0$, we calculate the remittance fee f charged to transactions as follows:

$$f = R - CE[T]\big|_{\lambda=\lambda^*}. \tag{24}$$

By setting the block-verification rate $\mu = 2$, the reward $R = 15$, the cost $\beta = 1.5$ and the coinbase reward $\psi = 3$ as an example, we present numerical results for the remittance fee with different block capacity b and different mining parameter θ in Table 1.

TABLE 1. Numerical results for the remittance fee.

Mining parameter (θ)	Block capacity (b)	Socially optimal arrival rate (λ^*)	Maximum social benefit ($U_S(\lambda^*)$)	Remittance fee (f)
0.5	40	11.0493	101.0826	9.0397
0.5	80	22.0986	202.2892	9.0996
1.0	40	20.9083	225.1180	10.6713
1.0	80	41.8166	450.5689	10.7271
1.5	40	28.0302	318.0631	11.2554
1.5	80	56.2193	636.5389	11.2767

From Table 1, we note that for the same block capacity b, as the mining parameter θ increases, the mining process time is shorter, so a block is generated earlier. As a result, transactions in the block will be confirmed earlier, the average latency of transactions will be decreased, and the latency cost of transactions will be reduced. This ensures more transactions join the blockchain system, meaning the remittance fee charged to transactions should be set higher.

We also note that for the same mining parameter θ, as the block capacity b increases, transactions in the Transaction Memory Pool are put into a block earlier and are confirmed earlier. Therefore, the average latency of transactions will decrease, and the latency cost of transactions will be reduced. This means more transactions join the blockchain system, and the remittance fee charged to transactions should be set higher.

5. **Conclusions.** In this paper, we presented a modeling approach to capture the blockchain system and investigated the Nash equilibrium of transactions in the blockchain system. Based on the working flow of a blockchain system, we established a type of non-exhaustive queueing model with a limited batch service and a possible zero-transaction service. By employing the methods of an embedded Markov chain and a generating function, we obtained the stationary probability distribution for the number of transactions in the system at the Markov points. Furthermore, we derived the average latency of transactions by analyzing the elapsed time for a mining cycle. And then, from the perspective of economics, we constructed an individual benefit function and a social benefit function to investigate the Nash equilibrium and the socially optimal strategies of transactions in the blockchain system. Finally, we presented a method for motivating transactions to accept the socially optimal arrival rate by charging an appropriate remittance fee to transactions.

In our future research, we will consider the influence for the number of transactions on the time duration of a block-verification process in blockchain system, and investigate a type of non-exhaustive queueing model with a limited batch service by employing Lindley-type equation.

Acknowledgments. This work was supported in part by National Natural Science Foundation (Nos. 61872311 and 61973261) and Natural Science Foundation of Hebei Province (No. F2017203141), China, and was supported in part by MEXT, Japan.

REFERENCES

[1] M. Aguiar and A. Lauve, Lagrange's theorem for Hopf monoids in species, *Canadian Journal of Mathematics*, **65** (2013), 241–265.
[2] A. Antonopoulos and O. Media, *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*, O'Reilly Media, 2014.
[3] R. Howell and E. Schrohe, Unpacking Rouché's Theorem, *PRIMUS: Problems, Resources, and Issues in Mathematics Undergraduate Studies*, **27** (2017), 801–813.
[4] S. Kasahara and J. Kawahara, Effect on transaction-confirmation process, *J. Ind. Manag. Optim.*, **15** (2019), 365–386.
[5] Y. Kawase and S. Kasahara, Transaction-confirmation time for bitcoin: A queueing analytical approach to blockchain mechanism, *12th International Conference of Queueing Theory and Network Applications, LNCS*, (2017), 75–88.
[6] Q. Li, J. Ma and Y. Chang, Blockchain queueing theory, (2018). Available from: `https://arxiv.org/abs/1808.01795`.
[7] R. Memon, J. Li and J. Ahmed, Simulation model for blockchain systems using queuing theory, *Electronics*, **8** (2019), 234–252.
[8] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, (2008). Available from: `https://www.coindesk.com/bitcoin-peer-to-peer-electronic-cash-system`.
[9] O. Novo, Blockchain meets IoT: An architecture for scalable access management in IoT, *IEEE Internet of Things Journal*, **5** (2018), 1184–1195.
[10] W. Qian, Q. Shao, Y. Zhu, C. Jin and A. Zhou, Research problems and methods in blockchain and trusted data management, *Journal of Software*, **29** (2018), 150–159.
[11] S. Ross, *Stochastic Processes*, Second edition, Wiley Series in Probability and Statistics: Probability and Statistics, John Wiley & Sons, Inc., New York, 1996.
[12] S. Saremi, S. Mirjalili and A. Lewis, Grasshopper optimisation algorithm: Theory and application, *Advances in Engineering Software*, **105** (2017), 30–47.
[13] M. Turkanović, M. Holbl, K. Kosic, M. Hericko and A. Kamisalic, Eductx: A blockchain-based higher education credit platform, *IEEE Access*, **6** (2018), 5112–5127.
[14] M. Vlasiou, A non-increasing Lindley-type equation, *Queueing Systems*, **56** (2007), 41–52.
[15] L. Wang, X. Shen, J. Li, J. Shao and Y. Yang, Cryptographic primitives in blockchains, *Journal of Network and Computer Applications*, **127** (2019), 43–58.
[16] R. Wolff and Y. Yao, Little's law when the average waiting time is infinite, *Queueing Systems*, **76** (2014), 267–281.

[17] H. Zhao, P. Bai, Y. Peng and R. Xu, Efficient key management scheme for health blockchain, *CAAI Transactions on Intelligence Technology*, **3** (2018), 114–118.

[18] W. Zhao, S. Jin and W. Yue, Analysis of the average confirmation time of transactions in a blockchain system, *14th International Conference of Queueing Theory and Network Applications, LNCS*, (2019), 379–388.

Received May 2019; revised October 2019.

E-mail address: zwj8569@163.com
E-mail address: jsf@ysu.edu.cn
E-mail address: yue@konan-u.ac.jp