



UDC 519.21

DOI: 10.22363/2658-4670-2023-31-4-332-344

EDN: ECPYRN

# On the algorithmization of construction of the transition intensity matrix in systems with a large number of same elements

Sergey I. Matyushenko, Ivan S. Zaryadov

*RUDN University,  
6 Miklukho-Maklaya St., Moscow, 117198, Russian Federation*

(received: October 14, 2023; revised: December 1, 2023; accepted: December 29, 2023)

**Abstract.** In this article, using the example of a multi-channel exponential queuing system with reordering of requests, we study the problem of computer construction of the state space and coefficient matrix of a system of equilibrium equations. As a result, general principles for solving problems of this type are formulated.

**Key words and phrases:** queuing system, dynamic construction of transition intensity matrix, systems with reordering of requests

## 1. Introduction

Quite often, when constructing mathematical models of complex information and computing systems, developers are faced with the problem of the presence of a large number of similar elements in the system under study. This is especially true for multi-threaded systems or systems with a large number of servicing devices [1–7]. If the functioning of the system has a complex multivariate scenario, then the task of constructing a state space of a random process modeling the system under study becomes impossible without developing a special software product. In this article, using the example of a multi-channel queuing system with reordering of requests [8, 9], an algorithm will be developed for the automated construction of the state space and transition intensity matrix.

## 2. System description

A multi-channel queuing system (QS) is considered with  $m$  service devices,  $2 \leq m < \infty$  and a common storage device of limited capacity  $r$ . The system receives a Poisson flow of requests of intensity  $\lambda$ . The service times on device  $j$  are independent of each other, and also do not depend on the duration of service on other devices and are distributed according to an exponential law with the parameter  $\mu_j$ ,  $j = 1, \dots, m$ . An application entering the system when there are  $m + r$  applications in it is lost.



We will further assume that the devices are arranged in non-decreasing order of service intensity:  $\mu_1 \geq \dots \geq \mu_m$  and that a request that has the ability to select a device selects the device with the lowest serial number. The selection of applications from the storage occurs in accordance with the FIFO discipline [10].

It is assumed that all applications upon entry into the system are assigned a serial number. Moreover, if at the moment of completion of servicing of a request with number  $n$  ( $n$ -request), servicing of at least one request with a number less than  $n$  continues, the  $n$ -request is placed in the reordering buffer (RB). Otherwise, request  $n$  leaves the system and all requests with numbers differing by one, starting from  $n + 1$  (if there are any in the RB), leave the RB behind it. This assumption allows us to model the mechanism for maintaining the order in which applications leave the system in accordance with the order in which they arrive. Systems of this kind are called systems with reordering of requests [8, 9].

### 3. Construction of a mathematical model

Let us assume that all applications in the system are numbered in accordance with the order in which they were received, starting with one. Then the stochastic behavior of the considered QS can be described by a homogeneous Markov process  $X(t)$ ,  $t \geq 0$ , over the state space

$$x^m = \bigcup_{k=0}^{m+r} x_k^m,$$

$$x_k^m = \{(k, i_1, \dots, i_m), \quad i_j = \overline{0, k}, \quad \sum_{j=1}^m u(i_j) = k,$$

in this case, if  $i_j i_s > 0$ , then  $i_j \neq i_s, \quad j, s = \overline{1, m}\}, \quad k = \overline{0, m-1},$

$$x_k^m = \{(k, i_1, \dots, i_m), \quad i_j = \overline{0, m}, \quad i_j \neq i_s, \quad j, s = \overline{1, m}\}, \quad k = \overline{m, m+r},$$

where  $u(x)$  is the Heaviside function.

Here for some time  $t$ :  $X(t) = (k, i_1, \dots, i_m)$ , if there are  $k$  requests in the system,  $k = 0, \dots, m + r$  and device  $j$  is free if  $i_j = 0$ . Otherwise,  $i_j$  is the serial number of the request served on device  $j$ ,  $j = 1, \dots, m$ .

In what follows, we will call the subset  $x_k^m$  the  $k$ -th group of states. It's easy to see that

$$|x_k^m| = \begin{cases} (m)_k, & k = \overline{0, m}, \\ m!, & k = \overline{m, m+r}, \end{cases}$$

where  $(m)_k$  is the number of placements from  $m$  to  $k$ .

Hence,

$$|x^m| = \sum_{k=0}^m (m)_k + rm!.$$

It is obvious that as  $m$  increases, the dimension of the state space increases rapidly. So when  $m \geq 5$  and  $k \geq 10$  it exceeds  $10^3$ . Therefore, to construct a matrix of transition intensities and solve a system of equilibrium equations, it is necessary to develop an algorithm for constructing a space  $x^m$ .

#### 4. Algorithm for constructing the state space

Let  $Y_s$  be a set of sequences of length  $s + 1$  of non-negative integers.

**Definition 1.** We will call the operator  $L_j$  the  $j$ -insertion operator defined on the set  $Y_s$  if for  $(i_0, \dots, i_s) \in Y_s$

$$L_j(i_0, i_1, \dots, i_s) = (i_0+1, i_1, \dots, i_{j-1}, \max\{i_1, \dots, i_s\}+1, i_j, \dots, i_s), \quad j = \overline{1, s+1}.$$

Next, let  $Y_{s,\nu}$  be the subset  $Y_s$  of power  $\nu$ , i.e.  $Y_{s,\nu} = \{y_s^1, \dots, y_s^\nu\}$ , where  $y_s^n = (i_0^n, \dots, i_s^n)$ ,  $n = \overline{1, \nu}$ .

**Definition 2.** We will call the operator  $L$  the insertion operator defined on the set of different finite subsets of the set  $Y_s$ , if for  $Y_{s,\nu} \in Y_s$

$$L(Y_{s,\nu}) = \{L_1 y_s^1, \dots, L_1 y_s^\nu, L_2 y_s^1, \dots, L_2 y_s^\nu, \dots, L_{s+1} y_s^1, \dots, L_{s+1} y_s^\nu\}.$$

**Definition 3.**  $k$ -th degree  $L^k$  operator  $L$  will be called an operator whose action consists of  $k$  successive applications of the operator  $L$ ,  $k = 1, 2, 3, \dots$ . By the zero degree of the operator  $L$  we mean the identity operator.

**Definition 4.** We will call  $L_j^{-1}$  a  $j$ -removal operator defined on the set  $Y_s$  if for  $(i_0, \dots, i_s) \in Y_s$ :

$$L_j^{-1}(i_0, \dots, i_{j-1}, i_j, i_{j+1}, \dots, i_s) = (i_0 - 1, i_1, \dots, i_{j-1}, i_{j+1}, \dots, i_s), \quad j = \overline{1, s}.$$

Let's define a subset  $\tilde{Y}_s$  as set  $Y_s$  such that for  $(i_0, i_1, \dots, i_s) \in Y_s$  among the numbers there is at least one that is not equal to zero and all non-zero numbers are distinct.

**Definition 5.** We will call the operator  $M$  a maximum selection operator defined on the set  $\tilde{Y}_s$  if for  $(i_0, i_1, \dots, i_s) \in \tilde{Y}_s$ :  $M(i_0, i_1, \dots, i_s) = l$ , where  $l$  is such that  $i_l = \max\{i_1, \dots, i_s\}$ .

And, finally, let  $\hat{Y}_s$  is subset of the set  $Y_s$ , such that  $(i_1, \dots, i_s) \in \hat{Y}_s$ , if among the numbers  $i_1, \dots, i_s$  there is at least one that is equal to zero.

**Definition 6.** We will call the operator  $Z$  the zero selection operator on the set  $\hat{Y}_s$  if for  $(i_0, i_1, \dots, i_s) \in \hat{Y}_s$ :  $Z(i_1, \dots, i_s) = n$ , where  $n$  is the number of the first zero element in the sequence  $i_1, \dots, i_s$ .

Let us now proceed to constructing the state space and prove the validity of the following lemma.

**Lemma 1.** For any fixed  $m, m \geq 2$ :

$$x_k^m = \begin{cases} L^k(0, 0^{m-k}), & k = \overline{0, m-1}, \\ L^{m-1}(k-m+1, 1), & k = \overline{m, m+r}, \text{ where } 0^s = (0, \dots, 0). \end{cases} \tag{1}$$

**Proof.** We will carry out the proof using the method of mathematical induction. Let  $m = 2$ . Then

$$x_k^2 = \begin{cases} (0, 0, 0), & k = 0; \\ L^1(0, 0), & k = 1; \\ L^1(k-1, 1), & k = \overline{2, r+2}. \end{cases}$$

Hence,  $= \{(0, 0, 0); (1, 1, 0); (1, 0, 1); (k, 1, 2); (k, 2, 1), k = 2, \dots, r+2\}$ . It is obvious that the resulting set is the state space for the process  $X(t)$  in the case of  $m = 2$  [4].

Let the statement of the lemma be true for  $m = l$ . Then for  $m = l + 1$  we get

$$\begin{aligned} x_k^{l+1} &= \begin{cases} L^k(0, 0^{l+1-k}), & k = \overline{0, l}; \\ L^l(k-l, 1), & k = \overline{l+1, l+r+l} \end{cases} = \\ &= \begin{cases} (0, 0^{l+1}), & k = 0; \\ L(L^{k-1}(0, 0^{l+1-k})), & k = \overline{1, l}; \\ L(L^{l-1}(k-l, 1)), & k = \overline{l+1, l+r+l}. \end{cases} \end{aligned}$$

Replace  $k$  with  $k + 1$  and get:

$$x_{k+1}^{l+1} = \begin{cases} (0, 0^{l+1}), & k = -1; \\ L(L^k(0, 0^{l-k})), & k = \overline{0, l-1}; \\ L(L^{l-1}(k-l+1, 1)), & k = \overline{l, l+r} \end{cases} = \begin{cases} (0, 0^{l+1}), & k = -1; \\ L(x_k^l), & k = \overline{0, l-1}; \\ L(x_k^l), & k = \overline{l, l+r}. \end{cases}$$

Next we note that if  $(k, i_1, \dots, i_l) \in x_k^l$ , then

$$\max\{i_1, \dots, i_l\} = \begin{cases} k, & k = \overline{0, l-1}; \\ l, & k = \overline{l, l+r}. \end{cases}$$

Hence,

$$x_{k+1}^{l+1} = \begin{cases} (0, 0^{l+1}), & k = -1; \\ \{(k + 1, k + 1, i_1^1, \dots, i_l^1); (k + 1, k + 1, \dots, i_l^2); \dots; \\ \quad (k + 1, i_1^\nu, \dots, k + 1)\}, & k = \overline{0, l - 1}, \quad \nu = |x_k^l|; \\ \{(k + 1, l + 1, i_1^1, \dots, i_l^1); (k + 1, l + 1, \dots, i_l^2); \dots; \\ \quad (k + 1, i_1^\nu, \dots, l + 1)\}, & k = \overline{l, l + r}, \quad \nu = |x_k^l|. \end{cases}$$

Given the definition of a group of states  $x_k^l$ , the last relation can be written as:

$$x_{k+1}^{l+1} = \begin{cases} \{(k + 1, i_1, \dots, i_l) : \\ \quad i_j = \overline{0, k + 1}, \quad \sum_{j=1}^{l+1} u(i_j) = k + 1, \text{ in this case,} \\ \quad \text{if } i_j i_s > 0, \text{ then } i_j \neq i_s, \quad j, s = \overline{1, l + 1}\}, & k = \overline{-1, l - 1}; \\ \{(k + 1, i_1, \dots, i_{l+1}) : \\ \quad i_j = \overline{1, l + 1}, \quad i_j \neq i_s, \quad j, s = \overline{1, l + 1}\}, & k = \overline{l, l + r}. \end{cases}$$

And finally, having made the reverse replacement of  $k$  by  $k - 1$ , we come to the definition of the  $k$ -th group of states for the case  $m = l + 1$ . Thus, the lemma is proven. □

It obviously follows from lemma 1 that the proposed method of constructing  $x^m$  is recurrent in  $m$ . In addition, in the state space a certain order of the elements of this space is specified. For clarity, let us consider the diagram for constructing the state space in the case when  $m = 4$  and  $r = 1$  (figure 1).

Analysis of the diagram helps to notice that for any fixed  $m$ , the  $k$ -th group of states is divided into  $m$  subgroups of the same dimension. A sign that a state belongs to the  $n$ -th subgroup of the  $k$ -th group is that the request with the highest number is served on device  $n$ ,  $n = 1, \dots, m$ . Let us denote  $x_{k,n}^m$  by the  $n$ -th subgroup of the  $k$ -th group. It is easy to calculate that

$$|x_{k,n}^m| = (m - 1)_{\min\{k, m-1\}-1}, \quad n = \overline{1, m}, \quad k = \overline{1, m + r}. \tag{2}$$

The recurrent principle of constructing the state space and dividing groups into subgroups makes it possible to determine the serial number of the state in  $x^m$ .

**Lemma 2.** *The ordinal number of the state  $(k, i_1, \dots, i_m)$  in the state space  $x^m$  is determined by the expression*

$$n = \sum_{j=0}^{\min\{k-1, m-1\}} (m)_j + u(k - m)m! + \sum_{j=1}^{\min\{k, m-1\}} (s_j - 1)(m - j)_{\min\{k, m\}-j} + 1, \tag{3}$$

where  $s_1 = M(k, i_1, \dots, i_m)$ ,  $s_j = M(L_{s_{j-1}}^{-1} \dots L_{s_2}^{-1} L_{s_1}^{-1}(k, i_1, \dots, i_m))$ ,  $j = \overline{2, \min\{m - 1, k\}}$ .

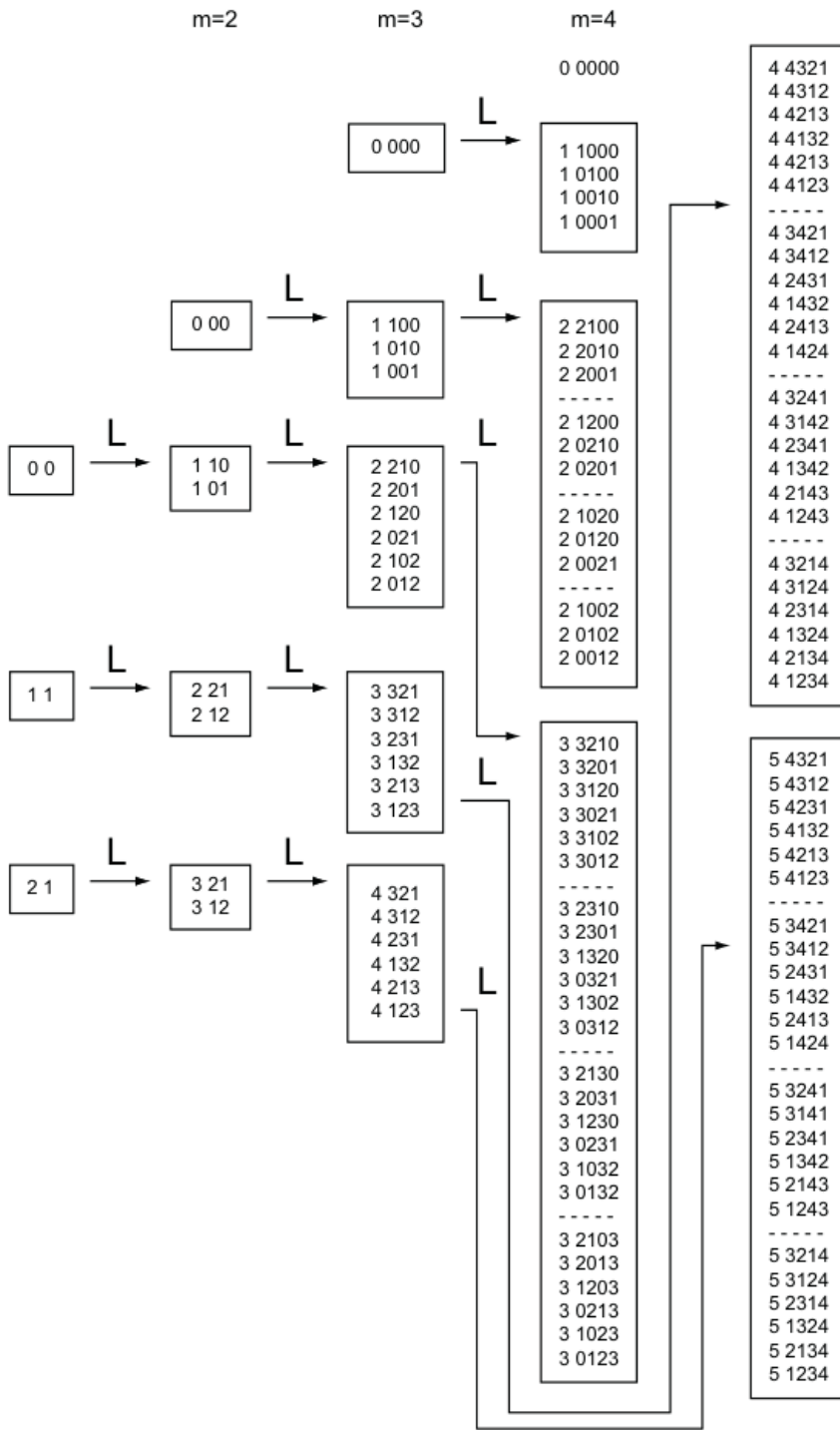


Figure 1. Diagram of the process of constructing a state space for a system with 4 devices

**Proof.** Note that the first two terms in our formula determines the total number of states in groups  $x_0^m, \dots, x_{k-1}^m$ . Therefore, it remains to show that  $\sum_{j=1}^{\min\{k,m-1\}} (s_j - 1)(m - j)_{\min\{k,m\}-j}$  determines the number of states of the  $k$ -th group preceding a given state. To do this, let us clarify the meaning of each term of this sum.

Notice, that  $s_1 = M(k, i_1, \dots, i_m)$  is the number in the subgroup  $x_k^m$  that belongs to state  $(k, i_1, \dots, i_m)$ , and the size of each of the subgroups is equal  $(m - 1)_{\min\{k,m\}-1}$ . Therefore, the expression  $(s_1 - 1)(m - 1)_{\min\{k,m\}-1}$  determines the total number of states in the subgroups  $x_{k,1}^m, \dots, x_{k,s_1-1}^m$ , preceding the subgroup  $x_{k,s_1}^m$ . Moreover, if  $k = 1$  or  $m = 2$ , then each of the subgroups contains one element and, therefore, the calculation process will be completed. Otherwise, you need to define the state number in the subgroup  $x_{k,s_1}^m$ .

We note that in the process of recurrent construction of the state space, the subgroup  $x_{k,s_1}^m$  is obtained as a result of the action of the operator  $L_{s_1}$  on the group  $x_{k-1}^{m-1}$ . Therefore, the ordinal number of state  $(k, i_1, \dots, i_m)$  in the subgroup  $x_{k,s_1}^m$  is equal to the ordinal number of state  $L_{s_1}^{-1}(k, i_1, \dots, i_m)$  in the group  $x_{k-1}^{m-1}$ . To determine it, we find  $s_2 = M(L_{s_1}^{-1}(k, i_1, \dots, i_m))$  — the number of the subgroup in the group  $x_{k-1}^{m-1}$  to which state  $L_{s_1}^{-1}(k, i_1, \dots, i_m)$  belongs. The size of each of these subgroups is equal  $(m - 2)_{\min\{k,m\}-2}$ . So expression  $(s_2 - 1)(m - 2)_{\min\{k,m\}-2}$  determines total number of states in the subgroups  $x_{k-1,1}^{m-1}, \dots, x_{k-1,s_2-1}^{m-1}$ , preceding the subgroup  $x_{k-1,s_2}^{m-1}$ . Moreover, if  $k = 2$  or  $m = 3$ , then each of the subgroups contains one element and the calculation process will be completed. Otherwise, it is necessary to continue similar reasoning, which will ultimately lead us to the desired result.

Thus the lemma is proven. □

In the future, we will need not only a formula for calculating the serial number of a state, but also an algorithm for the reverse action — restoring a state by its serial number. This task in our case is divided into two stages: determining the number of the group to which a given state belongs and calculating the serial number of the state in the group.

The first stage is simple. Let  $n$  be the serial number of the state, and  $N = |x^m|$ . Next, it is necessary to arrange a partition of the segment  $[0, N]$  into  $m + r + 1$  interval  $(n_k, n_{k+1}]$ ,  $k = 0, \dots, m + r$  in such a way that the condition  $n \in (n_k, n_{k+1}]$  means that the state with number  $n$  belongs to the  $k$ -th group. Obviously, as the boundaries of the indicated intervals it is necessary to take the numbers:

$$n_s = \begin{cases} 0, & s = 0; \\ n_{s-1} + (m)_{s-1}, & s = \overline{1, m}; \\ n_{s+1} + m!, & s = \overline{m + 1, m + r + 1}. \end{cases} \tag{4}$$

Further, carrying out arguments similar to those that took place in the proof of lemma 2, we arrive at the following result.

**Lemma 3.** *Let  $n$  be the serial number of the state, and  $k$  be the number of the group to which this state belongs, and let the sequence of numbers  $s_1, \dots, s_{\min\{k, m-1\}}$  be defined by the following recurrence relations*

$$\begin{aligned}
 l_j &= \begin{cases} n - n_k, & j = 1; \\ l_{j-1} - (s_{j-1} - 1)t_{j-1}, & j = \overline{2, \min\{m-1, k\}}; \end{cases} \\
 t_j &= (m - j)_{\min\{k, m\} - j}, \quad s_j = \left\lceil \frac{l_j}{t_j} \right\rceil, \quad j = \overline{1, \min\{m-1, k\}}.
 \end{aligned}
 \tag{5}$$

Then the state of the system is determined by the expression:

$$(k, i_1, \dots, i_m) = \begin{cases} L_{s_1} \dots L_{s_k}(0, 0^{m-k}), & k = \overline{0, m-1}; \\ L_{s_1} \dots L_{s_{m-1}}(k - m + 1, 1), & k = \overline{m, m+r}. \end{cases}
 \tag{6}$$

To illustrate the operation of our algorithm, we give an example of restoring a state by its serial number.

Let  $m = 4, r = 2, n = 61$ .

Then  $N = \sum_{j=0}^6 (4)_j = 1 + 4 + 12 + 24 + 24 + 24 = 89$  and the segment  $(0, 89]$  is divided into intervals:

$$\begin{aligned}
 (n_0, n_1] &\equiv (0, 1]; & (n_1, n_2] &\equiv (1, 5]; & (n_2, n_3] &\equiv (5, 17]; \\
 (n_3, n_4] &\equiv (17, 41]; & (n_4, n_5] &\equiv (41, 65]; & (n_5, n_6] &\equiv (65, 89].
 \end{aligned}$$

The number 61 belongs to the interval  $(n_4, n_5]$ . Therefore, group number  $k = 4$ . Next, we calculate the sequence of numbers.

$$\begin{aligned}
 l_1 &= 61 - 41 = 20, & t_1 &= (4 - 1)_{4-1} = 6, & s_1 &= \lceil 20/6 \rceil = 4; \\
 l_2 &= 20 - (4 - 1) \cdot 6 = 2, & t_2 &= (4 - 2)_{4-2} = 2, & s_2 &= \lceil 2/2 \rceil = 1; \\
 l_3 &= 2 - (1 - 1) \cdot 2 = 2, & t_3 &= (4 - 3)_{4-3} = 1, & s_3 &= \lceil 2/1 \rceil = 2.
 \end{aligned}$$

And finally, the desired state is determined by the expression:

$$(4, i_1, i_2, i_3, i_4) = L_4 L_1 L_2(1, 1) = L_4 L_1(2, 1, 2) = L_4(3, 3, 1, 2) = (4, 3, 1, 2, 4).$$

Using lemma 2, we check the result:

$$\begin{aligned}
 s_1 &= M(4, 3, 1, 2, 4) = 4; \\
 s_2 &= M(L_4^{-1}(4, 3, 1, 2, 4)) = M(3, 3, 1, 2) = 1; \\
 s_3 &= M(L_1^{-1}(3, 3, 1, 2)) = M(2, 1, 2) = 2; \\
 n &= \sum_{j=0}^3 (4)_j + \sum_{j=1}^3 (s_j - 1)(4 - j)_{4-j} + 1 = 41 + 20 = 61.
 \end{aligned}$$



## 5. Algorithm for constructing the matrix of transition intensities

Let us develop an algorithm for constructing the matrix of MP transition intensities  $X(t)$ . Note that MP transitions are possible only for states from neighboring groups: transition from the group  $x_{k-1}^m$  to a group  $x_k^m$  due to the receipt of an application, and from group  $x_{k+1}^m$  to group  $x_k^m$  — due to service on one of the devices. Consequently, the transition intensity matrix  $A$  will have a 3-diagonal form.

Let us introduce notation for non-zero blocks  $A$ . We denote over-diagonal blocks by  $\Lambda_{k-1,k}$ , sub-diagonal — through  $M_{k,k-1}$ , and diagonal — through  $N_{kk}$ .

It is not difficult to notice that in each column of the block  $\Lambda_{k-1,k}$  there can be only one non-zero element equal to  $\lambda$ . In each line of the block  $M_{k,k-1}$  corresponding to state  $(k, i_1, \dots, i_m)$  there will be no more than  $m$  elements for  $j$  such that  $i_j \neq 0$ . Finally, the blocks  $N_{k,k}$  are diagonal matrices with the diagonal element corresponding to state  $(k, i_1, \dots, i_m)$  equal to  $\sum_{j=1}^m u(i_j)\mu_j - u(m+r-k)\lambda_k$ .

Thus, to construct matrix  $A$  it is necessary:

- develop an algorithm  $E_0$  that determines for any state of the system the transition conditions and the state of the system from which one can get to the given one due to the receipt of an application;
- develop an algorithm  $E_j$ , which defines the transition conditions and the state of the system to which it is possible to move from this state due to servicing on device  $j$ .

The implementation of these algorithms will make it possible to determine non-zero elements of blocks  $\Lambda_{k-1,k}$  and  $M_{k,k-1}$ .

Before moving on to a step-by-step description of the algorithm  $E_0$ , we will give some explanation of one of the conditions that will be checked in this algorithm. We are talking about when it is impossible to transition to state  $(k, i_1, \dots, i_m)$ , from any other due to the arrival of a request, except for the trivial case  $k = 0$ . In the description of the system it was said that an application that has the ability to select a device selects the device with the lowest serial number. Consequently, due to the receipt of a request, it is impossible to get into a state for which the number of the device busy servicing the request with the maximum sequence number is greater than the number of the first of the free devices. Formally, for any state  $(k, i_1, \dots, i_m)$ , this condition can be written as follows

$$M(k, i_1, \dots, i_m) > Z(k, i_1, \dots, i_m),$$

where  $M$  and  $Z$  are the maximum and zero selection operators, respectively. Moreover, this condition makes sense to check when  $k < m$ . Otherwise the check is trivial. The meaning of the remaining steps of the algorithm is quite obvious, so we will not give a detailed explanation.

So, **algorithm  $E_0$** :

**Start:** enter state  $(k, i_1, \dots, i_m)$ .

**Step 1.** Check the condition  $k > m$ . If the condition is met, then go to step 5.

**Step 2.** Check the condition  $k = m$ . If the condition is met, then go to step 4.

**Step 3.** Check the condition  $M(k, i_1, \dots, i_m) > Z(k, i_1, \dots, i_m)$ . If the condition is met, then the end of the algorithm.

**Step 4.**  $l := Z(k, i_1, \dots, i_m)$ ,  $i_l := 0$ .

**Step 5.**  $k := k - 1$ .

**Step 6.** Output values  $k, i_1, \dots, i_m$ .

**End algorithm.**

Now let's move on to the algorithm  $E_j$ . Note that from any state  $(k, i_1, \dots, i_m)$  you can get to another due to servicing on device  $j$  if this device was busy with servicing, i.e. if  $i_j \neq 0$ . After the end of service on device  $j$ , the number of requests decreases by one. In addition, the serial numbers of those applications that entered the system later than the serviced application are reduced. Further, if at the time of the end of service there were requests in the queue awaiting service, then the first of them will arrive at device  $j$  and will be assigned a serial number  $m$ . Otherwise, device  $j$  will become free.

Considering the above, the **algorithm**  $E_j$  can be written as:

**Start:** Enter state  $(k, i_1, \dots, i_m)$ .

**Step 1.** Check condition  $i_j = 0$ . If the condition is met, then the end of the algorithm.

**Step 2.**  $k := k - 1$ ,  $l := 0$ .

**Step 3.**  $l := l + 1$ .

**Step 4.** Check the condition  $((i_l > 0) \text{ and } (i_l < i_j))$ . If the condition is met, then go to step 6.

**Step 5.**  $i_l := i_l - 1$ .

**Step 6.** Check the condition  $l \neq m$ . If the condition is met, then go to step 3.

**Step 7.**  $i_j := 0$ .

**Step 8.** Check the condition  $k < m$ . If the condition is met, then go to step 10.

**Step 9.**  $i_j := m$ .

**Step 10.** Print values  $k, i_1, \dots, i_m$ .

**End algorithm.**

And finally, we will describe the algorithm  $E$  for constructing matrix  $A$ . In fact, we need to perform the following actions for all serial numbers of states of the system  $n$ :

**Step 1.** Using lemma 3, from the ordinal number  $n$ , restore the state of the system  $(k, i_1, \dots, i_m)$ .

**Step 2.** Using an algorithm  $E_0$  (algorithms  $E_j$ ), determine the transition conditions, and if they are fulfilled, the state of the system from which (to which) one can get to the given one (from the given one).

**Step 3.** Using lemma 2, determine the serial numbers of states  $n_j$ ,  $j = 0, 1, \dots, m$ , defined in the previous step, and perform the following assignment operations:

$$a_{n_0 n} := \lambda, \quad (7)$$

$$a_{n n_j} := \mu_j, \quad (8)$$

$$a_{nn} := -u(m + rk)\lambda - \sum_{j=1}^m u(i_j)\mu_j. \quad (9)$$

Thus, the main result of our study has been obtained. Let us formulate it in the form of a theorem.

**Theorem 1.** *The numbering of non-zero elements of the matrix  $A$  of the intensities of transitions of the MP  $X(t)$ , which describes the functioning of the system under consideration, is determined in accordance with the algorithm  $E$ , and their values are calculated using formulas (7)–(9).*

## 6. Conclusion

Despite the fact that in this work we were constructing a matrix of transition intensities for a very specific system, we can note a number of patterns and formulate recommendations for solving problems of this type when considering other systems with a large number of similar elements. Let's list the main stages:

1. Describe all possible states of the system in the form of sequences of corresponding numerical parameters.
2. Identify the main patterns of formation of the state space with an increase in the number of similar elements in the system and describe them using special operators.
3. To develop an algorithm for constructing a state space recurrently by the number of elements of the same type.
4. Define and formalize with the help of logical operators all possible transitions between different states of the system.
5. Develop an algorithm that allows you to determine the ordinal number of a state in the state space.
6. Develop an inverse algorithm that allows, using the serial number of a state, to restore the state itself in the form of a sequence of parameters.
7. Write down formulas for determining non-zero elements of the matrix of intensities of transitions between states of the system depending on the serial numbers of these states.
8. Enumerate all states of the system in accordance with their serial numbers and, restoring them using the inverse algorithm, calculate all non-zero elements of the matrix using the formulas from the previous paragraph.

## References

- [1] V. L. Chugreev, "Development of a multiplicative model of sequentially connected information elements," *Young Scientist*, no. 3, pp. 147–149, 2013.
- [2] G. P. Basharin, S. N. Klapouschak, and N. V. Mitkina, "Mathematical model of adaptive high-speed system with elastic traffic," *Bulletin of the RUDN. Mathematics series. Computer science. Physics*, no. 3, pp. 31–39, 2008, in Russian.

- [3] A. Rumyantsev and E. Morozov, "Stability criterion of a multiserver model with simultaneous service," *Operation Reseach*, no. 5, pp. 31–39, 2017. DOI: 10.1007/s10479-015-1917-2.
- [4] S. A. Grishunina, "Multiserver queueing system with constant service time and simultaneous service of a customer by random number of servers," *Theory of Probability and its Applications*, vol. 64, no. 3, pp. 456–460, 2019. DOI: 10.1137/50040585X97T98960X.
- [5] B. Sun, M. H. Lee, S. A. Dudin, and A. N. Dudin, "Analysis of multi-server queueing system with opportunistic occupation and reservation of servers," *Mathematical Problems in Engineering*, no. 5, pp. 1–13, 2014. DOI: 10.1155/2014/178108.
- [6] U. Ayestab, P. Jackod, and V. Novak, "Scheduling of multi-class multi-server queueing systems with abandonments," *Journal of Scheduling*, vol. 20, pp. 129–145, 2015. DOI: 10.1007/s10951-015-0456-7.
- [7] M. Harchol-Balter, T. Osogami, A. Scheller-wolf, and A. Wierman, "Multi-server queueing systems with multiple priority classes," *Queueing Systems*, vol. 51, pp. 331–360, Dec. 2005. DOI: 10.1007/s11134-005-2898-7.
- [8] S. Matyushenko and A. Ermolaeva, "On stationary characteristics of a multi server exponential queueing system with reordering of requests," in *13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, ICUMT 2021, Brno, Czech Republic, 2021, pp. 98–103.
- [9] S. I. Matyushenko and A. V. Pechinkin, "Service system with reordering of applications," in *International Conference Distributed Computer and Communication Networks (ECN 2011)*, ECN 2011, Moscow, Russia, 2011.
- [10] P. P. Bocharov and A. V. Pechenkin, *Queueing theory [Teoriya massovogo obsluzhivaniya]*. Moscow: RUDN, 1995, 529 pp., in Russian.

**For citation:**

S. I. Matyushenko, I. S. Zaryadov, On the algorithmization of construction of the transition intensity matrix in systems with a large number of same elements, *Discrete and Continuous Models and Applied Computational Science* 31 (4) (2023) 332–344. DOI: 10.22363/2658-4670-2023-31-4-332-344.

**Information about the authors:**

**Matyushenko, Sergey I.** — Candidate of Physical and Mathematical Sciences, Assistant professor of Department of Probability Theory and Cyber Security of Peoples' Friendship University of Russia named after Patrice Lumumba (RUDN University) (e-mail: matyushenko-si@rudn.ru, ORCID: <https://orcid.org/0000-0001-8247-8988>)

**Zaryadov, Ivan S.** — Candidate of Physical and Mathematical Sciences, Assistant professor of Department of Probability Theory and Cyber Security of Peoples' Friendship University of Russia named after Patrice Lumumba (RUDN University) (e-mail: zaryadov-is@rudn.ru, ORCID: <https://orcid.org/0000-0002-7909-6396>)

УДК 519.21

DOI: 10.22363/2658-4670-2023-31-4-332-344

EDN: ECPYRN

## Об алгоритмизации построения матрицы интенсивностей переходов в системах с большим числом однотипных элементов

С. И. Матюшенко, И. С. Зарядов

*Российский университет дружбы народов,  
ул. Миклухо-Маклая, д. 6, Москва, 117198, Российская Федерация*

**Аннотация.** В данной статье на примере многоканальной экспоненциальной системы массового обслуживания с переупорядочиванием заявок изучается задача компьютерного построения пространства состояний и матрицы коэффициентов системы уравнений равновесия. В результате сформулированы общие принципы решения задач такого типа.

**Ключевые слова:** система массового обслуживания, динамическое построение матрицы интенсивностей переходов, система с переупорядочиванием заявок