mathematics

MDPI

Article

Analysis of Multi-Server Queueing System with Flexible Priorities

Konstantin Samouylov [1], Olga Dudina [2] and Alexander Dudin [2,*]

[1] Applied Mathematics and Communications Technology Institute, Peoples' Friendship University of Russia (RUDN University), 6 Miklukho-Maklaya St., 117198 Moscow, Russia
[2] Department of Applied Mathematics and Computer Science, Belarusian State University, 4, Nezavisimosti Ave., 220030 Minsk, Belarus
* Correspondence: dudin@bsu.by

Abstract: In this paper, a multi-server queueing system providing service to two correlated flows of requests was considered. Non-preemptive priority was granted to one flow via the preliminary delay of requests in the intermediate buffers with different rates of extracting from the buffers. Customers' impatience during waiting in the intermediate and main buffers was taken into account. The possibility of the use of the results of the mathematical analysis for managerial goals is numerically illustrated.

Keywords: priority queuing system; *MMAP*; impatience; capacity planning

MSC: 60K25; 60K30; 68M20; 90B22

1. Introduction

Queueing theory is a powerful tool for solving the problems of optimal sharing and scheduling limited resources in many real-world systems in the fields of telecommunications, transportation, logistics, emergency services, health-care, computer systems and networks, manufacturing, etc.; for recent references, see, e.g., [1–9]. While the main amount of the existing queueing literature is devoted to the systems with homogeneous requests, the efforts of many researches have been focused also on the queueing systems with heterogeneous requests having, in general, different requirements for the service time and different economic value. An important class of such queueing systems assumes the support of a certain system of priorities provided to different types of requests aiming to create more comfortable conditions for requests belonging to the higher-priority classes. Examples of such classes are the urgent (related to safety for life or the security of objects) and non-urgent information in communication networks; handover and new calls in mobile communication networks; primary and cognitive users in cognitive radio systems; injured patients with a danger to their lives or without this in health emergency services; emergency and public or private transport on the roads in the city; preferred or ordinary clients of banks and other systems, etc.

The classical books on priority queues are [10–13]. As recent papers dealing with priority queues, the papers [14–22] can be mentioned.

In priority queueing systems, usually, requests of different types are stored in different (physically or virtually) buffers. Customers of low priority can be picked up for service only when the buffers designed for higher-priority requests are empty. There is a variety of different priority schemes suitable for modelling and optimising various real-life systems, including non-preemptive (not allowing the interruption of ongoing service), preemptive (allowing the interruption of service), and alternating priorities. Due to the finiteness of the shared resource and the use of work-conserving disciplines, the better are the conditions guaranteed to the high-priority requests, the worse are the conditions provided to the low-priority requests. Traditional, statical, priority schemes suggest that the priority is

check for updates

Citation: Samouylov, K.; Dudina, O.; Dudin, A. Analysis of Multi-Server Queueing System with Flexible Priorities. *Mathematics* **2023**, *11*, 1040. https://doi.org/10.3390/math11041040

Academic Editor: Manuel Alberto M. Ferreira

Received: 26 January 2023
Revised: 16 February 2023
Accepted: 17 February 2023
Published: 18 February 2023

assigned in advance and does not depend on the queue lengths in the system. Thus, it is possible that, sometimes, there is a very long queue of low-priority requests while the queue of high-priority requests is short. This may not be fair with respect to low-priority requests. Therefore, different strategies of dynamically providing the priorities have been offered in the literature since the paper [23]; see the survey [24]. Such strategies suggest that, at any moment of decision-making about the type of the request to be taken for service, this type is defined via some control policy depending on the relation of the queue lengths of different types of requests. A popular class of such policies is monotone policies using thresholds. Other possibilities to make the statical priority more favourable with respect to the low-priority requests are the use of randomisation in the choice (providing, with a certain probability, the chance to a low-priority request to enter the service even in the presence of high-priority requests), the mandatory service of a low-priority request after service, in turn a certain number of high-priority requests, provisioning the weighted service rates, etc.

There are also many works considering the accumulation of priority during a request stay in the queue. For a short review of the corresponding research, see, e.g., the papers [25,26]. In [25], the model with a heterogeneous-batch-correlated arrival process of two types of requests and the phase-type distribution of service time (see, e.g., [27] for the definition and properties of such a distribution) was analysed. A non-priority request becomes the priority request after its waiting time exceeds some random time having the phase-type distribution. In [26], the model with the heterogeneous correlated arrival process of an arbitrary finite number of types of requests, a finite common buffer space, the phase-type distribution of the service time, and exponential distributions of times until priority upgrading was analysed. In both of these papers, the arrival flow was assumed to be defined by the Marked Markov Arrival Process ($MMAP$) (see, e.g., [28,29]), which is the generalisation of the well-known Markov Arrival Process (MAP) to the case of heterogeneous requests. In turn, the MAP is the significant generalisation of a stationary Poisson arrival process. In contrast to the stationary Poisson arrival process, the MAP is suitable for modelling, in particular, the flows in the modern communication networks and contact centres that exhibit the correlation and high variability of inter-arrival times. It is already well known that the ignorance of the correlation and high variability of inter-arrival times can lead to huge errors in evaluating the performance and the design of real-world systems. For the literature about the queues with the MAP, its properties, partial cases, and possible applications see, e.g., [6,7,30–38]. The literature on the priority queues and $MMAP$ arrival process is still not very extensive. Among the recent papers mentioned above, such an arrival process was assumed in [14,18,20,22].

In the paper [39], a new flexible scheme for non-preemptive priority provision was offered. The idea of that scheme is not to define the rule for picking up requests of different priorities from the buffer, but to regulate the rate of admission of these requests to the buffer. This is achieved via managing the auxiliary intermediate buffers for preliminarily storing the arriving requests. The capacities of two intermediate buffers are different, as well as the rates of the transition of requests from these buffers into the main buffer, from which all requests are picked up for service according to First In–First Out (FIFO) principle. Via the proper choice of these rates and capacities, it is possible to provide any degree of priority for requests of both types. Usually considered in the literature, non-preemptive priorities are obtained as a very particular case of this priority scheme.

In this paper, we extended the results of [39] in two directions. The first direction is the consideration of a multi-server system instead of a single-server system, as analysed in [39]. Multi-server queueing systems more adequately describe many real-world systems where the shared restricted resource is split into independent units providing service to the requests (operators in call centres, cashiers in stores, logical information transmission channels obtained from a single physical channel via the use of various multiplexing methods, etc.) and are a more difficult subject for investigation. The second direction is avoiding the loss of requests in the case when the intermediate buffers are overloaded.

Newly arriving requests to any intermediate buffer seeing that the buffer is full are not lost, as was assumed in [39], but push the first request from this buffer into the main buffer and occupy the vacant place in the intermediate buffer. This feature allows not only modelling the systems where the loss of requests due to the buffer overflow is not possible, but it allows dynamically giving additional priority to the requests from the currently long queue. As in [39], we took into account the possible impatience of requests in the intermediate and main buffers because it is well known (see, e.g., [40]) that requests in many systems exhibit impatience due to various reasons.

The structure of the rest of the paper is as follows. The mathematical model is described in Section 2. The multidimensional stochastic process describing the behaviour of the considered model is introduced and analysed in Section 3. In Section 4, the formulas for the computation of the key performance measures of the system are presented. In Section 5, the results of the numerical experiment are given. Section 6 concludes the paper.

2. Mathematical Model

We analysed a queueing system having N independent identical servers and a buffer of infinite capacity. Each server of the system can provide service to two types of requests at rate μ, $\mu > 0$, independent of the type of request.

The arrivals of requests occur according to an $MMAP$. The $MMAP$ is determined by the irreducible continuous-time Markov chain ($CTMC$) ν_t, $t \geq 0$, with the finite state space $\{1, 2, \ldots, W\}$. The transition rates of this $CTMC$ are defined by the generator, denoted as $D(1)$. The matrix $D(1)$ is represented in the additive form as

$$D(1) = D_0 + D_1 + D_2$$

where the sub-generator D_0 defines the transition rates of the $CTMC$ ν_t, which do not cause requests' arrival. The non-negative matrix D_k defines the transition rates of the $CTMC$ ν_t, which are accompanied with the Type-k request arrival, $k = 1, 2$.

Let $\boldsymbol{\theta}$ be the invariant probability row vector of the $CTMC$ ν_t. This vector is computed as the unique solution to the system of linear algebraic equations $\boldsymbol{\theta} D(1) = \mathbf{0}$, $\boldsymbol{\theta}\mathbf{e} = 1$. Here and further, $\mathbf{e}$ denotes a column vector of 1s and $\mathbf{0}$ denotes a row vector of 0s with the appropriate dimension. The average arrival rate λ_k of Type-k requests is computed by the formula $\lambda_k = \boldsymbol{\theta} D_k \mathbf{e}$, $k = 1, 2$. The total arrival rate of requests to the system is defined as $\lambda = \lambda_1 + \lambda_2$. Generally speaking, the lengths of the intervals between requests' arrivals are correlated. The formulas for the computation of the coefficients of variation and correlation can be found, e.g., in [36]. The methods for the estimation of the parameters of the $MMAP$ describing the flow of requests in some real-world system based on the finite set of the observed request arrival moments (timestamps) were presented, e.g., in [41].

We assumed that Type-1 requests have a priority over Type-2 requests provided via the application of a request admission procedure, described as follows. If the request of any type arrives at the system when at least one server is idle, this request immediately starts service on an arbitrary idle server and, then, after being exponentially distributed with rate μ time, departs from the system. If an arbitrary Type-k request sees that all servers are busy, it is stored in the kth intermediate buffer, $k = 1, 2$. The capacities of the first and second intermediate buffers are equal to K and R, respectively. If the corresponding buffer is full, this request is placed in the buffer while the first request staying in this buffer is immediately pushed out of the buffer and transits to the main buffer of an infinite capacity. Each request placed in the kth buffer should reside there during exponential time with the rate γ_k, $\gamma_k \geq 0$, $k = 1, 2$. After this time expires, the request immediately transits to the main buffer. After storing in this buffer, the requests of both types become indistinguishable and are picked up from this buffer for service according to the FIFO principle. If, at some service completion moment, the main buffer is empty, the released server picks up for service the first request from the first buffer, if any. If the first buffer is empty, the offer to start service receives the first request from the second buffer. If all buffers are empty,

the server waits until any request arrives at the system, and it will have a chance to start service for this request.

As was proclaimed above, the described admission procedure is flexible in the sense of the degree of the privilege provided to Type-1 requests. The privilege is given via: (i) the order of polling the intermediate buffers when some server becomes idle (the request from the first buffer is invited for service first); (ii) the choice of the rates γ_k of the transition of the requests from the intermediate buffers to the main buffer (rate γ_1 can be arbitrarily larger than γ_2); (iii) the proper choice of the capacities of the intermediate buffers. In contrast to [39], where the small capacity of the buffer might cause the loss of an arriving request and the capacity of the buffer for low-priority requests could be chosen as small, to drop part of these requests, in the model considered in this paper, a small buffer for some type of requests helps to obtain a quicker transition to the main buffer due to the push out mechanism.

Customers staying in the intermediate buffers are impatient. Customers staying in the kth buffer depart from the buffer independently of each other (are lost) instead of transitioning to the main buffer after residing in the buffer while being exponentially distributed with the rate α_k time, $\alpha_k \geq 0$, $k = 1, 2$. Therefore, the large capacity of a buffer and a large impatience rate may stimulate the frequent loss of low-priority requests. Customers staying in the main buffer also can be impatient. The patience time was assumed to have an exponential distribution with the rate φ, $\varphi \geq 0$. After this time expires, the request departs from the system without service (is lost).

The operation of the system is schematically illustrated in Figure 1.

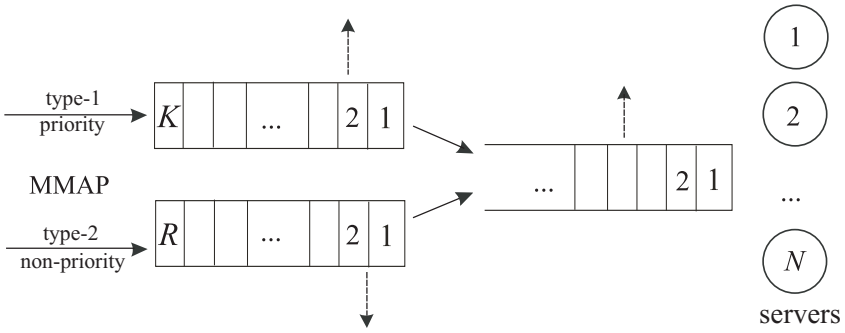

Figure 1. Structure of the system.

Our goals were to construct the Markov process describing the behaviour of the system, implement its steady-state analysis, and numerically highlight some dependencies of the system performance measures on the parameters of the model.

3. Random Process Defining the Behaviour of the System

3.1. Selection of the Random Process

Let:

- i_t, $i_t \geq 0$, be the total number of requests in service and in the main buffer;
- k_t, $k_t = \overline{0, K}$, be the number of requests in the first intermediate buffer;
- r_t, $r_t = \overline{0, R}$, be the number of requests in the second intermediate buffer;
- ν_t, $\nu_t = \overline{1, W}$, be the state of the underlying process of the $MMAP$;

at moment t, $t \geq 0$. Here, and further, notation like $\nu = \overline{1, W}$ means that the parameter ν admits values from the set $\{1, 2, \ldots, W\}$.

The four-dimensional $CTMC$ $\xi_t = \{i_t, k_t, r_t, \nu_t\}$, $t \geq 0$, is regular and irreducible. Its infinite state space is defined as

$$\left(\{i, 0, 0, \nu\},\ i = \overline{0, N-1}\right) \bigcup \left(\{i, k, r, \nu\},\ i \geq N\right),\ k = \overline{0, K},\ r = \overline{0, R},\ \nu = \overline{1, W}.$$

3.2. Generator of the Random Process

To write down the generator of the $CTMC\ \xi_t$, we need the following denotations:

$\mathrm{diag}\{a_1, \ldots, a_L\}$ is the diagonal matrix with the diagonal entries given by the numbers $\{a_1, \ldots, a_L\}$;

square matrices C_l, $\hat{I}_l$, $\tilde{I}_l$, E_l^-, and E_l^+ of size l, where $l = K+1$ or $l = R+1$, are given by:

$C_l = \mathrm{diag}\{0, 1, \ldots, l-1\}$;

$\hat{I}_l = \mathrm{diag}\{1, 0, \ldots, 0\}$;

$\tilde{I}_l = \mathrm{diag}\{0, \ldots, 0, 1\}$;

matrices E_l^- and E_l^+ have all zero entries, except the values $(E_l^-)_{k,k-1}$, $k = \overline{1, l-1}$, and $(E_l^+)_{k,k+1}$, $k = \overline{0, l-2}$, correspondingly, which are equal to 1;

$\hat{\boldsymbol{e}}_l$ is a row vector of size l : $\hat{\boldsymbol{e}}_l = (1, 0, \ldots, 0)$, $l = K+1,\ R+1$;

$\hat{\boldsymbol{e}}_l^T$ is the transposed vector $\hat{\boldsymbol{e}}_l$, $l = K+1,\ R+1$;

$\otimes$ is the symbol of the matrix Kronecker product; see, e.g., [42–44];

I is the identity matrix, and O is a square zero matrix of appropriate size. If needed, the size is indicated as the suffix.

To simplify the analysis of the multi-dimensional $CTMC\ \xi_t,\ t \geq 0$, having one countable component (i_t) and three finite components, let us enumerate its states in the direct lexicographic order of the components. We call the set of states of this $CTMC$, which have the value i of the countable component i_t, as *level* i of the $CTMC$.

Let Q be the generator of the $CTMC\ \xi_t,\ t \geq 0$.

Theorem 1. *The generator Q has the following block-tridiagonal structure:*

$$Q = \begin{pmatrix} Q_{0,0} & Q_{0,1} & O & O & O & O & \ldots \\ Q_{1,0} & Q_{1,1} & Q_{1,2} & O & O & O & \ldots \\ O & Q_{2,1} & Q_{2,2} & Q_{2,3} & O & O & \ldots \\ O & O & Q_{3,2} & Q_{3,3} & Q_{3,4} & O & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{1}$$

where the non-zero blocks $Q_{i,j}$, $i, j \geq 0$, contain the intensities of the transition of the CTMC from the states that belong to the level i to the states that belong to the level j.

These blocks are defined as follows:

$$Q_{0,0} = D_0,$$

$$Q_{i,i} = D_0 - \mu i I_W,\ i = \overline{1, N-1},$$

$$Q_{N,N} = I_{(K+1)(R+1)} \otimes (D_0 - \mu N I_W) + E_{K+1}^+ \otimes I_{R+1} \otimes D_1 + I_{K+1} \otimes E_{R+1}^+ \otimes D_2 -$$
$$-(\alpha_1 + \gamma_1) C_{K+1} \otimes I_{(R+1)W} - (\alpha_2 + \gamma_2) I_{K+1} \otimes C_{R+1} \otimes I_W +$$
$$+\alpha_1 C_{K+1} E_{K+1}^- \otimes I_{(R+1)W} + \alpha_2 I_{K+1} \otimes C_{R+1} E_{R+1}^- \otimes I_W +$$
$$+(E_{K+1}^- \otimes I_{R+1} + \hat{I}_{K+1} \otimes E_{R+1}^-) \otimes \mu N I_W,$$

$$Q_{i,i} = Q^0 - (i-N)\varphi I_{(K+1)(R+1)W},\ i > N,$$

$$Q^0 = I_{(K+1)(R+1)} \otimes (D_0 - \mu N I_W) + E_{K+1}^+ \otimes I_{R+1} \otimes D_1 + I_{K+1} \otimes E_{R+1}^+ \otimes D_2 -$$
$$-(\alpha_1 + \gamma_1) C_{K+1} \otimes I_{(R+1)W} - (\alpha_2 + \gamma_2) I_{K+1} \otimes C_{R+1} \otimes I_W +$$
$$+\alpha_1 C_{K+1} E_{K+1}^- \otimes I_{(R+1)W} + \alpha_2 I_{K+1} \otimes C_{R+1} E_{R+1}^- \otimes I_W,$$

$$Q_{i,i+1} = D_1 + D_2,\ i = \overline{0, N-2},$$

$$Q_{N-1,N} = \hat{\boldsymbol{e}}_{K+1} \otimes \hat{\boldsymbol{e}}_{R+1} \otimes (D_1 + D_2),$$

$$Q_{i,i+1} = Q^+ = \gamma_1 C_{K+1} E_{K+1}^- \otimes I_{(R+1)W} + \gamma_2 I_{K+1} \otimes C_{R+1} E_{R+1}^- \otimes I_W +$$

$$+\tilde{I}_{K+1} \otimes I_{R+1} \otimes D_1 + I_{K+1} \otimes \tilde{I}_{R+1} \otimes D_2,\ i \geq N,$$
$$Q_{i,i-1} = \mu i I_W,\ i = \overline{1, N-1},$$
$$Q_{N,N-1} = \hat{\boldsymbol{e}}_{K+1}^T \otimes \hat{\boldsymbol{e}}_{R+1}^T \otimes \mu N I_W,$$
$$Q_{i,i-1} = Q^- + (i-N)\varphi I_{(K+1)(R+1)W},\ i > N,$$
$$Q^- = \mu N I_{(K+1)(R+1)W}.$$

Proof. The proof of Theorem 1 was implemented by means of the analysis of the intensities of all possible transitions of the *CTMC* ξ_t during the infinitely small time and is presented below.

The block-tridiagonal structure of the generator Q stems from the fact that requests arrive at the system and depart from it (due to service completion or impatience) only one by one.

The form of the non-zero blocks $Q_{i,j}$, $i, j \geq 0$, is explained as follows:

- **The block** $Q_{0,0}$:
  If the system is empty ($i = 0$), that is all three buffers are empty and all servers are idle, the behaviour of the *CTMC* ξ_t is determined only by the process ν_t. The intensities of its transitions to other states are equal to the non-diagonal elements of the matrix D_0, and the rates of the exit from the corresponding states are determined up to the sign by the diagonal elements of this matrix. Thus, $Q_{0,0} = D_0$.
- **The diagonal entries of the blocks** $Q_{i,i}$, $i \geq 1$:
  These entries are negative. Their modules define the exit rate of the *CTMC* ξ_t from its state. The exit can occur due to the following reasons:
  (a) The underlying process ν_t departs from its current state. The rates of departures are given by the modules of the diagonal elements of the matrix D_0, if $i = \overline{1, N-1}$, or matrix $I_{(K+1)(R+1)} \otimes D_0$, if $i \geq N$.
  (b) Service completion in one busy server occurs. The rates are given by the diagonal elements of the matrix $\mu i I_W$, if $i = \overline{1, N}$, or matrix $\mu N I_{(K+1)(R+1)W}$, if $i > N$.
  (c) A Type-1 request departs from the dedicated intermediate buffer due to impatience or transfer to the main buffer. The rates are given by the matrix $(\alpha_1 + \gamma_1) C_{K+1} \otimes I_{(R+1)W}$, $i \geq N$.
  (d) A Type-2 request departs from the dedicated intermediate buffer due to impatience or transfer to the infinite buffer. The rates are given by the matrix $(\alpha_2 + \gamma_2) I_{K+1} \otimes C_{R+1} \otimes I_W$, $i \geq N$.
  (e) A request departs from the main buffer due to impatience. The rates are given by the matrix $(i-N)\varphi I_{(K+1)(R+1)W}$, $i > N$.
- **The non-diagonal entries of the blocks** $Q_{i,i}$, $i \geq 1$:
  These entries define the rates of transition of the *CTMC* ξ_t within the level i. Such transition rates are given by:
  (a) Non-diagonal entries of the matrices D_0, for $i = \overline{1, N-1}$, or $I_{(K+1)(R+1)} \otimes D_0$, for $i \geq N$, when the process ν_t makes a transition without the generation of a request.
  (b) Entries of the matrix $E_{K+1}^+ \otimes I_{R+1} \otimes D_1$, $i \geq N$, when a Type-1 request arrives and joins the first intermediate buffer.
  (c) Entries of the matrix $I_{K+1} \otimes E_{R+1}^+ \otimes D_2$, $i \geq N$, when a Type-2 request arrives and joins the second intermediate buffer.
  (d) Entries of the matrix $\alpha_1 C_{K+1} E_{K+1}^- \otimes I_{(R+1)W}$, $i \geq N$, when a Type-1 request departs from the intermediate buffer due to impatience.
  (e) Entries of the matrix $\alpha_2 I_{K+1} \otimes C_{R+1} E_{R+1}^- \otimes I_W$, $i \geq N$, when a Type-2 request departs from the intermediate buffer due to impatience.

(f) Entries of the matrix $(E_{K+1}^{-} \otimes I_{R+1} + \hat{I}_{K+1} \otimes E_{R+1}^{-}) \otimes \mu N I_W$ when the main buffer is empty at some service completion moment, while the intermediate buffers are not both empty.

- **The blocks** $Q_{i,i+1}$, $i \geq 0$:
  These blocks define the rates of transition of the $CTMC$ ξ_t when the number of requests in service or in the main buffer increases from i to $i+1$.
  If $i < N-1$, i.e., there is at least one idle server, this occurs when a request of any type arrives at the system and the request starts service. The transition rates of the process ν_t at the moment of a request arrival are determined by the elements of the matrix $D_1 + D_2$. When $i = N-1$, the arrived request occupies the last idle server, and from this moment, the numbers of requests in the intermediate buffers should be counted. Row vector $\hat{e}_{K+1} \otimes \hat{e}_{R+1}$ fixes that both of these buffers are empty. Therefore, the block $Q_{N-1,N}$ is determined by the matrix $\hat{e}_{K+1} \otimes \hat{e}_{R+1} \otimes (D_1 + D_2)$.
  Let now $i \geq N$. The increase of the number of requests in the infinite buffer may occur due to the transition of a request from some intermediate buffer to the infinite buffer. Matrix $\gamma_1 C_{K+1} E_{K+1}^{-}$ determines the rate of transition of a request from Buffer 1 to the infinite buffer under the current number of requests in Buffer 1 and the decrease of the number of requests in Buffer 1. No transition of the number of requests in the infinite buffer and underlying process ν_t can occur simultaneously with the transition of a request to the infinite buffer. Therefore, the intensities of all transitions of the $CTMC$ ξ_t at the moment of the request transition from Intermediate Buffer 1 to the infinite buffer are given by the matrix $\gamma_1 C_{K+1} E_{K+1}^{-} \otimes I_{(R+1)W}$. By analogy, it may be shown that the intensities of the transitions of the $CTMC$ ξ_t at the moment of the request transition from Intermediate Buffer 2 to the infinite buffer are given by the matrix $\gamma_2 I_{K+1} \otimes C_{R+1} E_{R+1}^{-} \otimes I_W$.
  The increase of the number of requests in service and the infinite buffer from i to $i+1$ when i, $i \geq N$, can occur also when Buffer 1 is full and a new Type-1 request arrives. This request pushes the first request out of this buffer to the infinite buffer. The rates of transition of the $CTMC$ ξ_t at this moment are determined by the matrix $\tilde{I}_{K+1} \otimes I_{R+1} \otimes D_1$. By analogy, it may be shown that the intensities of the transitions of the $CTMC$ ξ_t at the moment of the request being pushed out of Intermediate Buffer 2 to the infinite buffer are given by the matrix $I_{K+1} \otimes \tilde{I}_{R+1} \otimes D_2$. As a result, we obtain above-given formula for the block. $Q_{i,i+1}$, $i \geq N$.
- **The blocks** $Q_{i,i-1}$, $i \geq 1$:
  The transitions from the level i to the level $i-1$ are possible at the service completion moments (the corresponding rates are given by the matrix $\mu i I_W$, if $i = \overline{1, N-1}$, or $\mu N I_{(K+1)(R+1)W}$, if $i > N$) and the moments of requests' departure from the infinite buffer due to impatience (the corresponding rates are given by the matrix $(i-N)\varphi I_{(K+1)(R+1)W}$, $i > N$). If $i = N$, the service completion leads to emptying one server. Thus, the block $Q_{N,N-1}$ admits the form $\hat{e}_{K+1}^T \otimes \hat{e}_{R+1}^T \otimes \mu N I_W$, where the column vector $(\hat{e}_{K+1})^T \otimes (\hat{e}_{R+1})^T$ is used to cancel the components describing the numbers of requests in Buffer 1 and Buffer 2 (these numbers are equal to zero by default).

  Theorem 1 is proven. □

3.3. Ergodicity Condition for the Random Process

Having determined the generator of the $CTMC$ ξ_t, we can proceed to the derivation of the ergodicity condition of this $CTMC$.

Theorem 2. *The following statements are true.*

If the requests residing in the infinite buffer are impatient, i.e., the rate φ is positive, then the $CTMC$ ξ_t is ergodic for an arbitrary set of the parameters of the system.

If the requests in this buffer are patient, i.e., the rate φ is equal to zero, then the criterion of the ergodicity of the CTMC ξ_t is the fulfilment of the inequality:

$$\mathbf{y}Q^{+}\mathbf{e} < N\mu \tag{2}$$

where the vector $\mathbf{y}$ *is the unique solution to the system:*

$$\mathbf{y}(Q^{-} + Q^{0} + Q^{+}) = \mathbf{0},\ \mathbf{ye} = 1. \tag{3}$$

Proof. Let us first consider the case $\varphi \neq 0$. In this case, it is easy to verify that there exist the limits:

$$Y_0 = \lim_{i\to\infty} R_i^{-1}Q_{i,i-1} = I,\ Y_1 = \lim_{i\to\infty} R_i^{-1}Q_{i,i} + I = O,\ Y_2 = \lim_{i\to\infty} R_i^{-1}Q_{i,i+1} = O \tag{4}$$

where the matrix R_i is a diagonal matrix with the diagonal entries defined by the corresponding diagonal entries of the matrix $-Q_{i,i},\ i \geq 0$. Therefore, the $CTMC$ ξ_t belongs to the class of continuous-time asymptotically quasi-Toeplitz–Markov chains ($AQTMC$); see [36,45]. It follows from [45] that the sufficient condition for the ergodicity of the Markov chain ξ_t is the fulfilment of the inequality:

$$\mathbf{w}Y_0\mathbf{e} > \mathbf{w}Y_2\mathbf{e} \tag{5}$$

where the vector $\mathbf{w}$ is the unique solution to the system:

$$\mathbf{w}(Y_0 + Y_1 + Y_2) = \mathbf{w},\ \mathbf{we} = 1.$$

It is easy to check that, for the considered $CTMC$ ξ_t with the limiting matrices defined in (4) and (5), it transforms to the evident inequality $1 > 0$. This proves that the $CTMC$ ξ_t is ergodic for an arbitrary set of the parameters of the system.

Let us now consider the case $\varphi = 0$. In this case, the $CTMC$ ξ_t is the particular case of the quasi-birth-and-death processes (see [27]), and the criterion of the ergodicity the $CTMC$ ξ_t has the form:

$$\mathbf{y}Q^{-}\mathbf{e} > \mathbf{y}Q^{+}\mathbf{e} \tag{6}$$

where the vector $\mathbf{y}$ is the unique stochastic solution to the equation:

$$\mathbf{y}(Q^{-} + Q^{0} + Q^{+}) = \mathbf{0}. \tag{7}$$

Taking into account that $Q^{-} = \mu N I_{(K+1)(R+1)W}$ and, thus, $\mathbf{y}Q^{-}\mathbf{e} = \mu N$, Inequality (6) reduces to (2).

Theorem 2 is proven. □

Remark 1. *It is easy to check that the vector* $\mathbf{y}$ *has the following probabilistic sense. When the main buffer is overloaded, the vector* $\mathbf{y}$ *defines the joint stationary distribution of the number of requests and the underlying process of MMAP in the queueing system with the MMAP arrival process, no buffer, two parallel service groups consisting of K and R servers, correspondingly, and the exponential service time distribution in the servers belonging to the rth group with the rate $\alpha_r + \gamma_r,\ r = 1, 2$. It can be verified that the departure process of successfully serviced requests from this queueing system is the MAP defined by the matrices:*

$$H_0 = Q^{0} + Q^{-},\ H_1 = Q^{+}.$$

The mean departure rate from this system is $\mathbf{y}H_1\mathbf{e} = \mathbf{y}Q^{+}\mathbf{e}$. *In the situation when there are many requests in the main buffer, the discussed process of requests' departure from the system with two service groups defines the arrival process at the main buffer for service in the multi-server system with N servers. Therefore, the process defining the operation of this multi-server system when it is overloaded coincides with the process defining the operation of the MAP/M/N system*

with the MAP defined by the matrices H_0 and H_1 and the service rate in each server equal to μ. For the former system, it is well known that the ergodicity condition is $\mathbf{y}Q^+\mathbf{e} < N\mu$. This inequality, only derived based on intuitive reasoning, coincides with the strictly proven Condition (2) above.

Remark 2. *It can be verified that the obtained Condition (2) in the case of a single server (i.e., $N = 1$) does not coincide with the condition derived for a single-server queue in [39]. This is explained by the different assumptions about the fate of a request arriving when the dedicated intermediate buffer is full. Such a request is assumed to be lost in [39], while in the model under study in this paper, this request pushes out of the intermediate buffer the first request staying there, which joins the main buffer.*

3.4. Computation of the Stationary Distribution of the Random Process

Let the condition for the ergodicity of the *CTMC* ξ_t be fulfilled. This implies that the following limits (stationary of invariant probabilities) exist:

$$\pi(i,0,0,\nu) = \lim_{t\to\infty} P\{i_t = i, k_t = 0, r_t = 0, \nu_t = \nu\}, i = \overline{0,N-1}, \nu = \overline{1,W},$$

$$\pi(i,k,r,\nu) = \lim_{t\to\infty} P\{i_t = i, k_t = k, r_t = r, \nu_t = \nu\},\ i \geq N,\ k = \overline{0,K},\ r = \overline{0,R},\ \nu = \overline{1,W}.$$

We sequentially form the row vectors $\boldsymbol{\pi}(i,k,r)$, $\boldsymbol{\pi}(i,k)$, $\boldsymbol{\pi}_i$ of these probabilities as:

$$\boldsymbol{\pi}(i,0,0) = (\pi(i,0,0,1), \pi(i,0,0,2), \dots, \pi(i,0,0,W)),\ i = \overline{0,N-1},$$

$$\boldsymbol{\pi}(i,0) = \boldsymbol{\pi}(i,0,0),\ i = \overline{0,N-1},\ \boldsymbol{\pi}_i = \boldsymbol{\pi}(i,0),\ i = \overline{0,N-1},$$

$$\boldsymbol{\pi}(i,k,r) = (\pi(i,k,r,1), \pi(i,k,r,2), \dots, \pi(i,k,r,W)), i \geq N, k = \overline{0,K}, r = \overline{0,R},$$

$$\boldsymbol{\pi}(i,k) = (\boldsymbol{\pi}(i,k,0), \boldsymbol{\pi}(i,k,1), \dots, \boldsymbol{\pi}(i,k,R)),\ i \geq N, k = \overline{0,K},$$

$$\boldsymbol{\pi}_i = (\boldsymbol{\pi}(i,0), \boldsymbol{\pi}(i,1), \dots, \boldsymbol{\pi}(i,K)),\ i \geq N.$$

It is well known that the stationary probability vectors $\boldsymbol{\pi}_i$, $i \geq 0$, satisfy the system of equilibrium (or Chapman–Kolmogorov) equations:

$$(\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \dots)Q = \mathbf{0},\ \ (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \dots)\mathbf{e} = 1.$$

In the case of the patient requests in the infinite buffer ($\varphi = 0$), the way of solving this infinite system is well known; see [27,36]. In particular, the vectors $\boldsymbol{\pi}_i$, $i \geq N$, are computed by the formula:

$$\boldsymbol{\pi}_i = \boldsymbol{\pi}_N \mathcal{S}^{i-N},\ i \geq N,$$

where the matrix $\mathcal{S}$ is the minimal non-negative solution of the nonlinear matrix equation:

$$\mathcal{S}^2 Q^- + \mathcal{S}Q^0 + Q^+ = O.$$

The vectors $(\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_N)$ are computed as the unique solution to the finite subsystem of equilibrium equations.

In the case of the impatient requests in the main buffer ($\varphi > 0$), the solution of this infinite system is much more involved. However, it can be solved using the numerically stable methods developed for the *AQTMC*; see [45–47].

4. Performance Measures

To give some insight into the quantitative behaviour of the system, we need to have the possibility to compute some key performance measures of the system. A few of these are listed below.

The mean number of requests in the system is calculated by the formula:

$$L = \sum_{i=1}^{N-1} i\boldsymbol{\pi}(i,0,0)\mathbf{e} + \sum_{i=N}^{\infty}\sum_{k=0}^{K}\sum_{r=0}^{R}(i+k+r)\boldsymbol{\pi}(i,k,r)\mathbf{e}.$$

The mean number of busy servers is calculated as

$$N^{serv} = \sum_{i=1}^{N} i\boldsymbol{\pi}_i\mathbf{e} + N\sum_{i=N+1}^{\infty}\boldsymbol{\pi}_i\mathbf{e}.$$

The mean number of requests in the main buffer is computed by

$$N^{buf} = \sum_{i=N+1}^{\infty}(i-N)\boldsymbol{\pi}_i\mathbf{e}.$$

The mean number of requests in the first buffer is calculated by the formula:

$$N^{buf-1} = \sum_{i=N}^{\infty}\sum_{k=1}^{K} k\boldsymbol{\pi}(i,k)\mathbf{e}.$$

The mean number of requests in the second buffer is calculated by the formula:

$$N^{buf-2} = \sum_{i=N}^{\infty}\sum_{k=0}^{K}\sum_{r=1}^{R} r\boldsymbol{\pi}(i,k,r)\mathbf{e}.$$

The loss probability of an arbitrary Type-1 request from the first buffer due to impatience is calculated by the formula:

$$P_{buf-1}^{loss-1} = \frac{1}{\lambda_1}\sum_{i=N}^{\infty}\sum_{k=1}^{K} k\alpha_1\boldsymbol{\pi}(i,k)\mathbf{e} = \frac{\alpha_1}{\lambda_1}N^{buf-1}.$$

The loss probability of an arbitrary request from the first buffer due to impatience is calculated by the formula:

$$P_{buf-1}^{loss} = \frac{1}{\lambda}\sum_{i=N}^{\infty}\sum_{k=1}^{K} k\alpha_1\boldsymbol{\pi}(i,k)\mathbf{e} = \frac{\alpha_1}{\lambda}N^{buf-1}.$$

The loss probability of an arbitrary Type-2 request from the second buffer due to impatience is calculated by the formula:

$$P_{buf-2}^{loss-2} = \frac{1}{\lambda_2}\sum_{i=N}^{\infty}\sum_{k=0}^{K}\sum_{r=1}^{R} r\alpha_2\boldsymbol{\pi}(i,k,r)\mathbf{e} = \frac{\alpha_2}{\lambda_2}N^{buf-2}.$$

The loss probability of an arbitrary request from the second buffer due to impatience is calculated by the formula:

$$P_{buf-2}^{loss} = \frac{1}{\lambda}\sum_{i=N}^{\infty}\sum_{k=0}^{K}\sum_{r=1}^{R} r\alpha_2\boldsymbol{\pi}(i,k,r)\mathbf{e} = \frac{\alpha_2}{\lambda}N^{buf-2}.$$

The intensity of the output flow of successfully served requests is computed by

$$\lambda_{out} = \sum_{i=1}^{\infty}\min\{i,N\}\mu\boldsymbol{\pi}_i\mathbf{e}.$$

The loss probability of an arbitrary request is calculated by the formula:

$$P_{loss} = 1 - \frac{\lambda_{out}}{\lambda}.$$

The loss probability of an arbitrary request from the main buffer due to impatience is calculated by the formula:

$$P_{buf}^{loss} = \frac{1}{\lambda} \sum_{i=N+1}^{\infty} (i-N)\varphi \boldsymbol{\pi}_i \mathbf{e}.$$

Remark 3. *It should be noted that the following equalities hold well:* $L = N^{serv} + N^{buf} + N^{buf-1} + N^{buf-2}$ *and* $P_{loss} = P_{buf-1}^{loss} + P_{buf-2}^{loss} + P_{buf}^{loss}$*, which can be used for the control of the accuracy of the computer realisation of the computation of the stationary probability vectors* $\boldsymbol{\pi}_i$, $i \geq 0$, *and the performance characteristics of the model.*

The intensity of the arrival flow of requests at the main buffer or directly at the servers is computed by

$$\lambda_{arr} = \lambda - \lambda_1 P_{buf-1}^{loss-1} - \lambda_2 P_{buf-2}^{loss-2} = \lambda - \lambda(P_{buf-1}^{loss} + P_{buf-2}^{loss}).$$

The probability that an arbitrary Type-1 request will start servicing in the system immediately upon arrival is calculated by the formula:

$$P^{imm-1} = \frac{1}{\lambda_1} \sum_{i=0}^{N-1} \boldsymbol{\pi}_i D_1 \mathbf{e}.$$

The probability that an arbitrary Type-2 request will start service in the system immediately upon arrival is calculated by the formula:

$$P^{imm-2} = \frac{1}{\lambda_2} \sum_{i=0}^{N-1} \boldsymbol{\pi}_i D_2 \mathbf{e}.$$

The probability that an arbitrary Type-1 request will be selected for service from the first buffer without visiting the main buffer is calculated by the formula:

$$P^{choose-1} = \frac{1}{\lambda_1} \sum_{k=1}^{K} N\mu \boldsymbol{\pi}(N,k)\mathbf{e}.$$

The probability that an arbitrary Type-2 request will be selected for service from the second buffer without visiting the main buffer is calculated by the formula:

$$P^{choose-2} = \frac{1}{\lambda_2} \sum_{r=1}^{R} N\mu \boldsymbol{\pi}(N,0,r)\mathbf{e}.$$

The probability that an arbitrary Type-1 request upon arrival in the system will find the first buffer full and the first request from this buffer will go to the main buffer is calculated by the formula:

$$P^{push-1} = \frac{1}{\lambda_1} \sum_{i=N}^{\infty} \boldsymbol{\pi}(i,K) I_{R+1} \otimes D_1 \mathbf{e}.$$

The probability that an arbitrary Type-2 request upon arrival in the system will find the second buffer full and the first request from this buffer will go to the main buffer is calculated by the formula:

$$P^{push-2} = \frac{1}{\lambda_2} \sum_{i=N}^{\infty} \sum_{k=0}^{K} \boldsymbol{\pi}(i,k,R) D_2 \mathbf{e}.$$

5. Numerical Examples

The arrival flow of requests was modelled by the $MMAP$ arrival process defined by the following matrices:

$$D_0 = \begin{pmatrix} -51.0796 & 0.7866 & 0.7224 \\ 0.2904 & -4.4644 & 0.4 \\ 0.592 & 0.7748 & -3.5052 \end{pmatrix},$$

$$D_1 = \begin{pmatrix} 14.5453 & 0.28014 & 0.04578 \\ 0.02046 & 1.00008 & 0.11166 \\ 0.0054 & 0.1533 & 0.48282 \end{pmatrix}, D_2 = \begin{pmatrix} 33.9389 & 0.65366 & 0.10682 \\ 0.04774 & 2.33352 & 0.26054 \\ 0.0126 & 0.3577 & 1.12658 \end{pmatrix}.$$

The total rate of requests' (priority and non-priority) arrival at the system is $\lambda = 10.0009$. The coefficient of correlation of successive inter-arrival times in this arrival process is 0.300005, and the squared coefficient of variation is 4.00035. The average intensity of priority (Type-1) requests' arrival is $\lambda_1 = 3.00027$, and the average intensity of non-priority (Type-2) requests' arrival is $\lambda_2 = 7.00063$.

The intensities of impatience in the first and the second buffers are equal to $\alpha_1 = 0.03$ and $\alpha_2 = 0.01$; the intensities of the transitions from the first and the second buffers to the main buffer are $\gamma_1 = 0.5$ and $\gamma_2 = 0.2$, respectively. The mean service rate is $\mu = 1$.

We present the results of two experiments. In the first experiment, we fixed the capacities of the intermediate buffers and show the impact of the number of servers N and the impatience rate φ in the main buffer. In the second experiment, we fixed the values of N and φ and demonstrate the effect of changing the capacities K and R of the intermediate buffers.

Experiment 1. We assumed that the capacities of the intermediate buffers are $K = 10$ for priority requests and $R = 15$ for non-priority requests. Let us vary the intensity of the impatience φ over the interval [0.1,1] with a step of 0.1, and the number of servers N was varied over the interval $[1, 40]$ with a step of 1.

Figures 2–5 illustrate the dependence of the mean number of requests in the system L, the mean number of busy servers N^{serv}, and the mean number of requests N^{buf-1} in the first buffer and N^{buf-2} in the second buffer on the values of the intensity φ and the number of servers N.

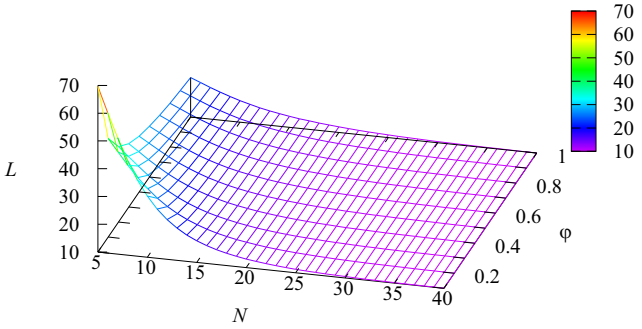

Figure 2. Dependence of the mean number of requests in the system L on φ and N.

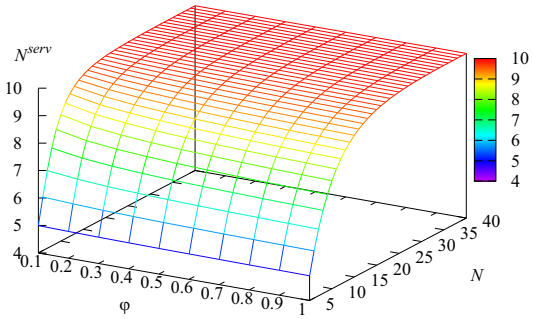

Figure 3. Dependence of the mean number of busy servers N^{serv} on φ and N.

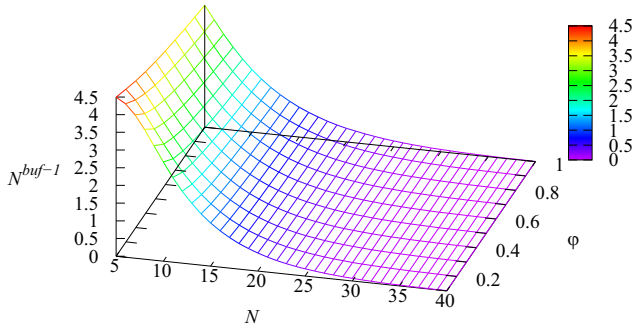

Figure 4. Dependence of the mean number of requests N^{buf-1} in the first buffer on φ and N.

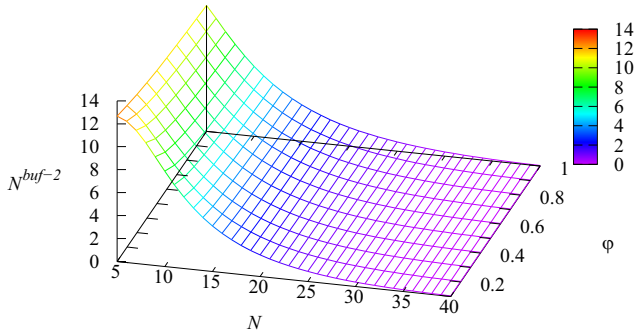

Figure 5. Dependence of the mean number of requests N^{buf-2} in the second buffer on φ and N.

It is evidently seen in Figure 2 that the mean number of requests in the system L is huge (about 70) when the number N of servers is relatively small ($N = 5$) and the impatience rate φ is also small. An explanation of this fact follows from Figure 3. It is seen in this figure that, when the number N of servers is 5, the average number N^{serv} of busy servers is close to 5. This means that all available servers are practically always busy. It is well known that, in such a situation, the queue length is very long. Because the average number of requests in the main buffer N^{buf} is the summand in the right-hand side of the expression $L = N^{serv} + N^{buf} + N^{buf-1} + N^{buf-2}$, it is easy to understand why L is huge when the number N of servers and the impatience rate are small. As expected, the value of L and all summands essentially decrease when the number of servers N and impatience rate φ increase. For large values of N ($N \geq 35$), the mean number of busy servers reduces to about 10, while the values of other summands become practically negligible. The influence of the impatience rate φ is essential only when the number N of servers is small. When it is sufficiently large, service is provided quickly, the main buffer is practically always empty, and requests very rarely depart from this buffer due to impatience.

It should be noted, based on Figures 4 and 5, that the average number N^{buf-2} of requests residing in the second buffer is essentially larger than the mean number N^{buf-1} of requests in the first buffer. This takes place because the arrival rate at the second buffer is 2.33-times higher than the arrival rate at the first buffer and due to the priority provided to

Type-1 requests via the higher transition rate to the main buffer and the smaller capacity of the intermediate buffer. For a small number N of servers, on average, only about 45 percent of the first buffer is occupied. The average percentage of occupation of the second buffer is about 1.9-times higher.

Figures 6–9 illustrate the dependence of the loss probability P^{loss}_{buf-1} of an arbitrary request from the first buffer, the loss probability P^{loss}_{buf-2} of an arbitrary request from the second buffer, the loss probability P^{loss}_{buf} of an arbitrary request from the main buffer, and the loss probability of an arbitrary request P_{loss} (all these losses occur due to impatience) on the values of the rate of impatience φ and the number of servers N. The shapes of the surfaces presented in these figures are similar to the shapes of surfaces presented in Figures 2–5. This was as anticipated because all the mentioned losses occurred due to impatience, and thus, the probabilities P^{loss}_{buf-1}, P^{loss}_{buf-2} and P^{loss}_{buf} of the losses from the two intermediate buffers and the main buffer were proportional (with the weights defined by the respective impatience rates) to the mean number of requests in each buffer. The probability P^{loss} of an arbitrary request loss is the sum of the loss probabilities P^{loss}_{buf-1}, P^{loss}_{buf-2} and P^{loss}_{buf}, which is confirmed by Figures 6–9. It may be concluded from these figures that all loss probabilities essentially depend on N. The dependence on φ is weaker, especially for the loss probabilities from the intermediate buffers.

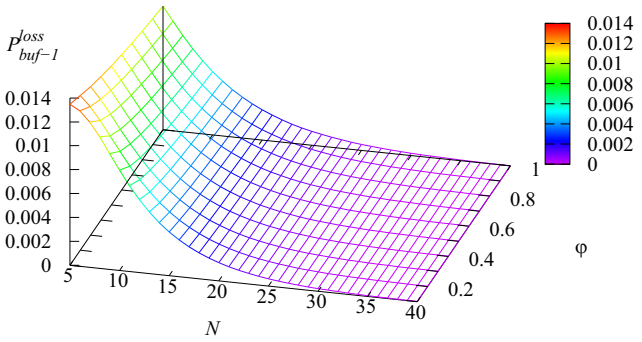

Figure 6. Dependence of the loss probability P^{loss}_{buf-1} of an arbitrary request from the first buffer on φ and N.

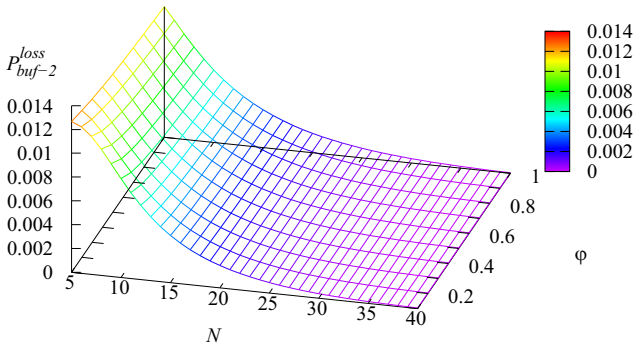

Figure 7. Dependence of the loss probability P^{loss}_{buf-2} of an arbitrary request from the second buffer on φ and N.

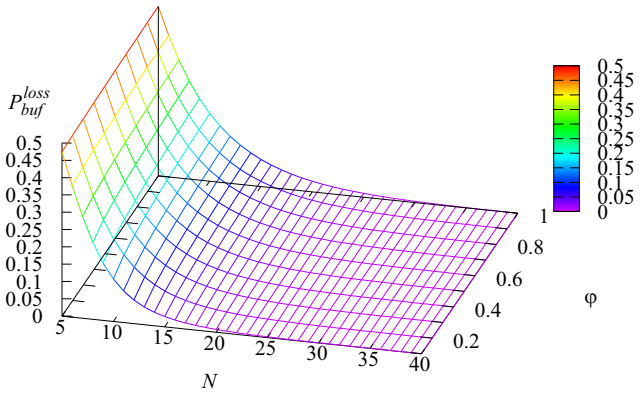

Figure 8. Dependence of the loss probability P_{buf}^{loss} of an arbitrary request from the main buffer on φ and N.

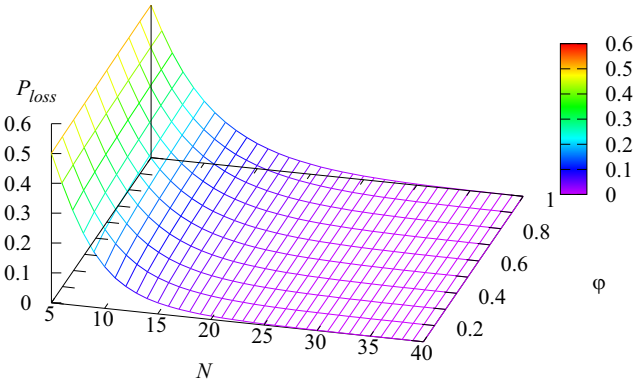

Figure 9. Dependence of the loss probability P_{loss} of an arbitrary request on φ and N.

Let us briefly illustrate the possibility of the use of the obtained results for the managerial goals. We considered the problem of the optimal choice of the number N of servers to maximise the profit of the system. It was assumed that the profit earned by the system during a unit of time under the fixed number N of servers is evaluated by the profit function:

$$E(N) = a\lambda_{out} - b_1\lambda_1 P_{buf-1}^{loss-1} - b_2\lambda_2 P_{buf-2}^{loss-2} - c\lambda_{arr} P_{buf}^{loss} - dN$$

where a is the profit gained via service provision to one request, b_k is the penalty of the system paid for the loss of one request from the kth intermediate buffer, $k = 1, 2$, c is the penalty of the system paid for the loss of a request from the main buffer, and d is the cost of the maintenance of one server per unit of time.

Let the cost coefficients $a,\ b_1,\ b_2,\ c,\ d$ be fixed as follows:

$$a = 1,\ b_1 = 2,\ b_2 = 1,\ c = 1.5,\ d = 0.05.$$

The surface showing the dependence of the cost function $E(N)$ on the number of servers N and the impatience rate φ is presented in Figure 10.

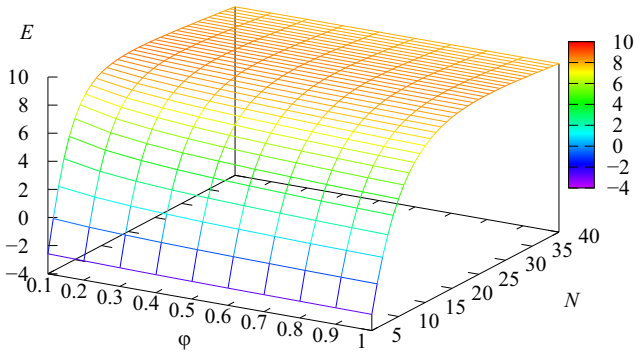

Figure 10. Dependence of the profit function $E(N)$ on the number of servers N and the impatience rate φ.

The optimal values of N were separately computed for each fixed value of the impatience rate φ. Table 1 contains the optimal value N^* of N and the corresponding optimal value $E(N^*)$ for ten fixed values of φ.

Table 1. Optimal values of the number of servers and the profit function for various values of φ.

Rate φ	Optimal Value of the Profit Function E^*	Optimal Value N^* of N
0.1	8.72093	21
0.2	8.6376	23
0.3	8.58863	24
0.4	8.55435	24
0.5	8.52905	25
0.6	8.50816	25
0.7	8.49059	26
0.8	8.47683	26
0.9	8.46432	26
1	8.45287	26

It is clear that the increase of the impatience rate φ implies a larger value of the probability P_{buf}^{loss}. To decrease this probability, it is necessary to decrease the mean number of requests in the buffer, which can be achieved via the increase of the number of servers N. This explains the growth of N^* when φ increases observed in Table 1. When the number of servers is sufficiently large, the servers succeed in providing service at such a speed that the queue length in the main buffer is very small and the increase of the impatience rate φ practically does not have an impact on the value of the profit function.

Example 1. *Let us now fix the number of servers* $N = 15$ *and the impatience rate in the main buffer* $\phi = 0.05$. *To show the impact of the capacities of the intermediate buffers R and K, we computed the values of various performance measures for the values of R and K varying in the range from 1 to 20 with a step of one.*

Figures 11–13 illustrate the dynamics of the mean number of requests N^{buf-1} and N^{buf-2} in the first and second buffers and the mean number of requests N^{buf} in the infinite buffer.

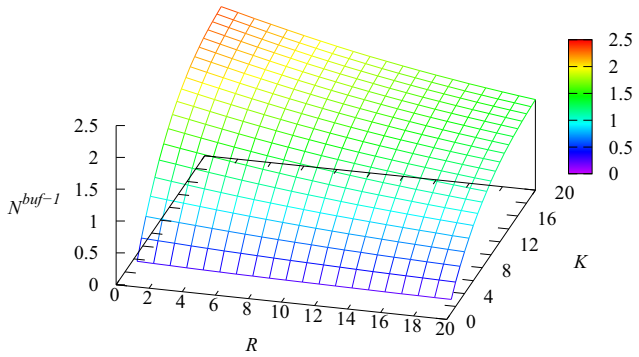

Figure 11. Dependence of the mean number of requests N^{buf-1} in the first buffer on K and R.

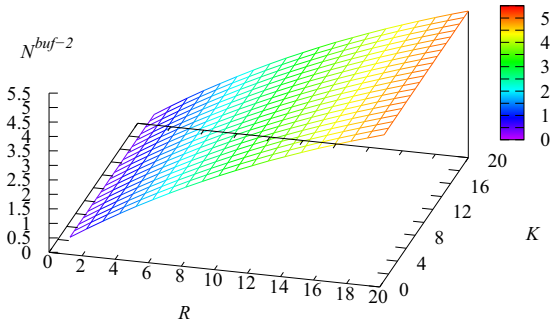

Figure 12. Dependence of the mean number of requests N^{buf-2} in the second buffer on K and R.

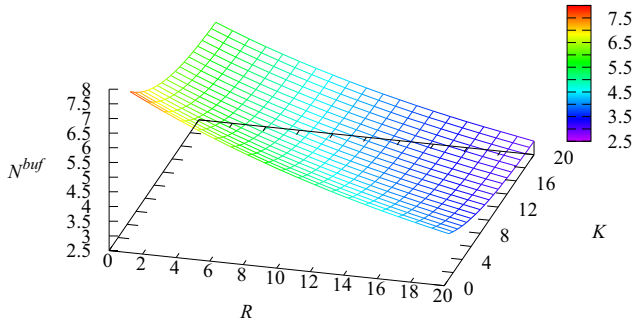

Figure 13. Dependence of the mean number of requests N^{buf} in the main buffer on K and R.

It is natural that the value of the mean number N^{buf-1} of requests in the first buffer increases when the capacity K of this buffer increases. The essential growth of N^{buf-1} when the capacity R of the second buffer decreases is explained as follows. When R decreases, more Type-2 requests are pushed out from the second buffer due to the arrival of new Type-2 requests. Therefore, the probability that the main buffer is empty decreases and the chances of Type-1 requests to realise their priority via the privilege to be taken for service when the main buffer becomes empty decrease. This leads to the increase of N^{buf-1}. The maximum of the mean number N^{buf-2} of requests in the second buffer is essentially larger than the maximum of N^{buf-1}. This occurs due to the higher arrival rate of Type-2 requests and the lower rate of transition from the intermediate buffer to the main one. However, the influence of the relation of the capacities of the intermediate buffers is also high. If R is small, clearly, this reduces the part of the priority of Type-1 requests achieved via their higher rate of transition from the intermediate buffer to the main buffer.

The maximum of the mean number N^{buf} of requests in the main buffer is achieved for a small capacity R of the second buffer. The arrival rate at this buffer is essentially higher than at the first buffer, and a small R leads to the short stay of Type-2 requests in the second buffer before being pushed out to the main buffer. When both K and R are larger, requests stay in the intermediate buffer during a more or less long time. This long delay reduces the

burstiness of the flow to the main buffer (we remind that the coefficient of correlation in the arrival process is about 0.3, which is rather large), while it is known in the literature that lower burstiness (or higher regularity) in the arrival process leads to a shorter queue in the system.

Figures 14 and 15 depict the dependence of the probability $P^{choose-k}$ that an arbitrary Type-k request will be selected for service from the k buffer, $k = 1, 2$, without visiting the main buffer on K and R. Recall that, for Type-1 requests, this can happen if all N servers are busy, the main buffer is empty, the service in one of the servers is completed, and the first intermediate buffer is not empty. For Type-2 requests, this can happen if all N servers are busy, the main buffer is empty, the service in one of the servers is completed, the first intermediate buffer is empty, and the second intermediate buffer is not empty. Figure 14 correlates with Figure 13. When K and R are large, the mean number N^{buf} is the minimal. Thus, the probability that the infinite buffer is empty at the moment of a server releasing is high and the probability $P^{choose-1}$ is large. Analogously, when K and R are small (the main role is played by the capacity R of the intermediate buffer, which stores a more intensive flow of Type-2 requests), the mean number N^{buf} is the max. Thus, the probability that the infinite buffer is empty at the moment of a server releasing is small, and correspondingly, the probability $P^{choose-1}$ is small. The growth of $P^{choose-1}$ with the increase of K (which is sharper when K is still relatively small) stems from the increase of the probability that the first buffer will not be empty at the moment of a server releasing. The reason for the growth of $P^{choose-2}$ with the increase of R is similar. The impact of the variation of K on the value of $P^{choose-2}$ is weak.

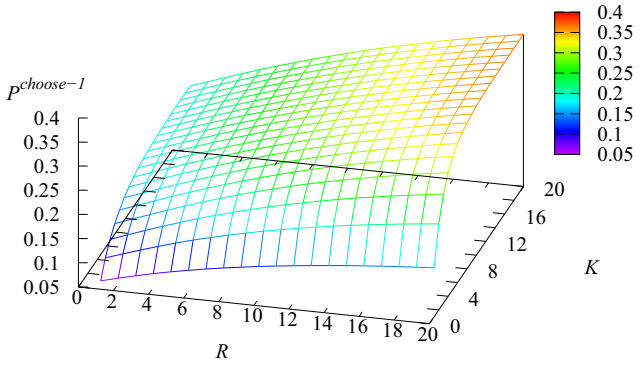

Figure 14. Dependence of the probability $P^{choose-1}$ that an arbitrary Type-1 request will be selected for service from the first buffer without visiting the main buffer on K and R.

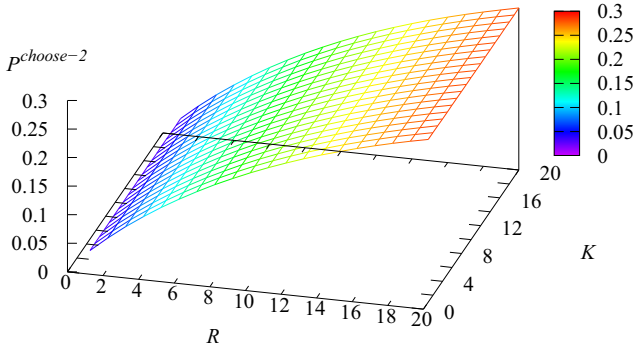

Figure 15. Dependence of the probability $P^{choose-2}$ that an arbitrary Type-2 request will be selected for service from the second buffer without visiting the main buffer on K and R.

Figures 16–19 show the dependence on K and R of the following loss probabilities: the probabilities P^{loss}_{buf-k} of an arbitrary request loss from the kth buffer, $k = 1, 2$, the probability P^{loss}_{buf} of an arbitrary request loss from the main buffer, and the probability P_{loss} of an arbitrary request loss.

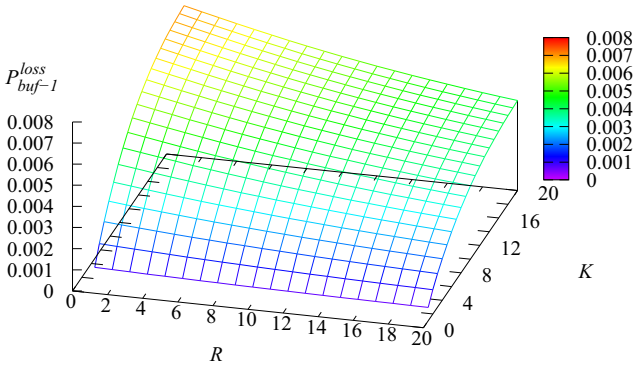

Figure 16. Dependence of loss probability P^{loss}_{buf-1} of an arbitrary request from the first buffer on K and R.

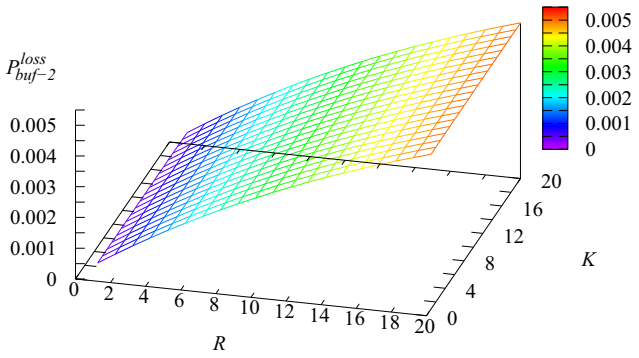

Figure 17. Dependence of the loss probability P^{loss}_{buf-2} of an arbitrary request from the second buffer on K and R.

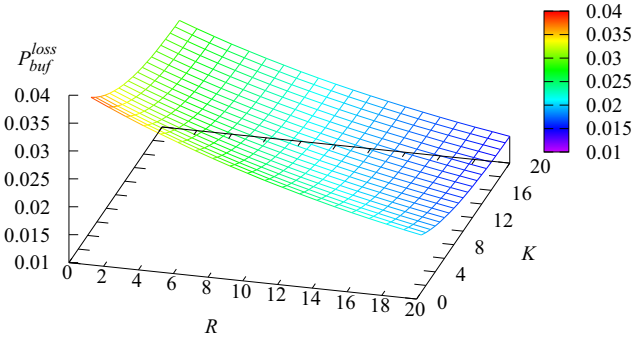

Figure 18. Dependence of the loss probability P^{loss}_{buf} of an arbitrary request from the main buffer on K and R.

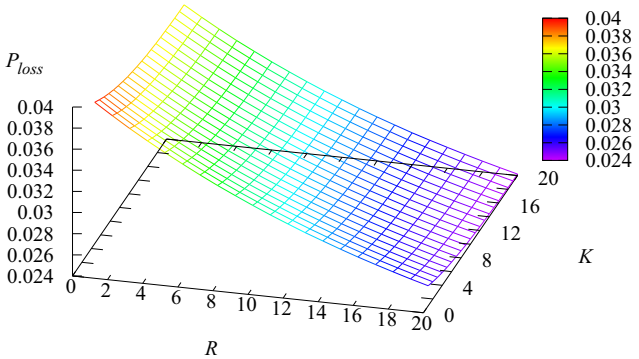

Figure 19. Dependence of the loss probability P_{loss} of an arbitrary request on K and R.

Because an arbitrary request loss in the intermediate buffer is due to impatience, it is clear that the loss probability P_{buf-k}^{loss} increases with the increase of the capacity of the kth intermediate buffer, $k = 1, 2$. Because the capacities of these buffers are relatively small and the impatience rate in the infinite main buffer is larger compared to the rates in the intermediate buffers, the probability P_{buf}^{loss} of an arbitrary request from the main buffer also is larger. As is seen from Figures 16–19, this probability is a dominating summand at the right-hand side of the relation $P_{loss} = P_{buf}^{loss} + P_{buf-1}^{loss} + P_{buf-2}^{loss}$. The decrease of the probability P_{buf}^{loss} when the capacity R grows is explained by the the increase of the probability P_{buf-2}^{loss}, leading to the decrease of the arrival rate at the main buffer, the decrease of the queue length in this buffer, and eventually, the decrease of the rate of requests' departure from the main buffer due to impatience.

The dependence of the probabilities P^{push-k} that an arbitrary Type-k request upon arrival in the system will find the kth buffer full, $k = 1, 2$, and the first request from this buffer will go to the main buffer on K and R is shown in Figures 20 and 21.

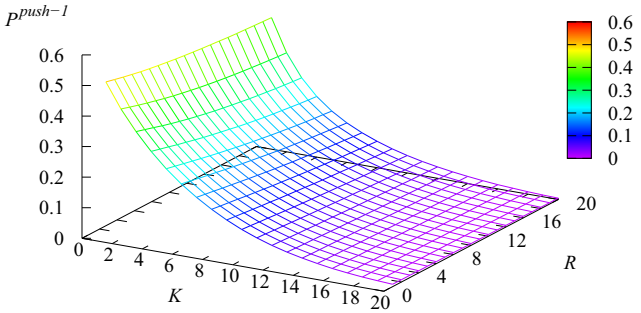

Figure 20. Dependence of the probability P^{push-1} that an arbitrary Type-1 request upon arrival will push the first request from the intermediate buffer to the main buffer on K and R.

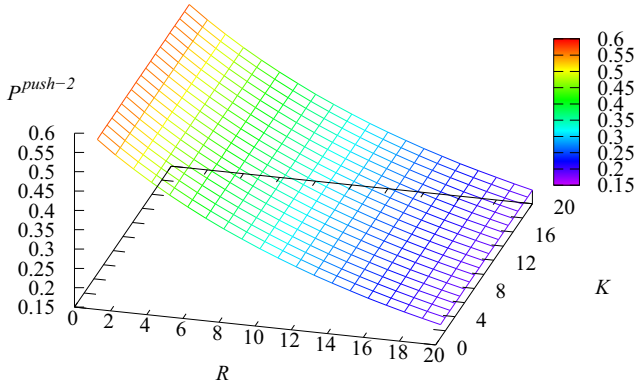

Figure 21. Dependence of the probability P^{push-2} that an arbitrary Type-2 request upon arrival will push the first request from the intermediate buffer to the main buffer on K and R.

As expected, the probabilities P^{push-k} are maximal when the capacity of the kth buffer is small and essentially decrease when this capacity increases. Furthermore, these probabilities are weakly sensitive with respect to the capacity of another buffer.

6. Conclusions

In this paper, a new flexible mechanism for providing preference to one type of request, which was offered in [39] for a single-server priority queueing system, was applied to a multi-server queueing system. The priority is granted via the introduction of intermediate buffers having finite capacities. Requests of different priorities are distinguished by the rate of transfer from these buffers to the main buffer and the rates of departing from the buffers without service. The arriving process of requests can be correlated and have a large inter-arrival time variance. Requests staying in the main buffer receive service in

the order of their transition to this buffer. A suitable choice of the rates of transition from the intermediate buffers to the main buffer, as well as the capacities of the intermediate buffers allows optimising the operation of the system. The impact of the capacities of the intermediate buffers, the number of servers, and the impatience rate in the main buffer was illustrated via the presented results of the numerical experiment.

The results obtained in the paper can be used for the optimisation of various real-world systems with heterogeneous requests having different importance for the system. They can be extended to the cases of the batch arrival of requests, the phase-type distribution of the service time and the patience time in the intermediate buffers, the possibility of server breakdowns or errors occurring during the service, an arbitrary number of priority classes, etc.

Author Contributions: Conceptualisation, K.S. and A.D.; methodology, O.D. and A.D.; software, O.D.; validation, O.D.; formal analysis, K.S., O.D. and A.D.; investigation, K.S., O.D. and A.D.; writing, original draft preparation, K.S. and A.D.; writing, review and editing, K.S., O.D. and A.D.; supervision, K.S. and A.D.; project administration, O.D. All authors have read and agreed to the published version of the manuscript.

Funding: This paper has been supported by the RUDN University Strategic Academic Leadership Program.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Glynn, P.W. Queueing theory: Past, present, and future. *Queueing Syst.* **2022**, *100*, 169–171. [CrossRef]
2. Elalouf, A.; Wachtel, G. Queueing problems in emergency departments: A review of practical approaches and research methodologies. *Oper. Res. Forum* **2022**, *3*, 1–46. [CrossRef]
3. Rece, L.; Vlase, S.; Ciuiu, D.; Neculoiu, G.; Mocanu, S.; Modrea, A. Queueing Theory-Based Mathematical Models Applied to Enterprise Organization and Industrial Production Optimization. *Mathematics* **2022**, *10*, 2520. [CrossRef]
4. Hu, Y.; Luo, X.; Bai, D. Passenger congestion alleviation in large hub airport ground-access system based on queueing theory. *Transp. B Transp. Dyn.* **2022**, 257–278. . [CrossRef]
5. Jia, W.; Huang, Y.L.; Zhao, Q.; Qi, Y. Modeling taxi drivers' decisions at airport based on queueing theory. *Res. Transp. Econ.* **2022**, *92*, 101093. [CrossRef]
6. Chakravarthy, S.R. *Introduction to Matrix-Analytic Methods in Queues 1: Analytical and Simulation Approach–Basics*; ISTE Ltd.: London, UK; John Wiley and Sons: New York, NY, USA, 2022.
7. Chakravarthy, S.R. *Introduction to Matrix-Analytic Methods in Queues 2: Analytical and Simulation Approach–Queues and Simulation*; ISTE Ltd.: London, UK; John Wiley and Sons: New York, NY, USA, 2022.
8. Baccara, M.; Lee, S.; Yariv, L. Task allocation and on-the-job training. *J. Econ. Theory* **2023**, *207*, 105587. [CrossRef]
9. Jenčová, E.; Koščák, P.; Koščáková, M. Dimensioning the Optimal Number of Parallel Service Desks in the Passenger Handling Process at Airports Considered as a Queueing System—Case Study. *Aerospace* **2023**, *10*, 50. [CrossRef]
10. Jaiswal, N.K. *Priority Queues*; Academic Press: New York, NY, USA, 1968.
11. Takagi, H. *Queueing Analysis: A Foundation of Performance Evaluation, Volume 1: Vacation and Priority Systems*; Elsevier: Amsterdam, The Netherlands, 1991.
12. Kleinrock, L. *Queueing Systems, Volume 2: Computer Applications*; Wiley: New York, NY, USA, 1976.
13. Gnedenko, B.V.; Danielyan, E.A.; Dimitrov, B.N.; Klimov, G.P.; Matvejev, V.F. *Priority Queueing Systems*; Moscow State University: Moscow, Russian, 1973. (In Russian)
14. Lee, S.; Dudin, A.; Dudina, O.; Kim, C. Analysis of a priority queueing system with the enhanced fairness of servers scheduling. *J. Ambient. Intell. Humaniz. Comput.* **2022**, 1–13. [CrossRef]
15. Walraevens, J.; Van Giel, T.; De Vuyst, S.; Wittevrongel, S. Asymptotics of waiting time distributions in the accumulating priority queue. *Queueing Syst.* **2022**, *101*, 221–244. [CrossRef]
16. Walraevens, J. Asymptotics in priority queues: From finite to infinite capacities. *Queueing Syst.* **2022**, *100*, 361–363. [CrossRef]
17. Alipour-Vaezi, M.; Aghsami, A.; Jolai, F. Prioritizing and queueing the emergency departments' patients using a novel data-driven decision-making methodology, a real case study. *Expert Syst. Appl.* **2022**, *195*, 116568. [CrossRef]
18. Bai, X.; Jin, S. Performance analysis of an energy-saving strategy in cloud data centres based on a $MMAP[K]/M[K]/N_1 + N_2$ non-preemptive priority queue. *Future Gener. Comput. Syst.* **2022**, *136*, 205–220. [CrossRef]
19. Wang, Z.; Fang, L. The effect of customer awareness on priority queues. *Nav. Res. Logist.* **2022**, *69*, 801–815. [CrossRef]

20. Chen, G.; Xia, L.; Jiang, Z.; Peng, X.; Xu, H. A two-class $MAP/PH/1$ weighted fair queueing system and its application to telecommunications. *J. Ambient. Intell. Humaniz. Comput.* **2022**, 1–12. [CrossRef]
21. Li, S.; Xu, Q.; Gaber, J.; Yang, N. Modeling and Performance Analysis of Channel Assembling Based on Ps-rc Strategy with Priority Queues in CRNs. *Wirel. Commun. Mob. Comput.* **2022**. [CrossRef]
22. Raj, R.; Jain, V. Optimization of traffic control in $MMAP[2]/PH[2]/S$ priority queueing model with PH retrial times and the preemptive repeat policy. *J. Ind. Manag. Optim.* **2023**, *19*, 2333–2353. [CrossRef]
23. Rykov, V.V.; Lembert, E. Optimal dynamic priorities in single-line queueing systems. *Eng. Cybern.* **1967**, *5*, 21–30.
24. Rykov, V.V. *Controllable Queueing Systems*; Itogi Nauki i Tekhniki, Teoriya Veroyatnostei, Matematicheskaya Statistika, Teoreticheskaya Kibernetika; CRC Press: Boca Raton, FL, USA, 1975; Volume 12, pp. 43–153.
25. Klimenok, V.; Dudin, A.; Dudina, O.; Kochetkova, I. Queuing System with Two Types of Customers and Dynamic Change of a Priority. *Mathematics* **2020**, *8*, 824. [CrossRef]
26. Lee, S.K.; Dudin, S.; Dudina, O.; Kim, C.S.; Klimenok, V. A Priority Queue with Many Customer Types, Correlated Arrivals and Changing Priorities. *Mathematics* **2020**, *8*, 1292. [CrossRef]
27. Neuts, M.F. *Matrix-Geometric Solutions in Stochastic Models*; The Johns Hopkins University Press: Baltimore, MD, USA, 1981.
28. He, Q.M. Queues with marked customers. *Adv. Appl. Probab.* **1996**, 28, 567–587. [CrossRef]
29. He, Q.-M. *Fundamentals of Matrix-Analytic Methods*; Springer: New York, NY, USA, 2014.
30. Latouche, G.; Ramaswami, V. *Introduction to Matrix Analytic Methods in Stochastic Modeling*; Society for Industrial and Applied Mathematics: Siam, Thailand, 1999.
31. Chakravarthy, S.R. The Batch Markovian Arrival Process: A Review and Future Work. In *Advances in Probability Theory and Stochastic Processes*; Krishnamoorthy, A., Ed.; Notable Publications, Inc.: Hoboken, NJ, USA, 2001; pp. 21–49.
32. Lucantoni, D.; Meier-Hellstern, K.S.; Neuts, M.F. A single-server queue with server vacations and a class of nonrenewal arrival processes. *Adv. Appl. Prob.* **1990**, *22*, 676–705. [CrossRef]
33. Lucantoni, D. New results on the single server queue with a batch Markovian arrival process. *Stoch. Model.* **1991**, *7*, 1–46. [CrossRef]
34. Neuts, M.F. A versatile Markovian point process. *J. Appl. Prob.* **1979**, *16*, 764–779. [CrossRef]
35. Neuts, M.F. Models based on the Markovian arrival processes. *IEICE Trans. Commun.* **1992**, *75*, 1255–1265.
36. Dudin, A.N.; Klimenok, V.I.; Vishnevsky, V.M. *The Theory of Queuing Systems with Correlated Flows*; Springer Nature: Berlin/Heidelberg, Germany, 2020.
37. Naumov, V.; Gaidamaka, Y.; Yarkina, N.; Samouylov, K. *Matrix and Analytical Methods for Performance Analysis of Telecommunication Systems*; Springer Nature: Berlin/Heidelberg, Germany, 2021.
38. Vishnevskii, V.M.; Dudin, A.N. Queueing systems with correlated arrival flows and their applications to modeling telecommunication networks. *Autom. Remote Control* **2017**, *78*, 1361–1403. [CrossRef]
39. Dudin, S.; Dudina, O.; Samouylov, K.; Dudin, A. Improvement of fairness of non-preemptive priorities in transmission of heterogeneous traffic. *Mathematics* **2020**, *8*, 929. [CrossRef]
40. Jouini, O.; Roubos, A. On multiple priority multi-server queues with impatience. *J. Oper. Res. Soc.* **2014**, *65*, 616–632. [CrossRef]
41. Buchholz, P.; Kemper, P.; Kriege, J. Multi-class Markovian arrival processes and their parameter fitting. *Perform. Eval.* **2010**, *67*, 1092–1106. [CrossRef]
42. Graham, A. *Kronecker Products and Matrix Calculus: With Applications*; Courier Dover Publications: Horwood Chichester, UK, 1981.
43. Steeb, W.-H.; Hardy, Y. *Matrix Calculus and Kronecker Product*; World Scientific Publishing: Singapore, 2011.
44. Horn, R.A.; Johnson, C.R. *Topics in Matrix Analysis*; Cambridge University Press: Cambridge, UK, 1991.
45. Klimenok, V.I.; Dudin, A.N. Multi-dimensional asymptotically quasi-Toeplitz Markov chains and their application in queueing theory. *Queueing Syst.* **2006**, *54*, 245–259. [CrossRef]
46. Dudin, S.; Dudina, O. Retrial multi-server queuing system with PHF service time distribution as a model of a channel with unreliable transmission of information. *Appl. Math. Model.* **2019**, *65*, 676–695. [CrossRef]
47. Dudin, S.; Dudin, A.; Kostyukova, O.; Dudina, O. Effective algorithm for computation of the stationary distribution of multi-dimensional level-dependent Markov chains with upper block-Hessenberg structure of the generator. *J. Comput. Appl. Math.* **2020**, *366*, 112425. [CrossRef]