A SIMPLE DISPATCHING POLICY FOR MINIMIZING MEAN RESPONSE TIME IN NON-OBSERVABLE QUEUES WITH SRPT POLICY OPERATING IN PARALLEL

Mikhail Konovalov and Rostislav Razumchik
Institute of Informatics Problems of the FRC CSC RAS
Vavilova, 44-2, 119333, Moscow, Russia
Email: mkonovalov@ipiran.ru, rrazumchik@ipiran.ru

KEYWORDS

shortest remaining processing time, server farm, non-observable, load balancing, customer assignment, dispatching, mean response time

ABSTRACT

Consider a non-observable system with a single dispatcher and $N \geq 2$ single server queues operating in parallel and independently. Each queue uses shortest remaining processing time discipline for scheduling the jobs. Jobs from a single flow arrive one by one to the dispatcher, which must immediately route them to one of the queues. Each decision is irrevocable. The dispatcher does not have any online information about the system and can base its decisions only on the job size distribution, job's inter-arrival time distribution, server's speeds, time instants of previously arrived jobs and previous routing decisions. Under these conditions, one is interested in the routing policies, which minimize the job's long-run mean response time. New simple single-parameter policy is proposed which is applicable in case of i.i.d. arrivals and i.i.d. service times and which, according to the numerical experiments, always outperforms the optimal probabilistic policy and may outperform the deterministic policy (under medium and low load).

INTRODUCTION

In this paper consideration is given to the problem of optimal scheduling in parallel non-observable single-server queues each with SRPT (shortest remaining processing time) scheduling discipline, fed by a single flow of jobs. The structure of the system is the following. There is one dispatcher, which does not have a queue for storing the jobs and thus routes arriving jobs immediately to one of the queues. Each queue has infinite capacity and a single server, which serves the jobs one by one according to the SRPT discipline (see Schrage and Miller (1966)). The non-observability means that the dispatcher has only static information about the system: cumulative distribution function (CDF) of job's inter-arrival times, CDF of job's size and servers' speeds. Any online information about the system's state (like queue sizes, remaining work etc.) is not available to the dispatcher.

Under the assumptions made above, one in interested in routing policies[1], which minimize job's long-run mean response time.

For the latest literature on non-observable queues and, in particular, dispatching systems one can refer to Anselmi (2017); Brun (2016); Grosof et al. (2019); Hyytiä et al. (2012); Hassin and Snitkovsky (2017); Konovalov and Razumchik (2018); Lingenbrink and Krishnamurthy (2017) and references therein. Performance analysis of queues with SRPT discipline has been the subject of extensive research in the past and still remains to be (see Grosof et al. (2018, 2019); Kruk (2019); Scully et al. (2019) and references therein).

To our knowledge the the problem of optimal routing in unobservable dispatching systems with SRPT scheduling in queues has not been studied before. Under non-observability only two class of routing policies are available to the dispatcher[2]: probabilistic and deterministic.

According to a probabilistic (also known as Bernoulli, random) policy a job is routed to the queue n, $1 \leq n \leq N$, with the probability p_n independently of the previous decisions. If the arrival flow is Poisson with rate λ, then the arrival process to each queue remains Poissonian and the queue n is the M/GI/1-SRPT queue with the arrival rate $p_n\lambda$. If the server's speed of the queue n is $v^{(n)}$, the job size distribution is $B(x)$ (with the mean $\mathbf{E}X = \int_0^\infty x dB(x)$), then, whenever the queue n is stable, the stationary mean sojourn time $\mathbf{E}[T_n]^{\text{SRPT}}$ is equal to

$$\mathbf{E}[T_n]^{\text{SRPT}} = \int_0^\infty \frac{\int_0^x u\left(1 - B\left(uv^{(n)}\right)\right) du}{x^2\left(1 - \lambda p_n \int_0^x u dB\left(uv^{(n)}\right)\right)} dx. \quad (1)$$

The optimal probabilistic routing policy[3], further referred to as RND-opt, is the probability distribution $(p_1, p_2, \ldots, p_N)$ that minimizes the job's mean response time

$$\sum_{n=1}^{N} p_n \mathbf{E}[T_n]^{\text{SRPT}} \quad (2)$$

under the constraint $0 \leq p_n\lambda\mathbf{E}(X/v^{(n)}) < 1$ for each n. This problem does not allow an analytical solution but

[1]Throughout the paper the terms routing policy, dispatching policy and algorithm are used as synonyms.

[2]This is in sharp contrast with the partially and fully observable dispatching systems, for which a wider range of routing policies exits.

[3]This is a well-known traffic/resource allocation problem (see, for example, Ibaraki and Katoh (1988)).

Communications of the ECMS, Volume 34, Issue 1,
Proceedings, ©ECMS Mike Steglich, Christian Mueller,
Gaby Neumann, Mathias Walther (Editors)
ISBN: 978-3-937436-68-5/978-3-937436-69-2(CD) ISSN 2522-2414

usually can be solved numerically at a satisfactory level[4]. If the arrival process is a general renewal process with the mean $1/\lambda$ and SCV (squared coefficient of variation) equal to C_A^2, then it is known that the arrival process to each of the N queues is also a renewal process with the mean $1/(\lambda p_n)$ and SCV equal to $1 + (C_A^2 - 1)p_n$. Each queue is thus a GI/GI/1-SRPT queue and (1)–(2) are not valid any more. We are unaware of any analytic or a simulation procedure for computing the optimal N-tuple $(p_1, p_2, \ldots, p_N)$. Yet simulation seems to be the only way to obtain a reasonable solution. The quality of the solution heavily depends on the algorithm (adaptive or meta-heuristic) used to perform the exhaustive search over the domain.

The other feasible class of policies in the considered setting are known as deterministic policies (see Hordijk and van der Laan (2004)). According to a deterministic policy jobs are dispatched according to the prescribed order i.e. according to the known infinite sequence $a_1, \ldots, a_n, \ldots$, where a_i is the queue number, whereto the i-th arriving job is routed. Round-Robin policy (RR policy), which rotates the jobs between the queues in the cyclic order, is the special case of such a policy (see Combé and Boxma (1994)). In the heterogeneous system i.e. when the servers' speed are not equal, the RR policy may lead to infinite mean response times and thus one has to pick up a different deterministic policy[5]. Finding the optimal deterministic sequence for N single server queues in parallel, is a difficult problem for which no general procedure exists so far. Yet very good results can be achieved by special deterministic sequences – billiard sequences, which can be constructed using greedy algorithms. One of such algorithms further referred to as SG (Special Greedy) is given in (Hordijk and van der Laan, 2004, p.184). According to the SG policy the i-th arriving job is routed to the queue a_i:

$$a_i = \mathrm{argmin}_{1 \le n \le N} \left(\frac{x_n + \kappa^n(i-1)}{p_n} \right), \qquad (3)$$

where $\kappa^n(i)$ is equal to the the number of jobs (among the first i jobs) sent to the queue n so far, p_n are the optimal probabilities of the RND policy (i.e. the solution of (2)) and x_n are properly chosen non-negative rational numbers[6].

The drawback of the SG algorithm as well as of the RND policy is the need to calculate or estimate $(N-1)$ parameters. But as numerical experiments and analytic analysis show (see, for example, Anselmi (2017)), deterministic routing is in general more effective than probabilistic routing with respect to the job's long-run mean response time.

The RND and SG policies seem to be the only policies available in the literature, which are applicable in the considered setting. In this paper a new simple routing policy is being proposed, which can outperform both the RND-opt and the SG policy. This new routing policy, further referred to as the AA policy (Arrival Aware), is based on the following intuitive idea: if the dispatcher can memorize its previous routing decisions and also the time instants at which those decisions were made, then this information must help in the problem of reducing the job's long-run mean response time.

The rest of the paper is organized as follows. In the next section the detailed description of system is given. The third section contains the overview of the AA policy, and is followed by the section with the numerical comparison of the AA policy with the RND-opt and SG algorithms. In the concluding section the discussion of the obtained results is presented.

MODEL DESCRIPTION AND ASSUMPTIONS

The system consists of $N \ge 2$ single server infinite capacity queues, operating in parallel. The queues are numbered from 1 to N. The server's speed of queue n is denoted by $v^{(n)}$, $1 \le n \le N$. The service discipline employed in each queue is SRPT. Jockeying between queues is not allowed. Inter-arrival times between jobs, which arrive one by one, and their sizes are i.i.d. with the known CDF $A(x)$ and $B(x)$ respectively. Upon receiving a job the dispatcher must immediately route it to one of the queues.

Fix an arbitrary integer $i \ge 1$. Let $0 \le t_1 < \cdots < t_i$ denote the arrival instants of the first i jobs and let $y_1, y_2, \ldots, y_{i-1}$ be the first $i-1$ routing decisions i.e. y_j is the server whereto the job arrived at instant t_j was routed. Each y_j takes a value from the set $\{1, 2, \ldots, N\}$. For the i-th job arrived at time instant t_i, the dispatcher in order to make a routing decision may use only the following information:

– the values of $t_1, t_2, \ldots, t_i$,

– the values of $y_1, y_2, \ldots, y_{i-1}$,

– the inter-arrival time distribution $A(x)$ and the jobs size distribution $B(x)$,

– the values $v^{(1)}, v^{(2)}, \ldots, v^{(N)}$ of the servers' speeds.

Online information (like the arriving job size, current queues' sizes etc.) is not available. The objective of the dispatcher is to route jobs in such a way, which minimizes job's long-run mean response time.

[4]Though one may face some technical problems, if $B(x)$ has very large variance.

[5]If the system is homogeneous i.e. $v^{(1)} = v^{(2)} = \cdots = v^{(N)}$ then the intuition suggests that the optimal dispatching policy with respect to the stationary mean response time is the one which maximizes the inter-arrival times to each of the N queues. Thus in the homogeneous case the RR policy is optimal policy irrespective of the inter-arrival time distribution and the job size distribution. To our knowledge the proof of this statement is known only in special cases (see, for example, Ephremides et al. (1980); Liu and Towsley (1994)).

[6]For example, if the servers' speeds are all different, then one can put (as in Anselmi and Gaujal (2011)) $x_n = 1$ if n is the fastest server i.e. if $v^{(n)} = \max_{1 \le j \le N} v^{(j)}$ and $x_n = 0$ otherwise.

OVERVIEW OF THE NEW POLICY

Assume that the system starts working at $t_0 = 0$ and the remaining workload in queue n (including server) at t_0 is equal to $b_n \geq 0$. Denote by $0 < t_1 < \cdots < t_i < \ldots$ the jobs' arrival instants and by $y_1, y_2, \ldots, y_i, \ldots$ the routing decisions. Let $\omega_i^{(n)}$ be the sojourn time in the queue n (i.e. the sum of waiting time and service time) of the i-th job arrived at time instant t_i and routed to queue n. The good routing decision y_i would be such, which prescribes to send the i-th job to the queue for which the value $\omega_i^{(n)}$ is minimum[7]. Unfortunately under the SRPT service discipline it seems to be impossible to compute $\omega_i^{(n)}$ and thus the routing decision y_i based on $\omega_i^{(n)}$ cannot be computed as well. The new routing policy, which is being proposed (AA policy), suggests to keep the same rule for choosing y_i but to replace $\omega_i^{(n)}$ with the other value (see $u_i^{(n)}$ below), which can be computed.

Fix the positive real $c > 0$. Let us associate with the i-th job arriving at the dispatcher, N numbers, say $u_i^{(1)}, \ldots, u_i^{(N)}$, which are defined recursively as follows:

$$\tilde{u}_i^{(n)} = \max\left(0, u_{i-1}^{(n)} - (t_i - t_{i-1})\right), \ 1 \leq n \leq N, \ i \geq 1,$$

$$u_i^{(n)} = \begin{cases} \tilde{u}_i^{(\tilde{y}_i)} + \frac{c}{v^{(\tilde{y}_i)}}, & \text{if } n = \tilde{y}_i, \\ \tilde{u}_i^{(n)}, & \text{otherwise,} \end{cases}$$

where[8]

$$\tilde{y}_i = \text{argmin}_{1 \leq n \leq N} \left(\tilde{u}_i^{(n)} + \frac{c}{v^{(n)}}\right),$$

$$u_0^{(1)} = b_1, \ldots, u_0^{(N)} = b_N.$$

Now everything is ready to define the AA policy[9]: route the i-th job to the queue $y_i = \tilde{y}_i$. The pseudo code for the policy is given below (see Algorithm 1). Unlike the RND and SG policies, the AA policy depends only on the single parameter c, which must be somehow estimated[10].

[7]This idea lead to some fruitful results for unobservable FIFO queues, see Konovalov and Razumchik (2018).

[8]When argmin() is being evaluated, ties are broken in the favour of the fastest server and randomly between the fastest servers.

[9]The idea behind the algorithm is the following. Assume that in addition to the considered (primary) system there are $M \geq 1$ analogous (secondary) systems running in parallel, but which are *fully observable*. The m-th secondary system has a single dispatcher, N servers with speeds $v^{(1)}, v^{(2)}, \ldots, v^{(N)}$, and the job size distribution of the arriving jobs is equal to $B^{(m)}(x)$. Secondary systems do not have independent job arrivals. Instead jobs' arrivals to all M secondary systems are synchronized with the jobs' arrivals to the original system. Upon arrival of the i-th job to the original system, the i-th job arrives to each secondary system. The dispatcher in the m-th secondary system, independently of other dispatchers, based on the job size distribution $B^{(m)}(x)$ and available remaining workloads in the queues, chooses for the i-th job the queue, which minimizes its virtual sojourn time in the m-th system, say $\tilde{y}_i^m$. Note that each $\tilde{y}_i^m$ takes a value in the set $\{1, \ldots, N\}$. Let y_i^* be the most frequent value among $\tilde{y}_i^1, \ldots, \tilde{y}_i^M$. If y_i^* is not unique, then the tie is broken in the favour of the fastest server, and randomly among the fastest servers. Once the y_i^* is chosen all $\tilde{y}_i^m$ are put equal to y_i^* i.e. $\tilde{y}_i^m = y_i^*$, $1 \leq m \leq M$, and the dispatcher in the original system routes the i-th job to server y_i^*. The AA policy is the special case of the described scheme, when $M = 1$ and $B^{(1)}(x) = 0$ for $x \leq c$ and $B^{(1)}(x) = 1$ for $x > c$.

[10]In all the numerical examples presented, the values of c were estimated on the trial-and-error basis using Monte-Carlo simulation.

We put further discussion of the AA policy off to the last section and proceed below with some numerical examples, which demonstrate its performance.

Algorithm 1 High-level description of the implementation procedure for the AA policy

```
function NextDecision($N$, $v^{(1)}, \ldots, v^{(N)}, u_{i-1}^{(1)}, \ldots, u_{i-1}^{(N)}$, $t_i, t_{i-1}, c$)
    for $n = 1 \to N$ do
        $u_i^{(n)} = \max\left(0, u_{i-1}^{(n)} - (t_i - t_{i-1})\right)$
    end for
    $y_i = \text{argmin}_{1 \leq n \leq N} \left(u_i^{(n)} + c/v^{(n)}\right)$
    $u_i^{(y_i)} = u_i^{(y_i)} + c/v^{(y_i)}$
    return $y_i, u_i^{(1)}, \ldots, u_i^{(N)}$
end function
```

[a] The function NextDecision(N,$v^{(1)}, \ldots, v^{(N)}, u_{i-1}^{(1)}, \ldots, u_{i-1}^{(N)}$,$t_i, t_{i-1}$) returns for the i-th arriving job the routing decision y_i based on the i-th job arrival instant t_i and the arrival instant t_{i-1} of the $(i-1)$-th job, servers' speeds $v^{(1)}, \ldots, v^{(N)}$, auxiliary values $u_{i-1}^{(1)}, \ldots, u_{i-1}^{(N)}$ and c.
[b] The values $u_0^{(1)}, \ldots, u_0^{(N)}$ are the initial remaining workloads in the queues (including server). For example, if the whole system is initially empty $u_0^{(1)} = \cdots = u_0^{(N)} = 0$.
[c] The positive real value c is the parameter of the algorithm, which must be set manually.

NUMERICAL EXPERIMENT

Numerical results presented below show the values of the job's long-run mean response[11] time under the three dispatching policies: RND-opt, SG and AA. The probabilities $(p_1, p_2, \ldots, p_N)$ of the RND-opt policy are the solution of (2); the SG policy uses (3) and $(p_1, p_2, \ldots, p_N)$; the AA policy uses Algorithm 1 and the values of the parameter c from Tables 2 and 4.

Two systems are considered: the system with two servers (see Table 1) and 8 servers (see Table 3). In each case servers have different speeds, the incoming flow of jobs is Poisson and the mean job size $\mathbf{E}X$ is equal to 1. The considered job size distributions are: exponential, uniform with SCV$= 0,083$ and bimodal with SCV$= 4$.

The numerical results evidence that if the system's load is not heavy, the AA policy (which is based both on the inter-arrival times and the decision history) always outperforms the RND-opt policy and can be also better than the SG policy. The relative gain (with respect to the RND-opt policy) is higher for the low variable job size distributions and shrinks with the increase of the job size variability and the system's size.

Although under heavy load the AA policy is always better than the RND-opt policy, it seems not to be case when compared with the SG policy. Yet, based on our numerical experiments, it is hard[12] to make a conclusion here.

[11]All the values of the job's long-run mean response (except for the RND-opt policy) were obtained by simulation (within the simulation framework described in Konovalov and Razumchik (2014); Konovalov (2014)).

[12]High variance of the response time under heavy load complicates the matter.

Table 1: Job's long-run mean response time in the system with 2 servers ($N = 2$). The server's speeds are $v^{(1)} = 1/3$ and $v^{(2)} = 2/3$. The job arrival flow is Poisson with rate λ. Mean job size $\mathbf{E}X = 1$. The offered load to the whole system is $\rho = \lambda \mathbf{E}X / \sum_{i=1}^{2} v^{(i)} = \lambda$.

		$\rho = 0,1$	$\rho = 0,2$	$\rho = 0,3$	$\rho = 0,4$	$\rho = 0,5$	$\rho = 0,6$	$\rho = 0,7$
Exp(1)	RND-opt	1,627	1,796	2,033	2,328	2,643	3,035	3,595
	SG	1,627	1,796	2,034	2,274	2,487	2,765	3,190
	AA	1,627	1,785	1,970	2,170	2,413	2,723	3,168
U[0,5; 1,5]	RND-opt	1,630	1,808	2,072	2,384	2,731	3,137	3,887
	SG	1,630	1,808	2,071	2,290	2,426	2,695	3,078
	AA	1,623	1,763	1,916	2,088	2,296	2,570	2,984
Bimodal	RND-opt	1,627	1,795	2,031	2,325	2,641	3,037	3,614
	SG	1,627	1,795	2,031	2,292	2,556	2,888	3,383
	AA	1,627	1,790	2,002	2,232	2,503	2,851	3,362

[a] Exp(1) – exponentially distributed job size with mean 1.
[b] U[0,5; 1,5] – uniformly distributed on [0,5; 1,5] job size.
[c] Binomal – job size distribution with two values 0, 5 and 9 with the probabilities 16/17 and 1/17 correspondingly.

Table 2: The values of the parameter c of the AA policy in Table 1.

	$\rho = 0,1$	$\rho = 0,2$	$\rho = 0,3$	$\rho = 0,4$	$\rho = 0,5$	$\rho = 0,6$	$\rho = 0,7$
Exp(1)	0,705	0,9	1,24	1,24	1,24	1,24	1,2
U[0,5; 1,5]	1,15	1,2	1,15	1,19	1,24	1,21	1,18
Bimodal	0,1	0,6	1,12	1,17	1,25	1,23	1,21

Table 3: Job's long-run mean response time in the system with 8 servers ($N = 8$). The server's speeds are $v^{(1)} = 1/36$, $v^{(2)} = 2/36$, $v^{(3)} = 3/36$, $v^{(4)} = 4/36$, $v^{(5)} = 5/36$, $v^{(6)} = 6/36$, $v^{(7)} = 7/36$ and $v^{(8)} = 8/36$. The job arrival flow is Poisson with rate λ. Mean job size $\mathbf{E}X = 1$. The offered load to the whole system is $\rho = \lambda \mathbf{E}X / \sum_{i=1}^{8} v^{(i)} = \lambda$.

		$\rho = 0,1$	$\rho = 0,2$	$\rho = 0,3$	$\rho = 0,4$	$\rho = 0,5$	$\rho = 0,6$	$\rho = 0,7$
Exp(1)	RND-opt	5,45	6,24	7,08	8,05	9,24	10,84	13,14
	SG	5,19	5,67	6,19	6,81	7,59	8,63	10,06
	AA	5,02	5,51	6,07	6,72	7,54	8,63	10,21
U[0,5; 1,5]	RND-opt	5,48	6,29	7,19	8,25	9,60	11,47	14,36
	SG	5,13	5,48	5,86	6,29	6,76	7,47	8,35
	AA	4,79	5,09	5,42	5,84	6,37	7,07	8,06
Bimodal	RND-opt	5,46	6,24	7,07	8,04	9,26	10,86	13,26
	SG	5,32	5,89	6,57	7,35	8,32	9,66	11,50
	AA	5,24	5,86	6,53	7,32	8,32	9,70	11,70

Table 4: The value of the parameter c of the AA policy in Table 3.

	$\rho = 0,1$	$\rho = 0,2$	$\rho = 0,3$	$\rho = 0,4$	$\rho = 0,5$	$\rho = 0,6$	$\rho = 0,7$
Exp(1)	1,75	1,7	1,45	1,45	1,44	1,336	1,255
U[0,5; 1,5]	1,21	1,23	1,25	1,27	1,27	1,25	1,21
Bimodal	1,5	1,65	1,7	1,59	1,52	1,42	1,4

SUMMARY

According to the numerical experiments, the proposed routing policy outperforms the RND-opt policy across all values of the system's load. With respect to the deterministic policy SG (see (3)), it gives lower long-run mean response[13] time under low and medium load. The relative gain of the AA policy with respect to the SG policy is not high (not greater than 5% in the studied cases), and for job size distributions with low variance it is higher than for those with high variance.

In general, the gain of the AA policy depends on the properties of the job size distribution, on the system's structure and the value of the policy parameter c. As can be seen from the Tables 2 and 4 the value of c seems to depend on the number of servers, their speeds and the system's load. So far we could not find a formula for c. More understanding is needed here.

The performance improvement promised in the Tables 1 and 3, comes almost for free. The new policy can be implemented in the dispatcher at very limited costs

[13]And lower variance in most cases.

(see Algorithm 1). Unlike the SG and RND-opt policy, which depend on $N-1$ parameters, the AA policy depends only on a single parameter, which seems to make the problem of its estimation easier.

It is also worth noticing that the described behaviour of the AA policy (with respect to RND and SG policies) remains the same for i.i.d. inter-arrival times as well as for other service disciplines in the queues (for example, FIFO).

REFERENCES

Anselmi, J. 2017. Asymptotically optimal open-loop load balancing. Queueing Systems. Vol. 87. No. 3-4. Pp. 245–267.

Anselmi, J. and B. Gaujal. (2011) The price of forgetting in parallel and non-observable queues. Perform. Eval. Vol. 68. No. 12. Pp. 1291–1311.

Brun, O. 2016. Performance of non-cooperative routing over parallel non-observable queues. Probability in the Engineering and Informational Sciences. Vol. 30. No. 3. Pp. 455–469.

Combé, M.B., Boxma, O.J. 1994. Optimization of static traffic allocation policies. Theor. Comput. Sci. Vol. 125. No. 1. Pp. 17–43.

Ephremides, A., P. Varaiya, and J. Walrand. 1980. A simple dynamic routing problem. IEEE Transactions on Automatic Control. Vol. 25. No 4. Pp. 690–693.

Grosof, I., Z. Scully, and Mor Harchol-Balter. 2019. Load balancing guardrails: keeping your heavy traffic on the road to low response times. SIGMETRICS Perform. Eval. Rev. 47, 1 (December 2019), 910.

Grosof, I., Z. Scully, and M. Harchol-Balter. 2018. SRPT for multiserver systems. Perform. Eval. Vol. 127-128. Pp. 154–175.

Hassin, R. and R.I. Snitkovsky. 2017. Strategic customer behavior in a queueing system with a loss subsystem. Queueing Systems. Vol. 86. No. 3-4. Pp. 361–387.

Hordijk, A. and D.A. van der Laan. 2004. Periodic routing to parallel queues and billiard sequences. Math. Method. Oper. Res., 2004. Vol. 59. No. 2. Pp. 173–192.

Hyytiä, E., S. Aalto, and A. Penttinen. 2012. Minimizing slowdown in heterogeneous size-aware dispatching systems. SIGMETRICS Perform. Eval. Rev. Vol. 40. No. 1. Pp. 29–40.

Ibaraki, T.I. and N. Katoh. 1988. Resource Allocation Problems. Cambridge: MIT Press.

Kruk, Ł. (2019) Diffusion limits for SRPT and LRPT queues via EDF approximations. In: Phung-Duc T., Kasahara S., Wittevrongel S. (eds) Queueing Theory and Network Applications. QTNA 2019. Lecture Notes in Computer Science. Vol. 11688. Pp. 263–275.

Konovalov, M.G. and R.V. Razumchik. 2018 Improving routing decisions in parallel non-observable queues. Computing. Vol. 100. No. 10. Pp. 1059–1079.

Konovalov, M. and R. Razumchik. 2014. Simulation Of Task Distribution In Parallel Processing Systems. Proceedings of the 6th International Congress on Ultra Modern Telecommunications and Control Systems. Pp. 657–663.

Konovalov, M.G. 2014. Building a simulation model for solving scheduling problems of computing resources. Systems and Means of Informatics. Vol. 24. No. 4. Pp. 45–62. (in Russian)

Lingenbrink, D. and K. Iyer. 2017. Optimal Signaling Mechanisms in Unobservable Queues with Strategic Customers. In Proceedings of the 2017 ACM Conference on Economics and Computation, New York, NY, USA. Pp. 347–347.

Liu, Z. and D. Towsley. 1994. Optimality of the Round-Robin Routing Policy. Journal of Applied Probability. Vol. 31. No. 2. Pp. 466–475.

Schrage, L.E. and L.W. Miller. 1966. The queue M/G/1 with the shortest remaining processing time discipline. Oper. Res. Vol. 14. No. 4. Pp. 670–684.

Scully, Z., M. Harchol-Balter, and A. Scheller-Wolf. 2019. Simple near-optimal scheduling for the M/G/1. SIGMETRICS Perform. Eval. Rev. Vol. 47. No. 2. Pp. 24–26.

AUTHOR BIOGRAPHIES

MIKHAIL KONOVALOV is a Doctor of Sciences in Technics and holds position of the principal scientist at the Institute of Informatics Problems of the Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences (FRC CSC RAS). His research activities are focused on adaptive control of random sequences, modelling and simulation of complex systems. His email address is `mkonovalov@ipiran.ru`.

ROSTISLAV RAZUMCHIK received his Ph.D. degree in Physics and Mathematics in 2011. Since then, he has worked as the leading research fellow at the Institute of Informatics Problems of the Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences (FRC CSC RAS). His current research activities are focused on queueing theory and its applications for performance evaluation of stochastic systems. His email address is `rrazumchik@ipiran.ru`