Journal of Computer and Communications, 2015, 3, 32-45
Published Online July 2015 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2015.37004

Real Time Systems with Nonpreemptive Priorities and Ample Maintenance Facilities

Joseph Kreimer[1], Edward Ianovsky[2]

[1]Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beersheba, Israel
[2]RAVAL ACS Ltd., Beersheba, Israel
Email: kremer@bgu.ac.il, e.ianovsky@hotmail.com

Received 21 June 2015; accepted 17 July 2015; published 20 July 2015

Abstract

We consider a real time data acquisition and processing multiserver system with identical servers (such as unmanned aerial vehicles, machine controllers, overhearing devices, medical monitoring devices, etc.) which can be maintained/programmed for different kinds of activities (e.g. passive or active). This system provides a service for real time tasks arriving via several channels (such as surveillance regions, assembly lines, communication channels, etc.) and involves maintenance. We focus on the worst case analysis of the system with ample maintenance facilities exponentially distributed time to failure and maintenance times. We consider two kinds of models (with and without nonpreemptive priorities) and provide balance equations for steady state probabilities and various performance measures, when both operation and maintenance times are exponentially distributed.

Keywords

Performance, Priority, Queueing, Real Time System, Unmanned Aerial Vehicles

1. Introduction

Real time systems (RTS) are imbedded in most modern technological structures, such as production control systems, robotic and telecommunications systems, radars, self-guided missiles, reconnaissance, aircraft and space stations, etc. These systems are widely used for the monitoring and control of physical processes.

The sustained demands of the environments in which RTS operate pose relatively rigid requirements (usually stated as time constraints) on their performance.

In RTS, a calculation that uses temporally invalid data or an action performed too early/late may be useless,

How to cite this paper: Kreimer, J. and Ianovsky, E. (2015) Real Time Systems with Nonpreemptive Priorities and Ample Maintenance Facilities. *Journal of Computer and Communications*, **3**, 32-45. http://dx.doi.org/10.4236/jcc.2015.37004

and sometimes harmful—even if such a calculation or action is functionally correct. These systems are often associated with applications, in which human lives or expensive machinery may be at stake.

There exists a rich literature covering RTS and different scientific communities are treating various problems in this area. We will site only a small portion of it.

Liu and Layland in their famous work [1] presented analytic combinatorial analysis for multiprogram scheduling on a single processor. They suggested two preemptive priority driven algorithms.

Dhall and Liu [2] treated the problem of specifying an order in which the requests made by a set of periodic real-time jobs are to be executed by a multiprocessor computing system, with the goal of meeting all the deadlines with a minimum number of processors. Xio *et al.* in [3] presented a general cost model and an optimal solution algorithm achieving required performance objective (reliability and availability) for redundant RTS.

Several authors treated real-time Flexible Manufacturing Systems (FMS). Tawegoum *et al.* [4] introduced management functions in real-time monitoring of FMS. Chakravarty and Balakrishnan [5] presented the procedure for the real-time revision of production and inventory schedules in a capacitated FMS.

Efficient metaheuristic methods, such as Tabu Search (Glover and Laguna [6]) and Greedy Randomized Adaptive Search Procedure (Feo *et al.* [7]) were proposed during the last two decades of the previous century. A comprehensive survey on real-time decision problems in OR perspective is given in (Seguin *et al.* [8]). Good survey on applications of Artificial Intelligence to RTS is given in (Strosnider and Paul [9]).

We will focus on RTS with a zero deadline for the beginning of job processing. Queueing of jobs in such systems is impossible, since jobs are executed immediately upon arrival, conditional on system availability.

That part of the job which is not processed immediately is lost forever and cannot be served later.

The following works treated this kind of RTS. Kreimer and Mehrez ([10] and [11]) showed that the non-mix policy of never relieving an operative server maximizes the availability of a single channel RTS involving preventive maintenance and working in general regime with any arrival pattern under consideration and constant service and maintenance times. Kreimer and Mehrez [12] and Kreimer [13] treated muliserver and multichannel (with identical servers and channels) RTS (with unrestricted and restricted number of maintenance facilities respectively), working under maximum load regime (worst case) as finite source queues (Gross and Harris, [14]). Kreimer [15] applied the two dimensional birth and death processes in worst case analysis of a multiserver RTS (with ample and limited maintenance facilities respectively) with two different channels, when both service and maintenance times were exponentially distributed. Ianovsky and Kreimer [16] obtained optimal assignment/routing probabilities to maximize the availability of RTS with limited maintenance facilities, for large number of servers and two different channels. Kreimer [17] obtained the steady state probabilities for RTS with arbitrary number of different channels operating under a maximum load regime. Kreimer [18] considered multiserver and multichannel RTS working in general regime. Shimshi and Kreimer [19] and Kreimer [20] studied single-channel RTS with two different kinds of activities. In [21] we computed analytically (for exponentially distributed service times) and via Cross Entropy [22]-[24] simulation approach (for generally distributed service times) optimal routing probabilities for RTS with ample maintenance facilities. Bassan and Kreimer [25] treated RTS with preemptive priorities.

The work presented here is a generalization of results obtained by Shimshi and Kreimer [19] and Kreimer [20] for RTS with single channel and two kinds of activities. It deals with multiserver RTS, providing the service to the requests of real-time jobs arriving via several channels. Servers are identical, but may be maintained/programmed for several (two or more) kinds of activities. We provide balance equations describing two models (with and without nonpreemptive priorities) operating under a maximum load regime and show how to compute various performance measures, when both operation and maintenance times are exponentially distributed.

The paper is organized as follows: In Section 2, the description of the model is presented. Section 3 provides balance equations for RTS with nonpreemptive priorities. Section 4 provides balance equations for RTS without priorities. Section 5 is devoted to computation of various performance characteristics. Finally, Section 6 contains conclusions.

2. Description of the Model

The most important characteristics of RTS with a zero deadline for the beginning of job processing are summarized in Kreimer and Mehrez [24] as follows:

1) Jobs/data acquisition and execution are as fast as the data arrival rate.
2) Jobs are executed immediately upon arrival, conditional on system availability. Between jobs arrival and their execution, delays are negligible.
3) That part of the job which is not executed in real time is lost forever. Storage of non-completed jobs or their parts is impossible. Thus, queues of jobs do not exist in such an RTS.

Nevertheless, some important elements of queueing theory can be successfully applied in analysis of these RTS.

We consider a multiserver multichannel RTS consisting of *N* identical servers (e.g. unmanned aerial vehicles-UAV, machine controllers, overhearing devices, medical monitoring devices, etc.) that provide service for the requests of real time jobs, arriving via *r* identical channels (e.g. surveillance regions, assembly lines, communication channels, hospital patients, etc.).

Each server can serve different channels, but not simultaneously. There is exactly one request in each channel at any instant (there are no additional job arrivals, while the channel is busy), and therefore one server at most is used (with others being on stand-by or in maintenance or providing the service to another channel) to process the job of this channel at any given time. Thus, the total number of jobs in the system is *r* (is equal to a number of channels). Different parts of the same job can be served by different servers. Any part of the job that is not processed immediately is lost.

It is assumed that there are ample identical maintenance teams available to repair (with repair times R_i being independent identically distributed random variables—i.i.d.r.v.) all *N* servers simultaneously, if needed. Thus, each broken server enters maintenance facilities without delay. A fixed server may be of one of *m* kinds (e.g. of activity or quality) with probabilities p_i, $i=1,\cdots,m$ respectively. These probabilities can be used sometimes as control parameters. Only after the maintenance is completed, the type of fixed server can be determined. The fixed server of *i*-th type is operative for a period of time S_i before requiring R_i hours of repair. Both S_i and R_i are independent exponentially distributed random values with parameters μ_i $(i=1,\cdots,m)$ and λ respectively. The duration R_i of maintenance does not depend on the server's type (neither before nor after the maintenance). After maintenance is completed, the fixed server will either be on stand-by or providing service to one of channels.

The system works under a maximum load (worst case) of nonstop data arrival, which is equivalent to the case of a unique job of infinite duration in each one of *r* channels. This kind of operation is typical in high performance data acquisition and control systems. Decisions based on the *nonstop* data flow arriving via all *r* channels must be made extremely fast—in real time. If, during some period of time of length *T*, there is no available server to serve one of the channels, it means that the part of the job of length *T* is lost forever.

Our purpose is to provide balance equations for steady-state probabilities of this system, its availability and other performance characteristics. We will consider two policies/models: 1) nonpreemptive priority; 2) random choice of the fixed server.

3. Model with Nonpreemptive Priority

First we will provide the list of various terms, which are used in our equations.

3.1. Equations

We suppose that servers of first activity type have the highest priority, servers of the second activity type are after them, and so on, finally, servers of the *m*-th activity type have the lowest priority (**Figure 1**). Operating server is not interrupted, independently of its priority. When the operating server must be repaired, the fixed server with highest priority takes its place.

We denote $\{(n_1,k_1),\cdots,(n_m,k_m)\}$ the state of the system, where n_i $(i=1,\cdots,m)$ is the total number of fixed servers of *i*-th activity type and k_i $(i=1,\cdots,m)$ is the number of operating (providing service) fixed servers of *i*-th activity type respectively, and $p\{(n_1,k_1),\cdots,(n_m,k_m)\}$ the corresponding steady state probability. It is clear that $n_1+\cdots+n_m \le N$, $n_i \ge k_i \ge 0, \sum_{i=1}^{m} k_i = \min\left[r, \sum_{i=1}^{m} n_i\right]$.

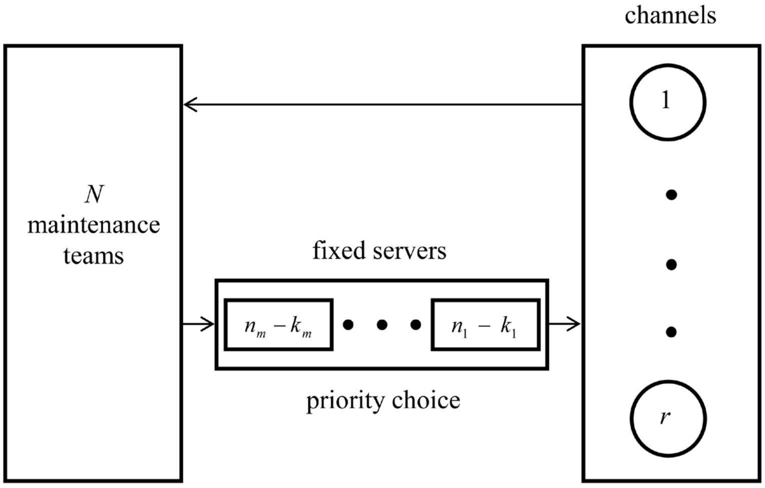

Figure 1. Model with priorities.

Theorem 1. Steady state probabilities for the model with nonpreemptive priorities can be found from the following system of balance equations:

$$\begin{aligned}
&\left[\sum_{i=1}^{m}k_i\mu_i+\left(N-\sum_{j=1}^{m}n_j\right)\lambda\right]p\left\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\right\}=\min\left(r-\sum_{i=1}^{m}k_i,1\right)\\
&\times\sum_{j=1}^{m}(k_j+1)\mu_j p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j+1,k_j+1),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}\\
&+\left(1-\min\left(r-\sum_{i=1}^{m}k_i,\ 1\right)\right)\sum_{j=1}^{m}\sum_{i=1}^{m}(k_j+1-\delta_{ij})\mu_j\left(1-\min\left(\sum_{s=1}^{i-1}(n_s-k_s),\ 1\right)\right)\\
&\times p\left\{(n_1,k_1),\cdots,(n_{i-1},k_{i-1}),(n_i+\delta_{ij},k_i-1+\delta_{ij}),(n_{i+1},k_{i+1}),\cdots,(n_{j-1},k_{j-1}),\right.\\
&\left.(n_j+1,k_j+1-\delta_{ij}),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\} \qquad (1)\\
&+\left(1-\min\left(\sum_{i=1}^{m}(n_i-k_i),1\right)\right)\sum_{j=1}^{m}\left(N-\sum_{i=1}^{m}n_i+1\right)\lambda p_j\\
&\times p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j-1,k_j-1),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}\\
&+\min\left(\sum_{i=1}^{m}(n_i-k_i),1\right)\sum_{j=1}^{m}\left(N-\sum_{i=1}^{m}n_i+1\right)\lambda p_j\\
&\times p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j-1,k_j),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\},
\end{aligned}$$

where $\left\{(n_1,k_1),\cdots,(n_m,k_m)\right\}$ is any feasible state of the system, $\delta_{ij}=\begin{cases}0, & \text{if } i\neq j\\ 1, & \text{if } i=j\end{cases}$.

And

$$\sum_{\{(n_1,k_1),\cdots,(n_m,k_m)\}\in E}p\left\{(n_1,k_1),\cdots,(n_m,k_m)\right\}=1 \qquad (2)$$

where $E=\left\{\left\{(n_1,k_1),\cdots,(n_m,k_m)\right\}:0\le k_i\le n_i,\ i=\overline{1,m},\ \sum_{i=1}^{m}n_i\le N,\ \sum_{i=1}^{m}k_i=\min\left(r,\sum_{j=1}^{m}n_j\right)\right\}$ is the set of all feasible states of the system.

Proof:

1) First we consider the case $r\ge N$. Then it is clear that $n_i=k_i$, $i=\overline{1,m}$, *i.e.* all fixed servers must be

operative and priorities do not work in this case (each server coming from maintenance immediately start to work independently of its priority class). We have therefore [26]

$$N\lambda\cdot p\{(0,0),(0,0),\cdots,(0,0)\}=\sum_{j=1}^{m}\mu_j p\{(0,0),\cdots,(0,0),(n_j=1,k_j=1),(0,0),\cdots,(0,0)\} \tag{3}$$

for the state $\{(0,0),\cdots,(0,0)\}$, *i.e.* $n_i=k_i=0$, $i=\overline{1,m}$;

$$\begin{aligned}&\left[\sum_{j=1}^{s}k_{i_j}\mu_{i_j}+\left(N-\sum_{l=1}^{s}k_{i_l}\right)\lambda\right]p\{(k_1,\ k_1),(k_2,\ k_2),\cdots,(k_m,k_m)\}\\&=\sum_{j=1}^{m}(k_j+1)\mu_j p\{(k_1,\ k_1),\cdots,(k_{j-1},\ k_{j-1}),(k_j+1,k_j+1),(k_{j+1},k_{j+1}),\cdots,(k_m,k_m)\}\\&\quad+\sum_{j=1}^{s}\left(N-\sum_{l=1}^{s}k_{i_l}+1\right)\lambda p_{i_j}*p\{(k_1,k_1),\cdots,(k_{i_j-1},k_{i_j-1}),(k_{i_j}-1,k_{i_j}-1),(k_{i_j+1},k_{i_j+1}),\cdots,(k_m,k_m)\},\end{aligned} \tag{4}$$

for the states $n_i=k_i$, $i=\overline{1,m}$, $k_{i_j}>0$, $j=\overline{1,s}$, $k_{i_j}=0$, $j=\overline{s+1,m}$, $0<s<m$, $\sum_{j=1}^{s}n_{i_j}=\sum_{j=1}^{s}k_{i_j}<N$;

$$\begin{aligned}&\left[\sum_{j=1}^{s}k_{i_j}\mu_{i_j}\right]p\{(k_1,k_1),(k_2,k_2),\cdots,(k_m,k_m)\}\\&=\sum_{j=1}^{s}\lambda p_{i_j}p\{(k_1,k_1),\cdots,(k_{i_j-1},k_{i_j-1}),(k_{i_j}-1,k_{i_j}-1),(k_{i_j+1},k_{i_j+1}),\cdots,(k_m,k_m)\},\end{aligned} \tag{5}$$

for the states $n_i=k_i$, $i=\overline{1,m}$, $k_{i_j}>0$, $j=\overline{1,s}$, $k_{i_j}=0$, $j=\overline{s+1,m}$, $0<s<m$, $\sum_{j=1}^{s}n_{i_j}=\sum_{j=1}^{s}k_{i_j}=N$;

$$\begin{aligned}&\left[\sum_{i=1}^{m}k_i\mu_i+\left(N-\sum_{j=1}^{m}k_j\right)\lambda\right]p\{(k_1,k_1),(k_2,k_2),\cdots,(k_m,k_m)\}\\&=\sum_{j=1}^{m}(k_j+1)\mu_j p\{(k_1,k_1),\cdots,(k_{j-1},k_{j-1}),(k_j+1,k_j+1),(k_{j+1},k_{j+1}),\cdots,(k_m,k_m)\}\\&\quad+\sum_{j=1}^{m}\left(N-\sum_{i=1}^{m}k_i+1\right)\lambda p_j*p\{(k_1,k_1),\cdots,(k_{j-1},k_{j-1}),(k_j-1,k_j-1),(k_{j+1},k_{j+1}),\cdots,(k_m,k_m)\},\end{aligned} \tag{6}$$

for the states $n_i=k_i>0$, $i=\overline{1,m}$, $\sum_{i=1}^{m}n_i=\sum_{i=1}^{m}k_i<N$;

$$\begin{aligned}&\left[\sum_{i=1}^{m}k_i\mu_i\right]p\{(k_1,k_1),(k_2,k_2),\cdots,(k_m,k_m)\}\\&=\sum_{j=1}^{m}\lambda p_j p\{(k_1,\ k_1),\cdots,(k_{j-1},k_{j-1}),(k_j-1,k_j-1),(k_{j+1},k_{j+1}),\cdots,(k_m,k_m)\},\end{aligned} \tag{7}$$

for the states $n_i=k_i>0$, $i=\overline{1,m}$, $\sum_{i=1}^{m}n_i=\sum_{i=1}^{m}k_i=N$;

Now Equation (1) follows from Equations (3)-(7).

2) Next, we consider the case $r<N$. Then we have:

Equation (3) again for the state $\{(0,0),\cdots,(0,0)\}$, *i.e.* $n_i=k_i=0$, $i=\overline{1,\ m}$;

Equation (4) for the states $n_i=k_i$, $i=\overline{1,m}$, $k_{i_j}>0$, $j=\overline{1,s}$, $k_{i_j}=0$, $j=\overline{s+1,m}$, $0<s<m$,

$\sum_{j=1}^{s} n_{i_j} = \sum_{j=1}^{s} k_{i_j} < r$;

Equation (6) for the states $n_i = k_i > 0$, $i = \overline{1,m}$, $\sum_{i=1}^{m} n_i = \sum_{i=1}^{m} k_i < r$;

$$\begin{aligned}
&\left[\sum_{j=1}^{s} k_{i_j}\mu_{i_j} + (N-r)\lambda\right] p\{(k_1, k_1),\ (k_2, k_2),\ \ldots,\ (k_m, k_m)\} \\
&= \sum_{j=1}^{s} k_{i_j}\mu_{i_j}\, p\left\{(k_1, k_1),\ \ldots,\ \left(k_{i_j-1}, k_{i_j-1}\right),\ \left(k_{i_j}+1, k_{i_j}\right),\ \left(k_{i_j+1}, k_{i_j+1}\right),\ \ldots,\ (k_m, k_m)\right\} \\
&\quad + \sum_{j=1}^{m} \sum_{\substack{l=1 \\ i_l \neq j}}^{s} (k_j+1)\mu_j\, p\left\{(k_1, k_1),\ \ldots,\ \left(k_{i_l-1}, k_{i_l-1}\right),\ \left(k_{i_l}, k_{i_l}-1\right),\ \left(k_{i_l+1}, k_{i_l+1}\right),\ \ldots,\ \left(k_{j-1}, k_{j-1}\right),\right. \\
&\quad \left.(k_j+1, k_j+1),\ \left(k_{j+1}, k_{j+1}\right),\ \ldots,\ (k_m, k_m)\right\} + \sum_{j=1}^{s} (N-r+1)\lambda p_{i_j} \\
&\quad \times p\left\{(k_1, k_1),\ \ldots,\ \left(k_{i_j-1}, k_{i_j-1}\right),\ \left(k_{i_j}-1, k_{i_j}-1\right),\ \left(k_{i_j+1}, k_{i_j+1}\right),\ \ldots,\ (k_m, k_m)\right\}
\end{aligned} \tag{8}$$

for the states $n_i = k_i$, $i = \overline{1,m}$, $k_{i_j} > 0$, $j = \overline{1,s}$, $k_{i_j} = 0$, $j = \overline{s+1,m}$, $0 < s < m$, $\sum_{j=1}^{s} n_{i_j} = \sum_{j=1}^{s} k_{i_j} = r$;

$$\begin{aligned}
&\left[\sum_{i=1}^{m} k_i\mu_i + (N-r)\lambda\right] p\{(k_1,k_1),(k_2,k_2),\cdots,(k_m,k_m)\} \\
&= \sum_{j=1}^{m}\sum_{i=1}^{m} \left(k_j+1-\delta_{ij}\right)\mu_j + \sum_{j=1}^{m} (N-r+1)\lambda p_j \\
&\quad * p\left\{(k_1,k_1),\cdots,\left(k_{j-1},k_{j-1}\right),\left(k_j-1,k_j-1\right),\left(k_{j+1},k_{j+1}\right),\cdots,(k_m,k_m)\right\},
\end{aligned} \tag{9}$$

for the states $n_i = k_i > 0$, $i = \overline{1,m}$, $\sum_{i=1}^{m} n_i = \sum_{i=1}^{m} k_i = r$;

$$\begin{aligned}
&\left[\sum_{i=1}^{m} k_i\mu_i + \left(N - \sum_{j=1}^{m} n_j\right)\lambda\right] p\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\} \\
&= \sum_{j=1}^{m} k_j\mu_j \left(1-\min\left(\sum_{s=1}^{j-1}(n_s-k_s),\ 1\right)\right) \\
&\quad * p\left\{(n_1,k_1),\cdots,\left(n_{j-1},k_{j-1}\right),\left(n_j+1,k_j\right),\left(n_{j+1},k_{j+1}\right),\cdots,(n_m,k_m)\right\} \\
&\quad + \sum_{j=1}^{m}\sum_{\substack{l=1 \\ i_l \neq j}}^{t} (k_j+1)\mu_j \left(1-\min\left(\sum_{s=1}^{i_l-1}(n_s-k_s),1\right)\right) \\
&\quad * p\left\{(n_1,k_1),\cdots,\left(n_{i_l-1},k_{i_l-1}\right),\left(n_{i_l},k_{i_l}-1\right),\left(n_{i_l+1},k_{i_l+1}\right),\cdots,\left(n_{j-1},k_{j-1}\right),\right. \\
&\quad \left.\left(n_j+1,k_j+1\right),\left(n_{j+1},k_{j+1}\right),\cdots,(n_m,k_m)\right\} + \sum_{j=1}^{s}\left(N-\sum_{i=1}^{m} n_i+1\right)\lambda p_{i_j} \\
&\quad * p\left\{(n_1,k_1),\cdots,\left(n_{i_j-1},k_{i_j-1}\right),\left(n_{i_j}-1,k_{i_j}\right),\left(n_{i_j+1},k_{i_j+1}\right),\cdots,(n_m,k_m)\right\},
\end{aligned} \tag{10}$$

for the states $k_{i_l} > 0$, $l = \overline{1,t}$, $k_{i_l} = 0$, $l = \overline{t+1,m}$, $0 < t < m$, $n_{i_j} > k_{i_j}$, $j = \overline{1,s}$, $n_{i_j} = k_{i_j}$, $j = \overline{s+1,m}$, $0 < s < m$, $\sum_{l=1}^{t} k_{i_l} = r$, $\sum_{i=1}^{m} n_i < N$;

$$\begin{aligned}
&\left[\sum_{i=1}^{m}k_i\mu_i+\left(N-\sum_{j=1}^{m}n_j\right)\lambda\right]p\left\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\right\}\\
&=\sum_{j=1}^{m}k_j\mu_j\left(1-\min\left(\sum_{s=1}^{j-1}(n_s-k_s),1\right)\right)\\
&\quad * p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j+1,k_j),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}\\
&\quad +\sum_{j=1}^{m}\sum_{\substack{i=1\\ i\neq j}}^{m}(k_j+1)\mu_j\left(1-\min\left(\sum_{s=1}^{i-1}(n_s-k_s),1\right)\right)\\
&\quad * p\left\{(n_1,\ k_1),\cdots,(n_{i-1},k_{i-1}),(n_i,k_i-1),(n_{i+1},k_{i+1}),\cdots,(n_{j-1},k_{j-1}),\right.\\
&\quad \left.(n_j+1,k_j+1),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}+\sum_{j=1}^{s}\left(N-\sum_{i=1}^{m}n_i+1\right)\lambda p_{i_j}\\
&\quad * p\left\{(n_1,\ k_1),\cdots,(n_{i_j-1},k_{i_j-1}),(n_{i_j}-1,k_{i_j}),(n_{i_j+1},k_{i_j+1}),\cdots,(n_m,k_m)\right\},
\end{aligned}\tag{11}$$

for the states $k_i>0$, $i=\overline{1,m}$, $n_{i_j}>k_{i_j}$, $j=\overline{1,s}$, $n_{i_j}=k_{i_j}$, $j=\overline{s+1,m}$, $0<s<m$, $\sum_{i=1}^{m}k_i=r$, $\sum_{i=1}^{m}n_i<N$;

$$\begin{aligned}
&\left[\sum_{i=1}^{m}k_i\mu_i+\left(N-\sum_{j=1}^{m}n_j\right)\lambda\right]p\left\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\right\}\\
&=\sum_{j=1}^{m}k_j\mu_j\left(1-\min\left(\sum_{s=1}^{j-1}(n_s-k_s),1\right)\right)\\
&\quad * p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j+1,k_j),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}\\
&\quad +\sum_{j=1}^{m}\sum_{\substack{l=1\\ i_l\neq j}}^{t}(k_j+1)\mu_j\left(1-\min\left(\sum_{s=1}^{i_l-1}(n_s-k_s),1\right)\right)\\
&\quad * p\left\{(n_1,k_1),\cdots,(n_{i_l-1},k_{i_l-1}),(n_{i_l},k_{i_l}-1),(n_{i_l+1},k_{i_l+1}),\cdots,(n_{j-1},k_{j-1}),\right.\\
&\quad \left.(n_j+1,k_j+1),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}+\sum_{j=1}^{m}\left(N-\sum_{i=1}^{m}n_i+1\right)\lambda p_j\\
&\quad * p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j-1,k_j),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\},
\end{aligned}\tag{12}$$

for the states $k_i<n_i$, $i=\overline{1,m}$, $k_{i_l}>0$, $l=\overline{1,t}$, $k_{i_l}=0$, $l=\overline{t+1,m}$, $0<t<m$, $\sum_{l=1}^{t}k_{i_l}=r$, $\sum_{i=1}^{m}n_i<N$;

$$\begin{aligned}
&\left[\sum_{i=1}^{m}k_i\mu_i+\left(N-\sum_{j=1}^{m}n_j\right)\lambda\right]p\left\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\right\}\\
&=\sum_{j=1}^{m}\sum_{i=1}^{m}(k_j+1-\delta_{ij})\mu_j\left(1-\min\left(\sum_{s=1}^{i-1}(n_s-k_s),1\right)\right)\\
&\quad * p\left\{(n_1,k_1),\cdots,(n_{i-1},k_{i-1}),(n_i+\delta_{ij},k_i-1+\delta_{ij}),(n_{i+1},k_{i+1}),\cdots,(n_{j-1},k_{j-1}),\right.\\
&\quad \left.(n_j+1,k_j+1-\delta_{ij}),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}+\sum_{j=1}^{m}\left(N-\sum_{i=1}^{m}n_i+1\right)\lambda p_j\\
&\quad * p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j-1,k_j),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\},
\end{aligned}\tag{13}$$

for the states $0 < k_i < n_i$, $i = \overline{1,m}$, $\sum_{i=1}^{m} k_i = r$, $\sum_{i=1}^{m} n_i < N$;

$$\left[\sum_{i=1}^{m} k_i \mu_i\right] p\left\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\right\} = \sum_{j=1}^{s} \lambda p_{i_j} p\left\{(n_1,k_1),\cdots,(n_{i_j-1},k_{i_j-1}),(n_{i_j}-1,k_{i_j}),(n_{i_j+1},k_{i_j+1}),\cdots,(n_m,k_m)\right\}, \tag{14}$$

for the states $k_i \ge 0$, $i = \overline{1,m}$, $n_{i_j} > k_{i_j}$, $j = \overline{1,s}$, $n_{i_j} = k_{i_j}$, $j = \overline{s+1,m}$, $0 < s < m$, $\sum_{i=1}^{m} k_i = r$, $\sum_{i=1}^{m} n_i = N$;

$$\left[\sum_{i=1}^{m} k_i \mu_i\right] p\left\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\right\} = \sum_{j=1}^{m} \lambda p_j p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j-1,k_j),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}, \tag{15}$$

for the states $0 \le k_i < n_i$, $i = \overline{1,m}$, $\sum_{i=1}^{m} k_i = r$, $\sum_{i=1}^{m} n_i = N$;

Now, Equation (1) follows from Equations (3), (4), (6), (8)-(15).

Q.E.D.

4. Model without Priorities

We suppose that when the operating server must go to maintenance, the server which takes its place is chosen among the fixed servers in a random way (**Figure 2**). Namely, if there are $n_i - k_i$ fixed servers of the *i*-th activity type $(i = 1,\cdots,m)$ on stand-by, then it is chosen for operation with probability $(n_i - k_i)\Big/\sum_{u=1}^{m}(n_u - k_u)$. Denoting the state of the system $\left\{(n_1,k_1),\cdots,(n_m,k_m)\right\}$ and the corresponding steady state probability $p\left\{(n_1,k_1),\cdots,(n_m,k_m)\right\}$ as in previous Section, we obtain

Theorem 2. Steady state probabilities for the model without priorities can be found from the following system of balance equations:

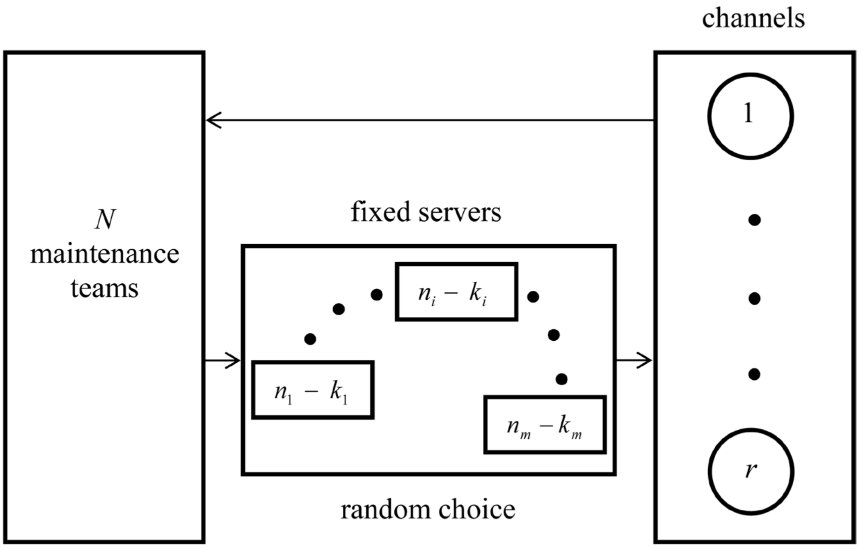

Figure 2. Model without priorities.

$$\begin{aligned}
&\left[\sum_{i=1}^{m} k_i\mu_i + \left(N-\sum_{j=1}^{m} n_j\right)\lambda\right] p\left\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\right\} = \min\left(r-\sum_{i=1}^{m} k_i,1\right) \\
&*\sum_{j=1}^{m}\left(k_j+1\right)\mu_j p\left\{(n_1,k_1),\cdots,(n_{j-1},\ k_{j-1}),(n_j+1,k_j+1),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\} \\
&+\left(1-\min\left(r-\sum_{i=1}^{m} k_i,1\right)\right)\sum_{j=1}^{m}\sum_{i=1}^{m}\left(k_j+1-\delta_{ij}\right)\mu_j \frac{n_i-k_i+1}{\sum_{s=1}^{m}(n_s-k_s)+1} * p\left\{(n_1,k_1),\cdots,(n_{i-1},k_{i-1}),\right. \\
&\left.(n_i+\delta_{ij},k_i-1+\delta_{ij}),(n_{i+1},k_{i+1}),\cdots,(n_{j-1},k_{j-1}),(n_j+1,k_j+1-\delta_{ij}),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\} \qquad (16) \\
&+\left(1-\min\left(\sum_{i=1}^{m}(n_i-k_i),1\right)\right) * \sum_{j=1}^{m}\left(N-\sum_{i=1}^{m} n_i+1\right)\lambda p_j * p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j-1,k_j-1),\right. \\
&\left.(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\} + \min\left(\sum_{i=1}^{m}(n_i-k_i),1\right)\sum_{j=1}^{m}\left(N-\sum_{i=1}^{m} n_i+1\right)\lambda p_j \\
&* p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j-1,k_j),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\},
\end{aligned}$$

where $\left\{(n_1,k_1),\cdots,(n_m,k_m)\right\}$ is any feasible state of the system, $\delta_{ij}=\begin{cases}0, & \text{if } i\neq j\\ 1, & \text{if } i=j\end{cases}$.

And

$$\sum_{\{(n_1,k_1),\cdots,(n_m,k_m)\}\in E} p\left\{(n_1,k_1),\cdots,(n_m,k_m)\right\} = 1 \qquad (17)$$

where $E=\left\{\left\{(n_1,k_1),\cdots,(n_m,k_m)\right\}: 0\le k_i\le n_i,\ i=\overline{1,m},\ \sum_{i=1}^{m} n_i\le N,\ \sum_{i=1}^{m} k_i=\min\left(r,\sum_{j=1}^{m} n_j\right)\right\}$ is the set of all feasible states of the system.

Proof:

1) First, we consider the case $r\ge N$. Then, it is clear that there are no fixed servers on stand-by, *i.e.* all fixed servers are operative. Each server coming back from maintenance immediately starts to work. We, therefore, have the same situation (and Equations (3)-(7)) as in Theorem 1 of the previous section.

2) Next, we consider the case $r<N$. Then equations of previous Section do not change for the following types of states:

Equation (3) for the state $\left\{(0,0),\cdots,(0,0)\right\}$, *i.e.* $n_i=k_i=0$, $i=\overline{1,m}$;

Equation (4) for the **s**tates $n_i=k_i$, $i=\overline{1,m}$, $k_{i_j}>0$, $j=\overline{1,s}$, $k_{i_j}=0$, $j=\overline{s+1,m}$, $0<s<m$, $\sum_{j=1}^{s} n_{i_j}=\sum_{j=1}^{s} k_{i_j}<r$;

Equation (6) for the states $n_i=k_i>0$, $i=\overline{1,m}$, $\sum_{i=1}^{m} n_i=\sum_{i=1}^{m} k_i<r$;

Equation (8) for the states $n_i=k_i$, $i=\overline{1,m}$, $k_{i_j}>0$, $j=\overline{1,s}$, $k_{i_j}=0$, $j=\overline{s+1,m}$, $0<s<m$, $\sum_{j=1}^{s} n_{i_j}=\sum_{j=1}^{s} k_{i_j}=r$;

Equation (9) for the states $n_i=k_i>0$, $i=\overline{1,m}$, $\sum_{i=1}^{m} n_i=\sum_{i=1}^{m} k_i=r$;

Equation (14) for the states $k_i\ge 0$, $i=\overline{1,m}$, $n_{i_j}>k_{i_j}$, $j=\overline{1,s}$, $n_{i_j}=k_{i_j}$, $j=\overline{s+1,m}$, $0<s<m$, $\sum_{i=1}^{m} k_i=r$, $\sum_{i=1}^{m} n_i=N$;

And Equation (15) for the states $0 \le k_i < n_i$, $i = \overline{1,m}$, $\sum_{i=1}^{m} k_i = r$, $\sum_{i=1}^{m} n_i = N$.

For the states below, we will have following new equations:

$$\begin{aligned}
&\left[\sum_{i=1}^{m} k_i\mu_i + \left(N - \sum_{j=1}^{m} n_j\right)\lambda\right] p\left\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\right\}\\
&= \sum_{j=1}^{m} k_j\mu_j \frac{n_j - k_j + 1}{\sum_{s=1}^{m}(n_s - k_s) + 1} \times p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j+1,k_j),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}\\
&+\sum_{j=1}^{m}\sum_{\substack{l=1\\ i_l \ne j}}^{t}(k_j+1)\mu_j \frac{n_{i_l} - k_{i_l} + 1}{\sum_{s=1}^{m}(n_s - k_s) + 1} \times p\left\{(n_1,k_1),\cdots,(n_{i_l-1},k_{i_l-1}),(n_{i_l},k_{i_l}-1),\right.\\
&\left.(n_{i_l+1},k_{i_l+1}),\cdots,(n_{j-1},k_{j-1}),(n_j+1,k_j+1),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}\\
&+\sum_{j=1}^{s}\left(N - \sum_{i=1}^{m} n_i + 1\right)\lambda p_{i_j} * p\left\{(n_1,k_1),\cdots,(n_{i_j-1},k_{i_j-1}),(n_{i_j}-1,k_{i_j}),(n_{i_j+1},k_{i_j+1}),\cdots,(n_m,k_m)\right\},
\end{aligned}\tag{18}$$

for the states $k_{i_l} > 0$, $l = \overline{1,t}$, $k_{i_l} = 0$, $l = \overline{t+1,m}$, $0 < t < m$, $n_{i_j} > k_{i_j}$, $j = \overline{1,s}$, $n_{i_j} = k_{i_j}$, $j = \overline{s+1,m}$, $0 < s < m$, $\sum_{l=1}^{t} k_{i_l} = r$, $\sum_{i=1}^{m} n_i < N$;

$$\begin{aligned}
&\left[\sum_{i=1}^{m} k_i\mu_i + \left(N - \sum_{j=1}^{m} n_j\right)\lambda\right] p\left\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\right\}\\
&= \sum_{j=1}^{m} k_j\mu_j \frac{n_j - k_j + 1}{\sum_{s=1}^{m}(n_s - k_s) + 1} \times p\left\{(n_1,\ k_1),\cdots,(n_{j-1},k_{j-1}),(n_j+1,k_j),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}\\
&+\sum_{j=1}^{m}\sum_{\substack{i=1\\ i \ne j}}^{m}(k_j+1)\mu_j \frac{n_i - k_i + 1}{\sum_{s=1}^{m}(n_s - k_s) + 1} \times p\left\{(n_1,k_1),\cdots,(n_{i-1},k_{i-1}),(n_i,k_i-1),\right.\\
&\left.(n_{i+1},k_{i+1}),\cdots,(n_{j-1},k_{j-1}),(n_j+1,k_j+1),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}\\
&+\sum_{j=1}^{s}\left(N - \sum_{i=1}^{m} n_i + 1\right)\lambda p_{i_j} * p\left\{(n_1,k_1),\cdots,(n_{i_j-1},k_{i_j-1}),(n_{i_j}-1,k_{i_j}),(n_{i_j+1},k_{i_j+1}),\cdots,(n_m,k_m)\right\},
\end{aligned}\tag{19}$$

for the states $k_i > 0$, $i = \overline{1,m}$, $n_{i_j} > k_{i_j}$, $j = \overline{1,s}$, $n_{i_j} = k_{i_j}$, $j = \overline{s+1,m}$, $0 < s < m$, $\sum_{i=1}^{m} k_i = r$, $\sum_{i=1}^{m} n_i < N$;

$$\begin{aligned}
&\left[\sum_{i=1}^{m} k_i\mu_i + \left(N - \sum_{j=1}^{m} n_j\right)\lambda\right] p\left\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\right\}\\
&= \sum_{j=1}^{m} k_j\mu_j \frac{n_j - k_j + 1}{\sum_{s=1}^{m}(n_s - k_s) + 1} \times p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j+1,k_j),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}\\
&+\sum_{j=1}^{m}\sum_{\substack{l=1\\ i_l \ne j}}^{t}(k_j+1)\mu_j \frac{n_{i_l} - k_{i_l} + 1}{\sum_{s=1}^{m}(n_s - k_s) + 1} \times p\left\{(n_1,k_1),\cdots,(n_{i_l-1},k_{i_l-1}),(n_{i_l},k_{i_l}-1),\right.\\
&\left.(n_{i_l+1},k_{i_l+1}),\cdots,(n_{j-1},k_{j-1}),(n_j+1,k_j+1),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\}\\
&+\sum_{j=1}^{m}\left(N - \sum_{i=1}^{m} n_i + 1\right)\lambda p_j * p\left\{(n_1,k_1),\cdots,(n_{j-1},k_{j-1}),(n_j-1,k_j),(n_{j+1},k_{j+1}),\cdots,(n_m,k_m)\right\},
\end{aligned}\tag{20}$$

for the states $k_i < n_i$, $i = \overline{1,m}$, $k_{i_l} > 0$, $l = \overline{1,t}$, $k_{i_l} = 0$, $l = \overline{t+1,m}$, $0 < t < m$, $\sum_{l=1}^{t} k_{i_l} = r$, $\sum_{i=1}^{m} n_i < N$;

$$\begin{aligned}
&\left[\sum_{i=1}^{m} k_i\mu_i + \left(N - \sum_{j=1}^{m} n_j\right)\lambda\right] p\left\{(n_1,k_1),(n_2,k_2),\cdots,(n_m,k_m)\right\} \\
&= \sum_{j=1}^{m}\sum_{i=1}^{m}\left(k_j + 1 - \delta_{ij}\right)\mu_j \frac{n_i - k_i + 1}{\sum_{s=1}^{m}(n_s - k_s) + 1} \\
&\times p\Big\{(n_1,k_1),\cdots,(n_{i-1},k_{i-1}),\left(n_i + \delta_{ij}, k_i - 1 + \delta_{ij}\right),(n_{i+1},k_{i+1}),\cdots,\left(n_{j-1},k_{j-1}\right), \\
&\left(n_j + 1, k_j + 1 - \delta_{ij}\right),\left(n_{j+1},k_{j+1}\right),\cdots,(n_m,k_m)\Big\} + \sum_{j=1}^{m}\left(N - \sum_{i=1}^{m} n_i + 1\right)\lambda p_j \\
&\times p\Big\{(n_1,k_1),\cdots,\left(n_{j-1},k_{j-1}\right),\left(n_j - 1, k_j\right),\left(n_{j+1},k_{j+1}\right),\cdots,(n_m,k_m)\Big\},
\end{aligned} \tag{21}$$

for the states $0 < k_i < n_i$, $i = \overline{1,m}$, $\sum_{i=1}^{m} k_i = r$, $\sum_{i=1}^{m} n_i < N$;

Now, Equations (16) follows from Equations (3), (4), (6), (8), (9), (14), (15), (18)-(21).

Q.E.D.

5. Performance Characteristics

In this section, we shall show how to compute some useful performance characteristics of the RTS under consideration.

Each server can be in one of following positions:

a) Busy (operating);

b) Out of order (in maintenance);

c) On stand-by;

Each channel can be in one of two positions:

a) In service;

b) Out of service.

Keeping in mind that, the state of the system is represented by the vector $\left\{(n_1,k_1),\cdots,(n_m,k_m)\right\}$, where n_i $(i = 1,\cdots,m)$ is the total number of fixed servers of *i*-th activity type and k_i $(i = 1,\cdots,m)$ is the number of operating fixed servers of *i*-th activity type, we can represent the number of servers and channels in different positions in terms of n_i and k_i, namely:

Number of fixed servers is $\sum_{i=1}^{m} n_i$;

Number of operating fixed servers (the number of channels in service) is $\sum_{i=1}^{m} k_i$;

Number of fixed servers of *i*-th activity type being on stand-by is $n_i - k_i$;

Number of all fixed servers being on stand-by is $\sum_{i=1}^{m}(n_i - k_i)$;

Number of broken servers is $N - \sum_{i=1}^{m} n_i$;

Number of channels out of service is $r - \sum_{i=1}^{m} k_i$;

Now we can obtain corresponding average values, using steady state probabilities, namely:

$L_i = E[n_i]$—average number of fixed servers of *i*-th activity type $\left(i = \overline{1,m}\right)$;

$L=\sum_{i=1}^{m}L_i$ —average number of all fixed servers;

$N-L$ —average number of broken servers (also the average number of servers in maintenance);

$E[k_i]$—average number of operating servers of *i*-th activity type $\left(i=\overline{1,m}\right)$;

$\sum_{i=1}^{m}E[k_i]$—average number of all operating servers (as well as the average number of channels in service);

$r-\sum_{i=1}^{m}E[k_i]$—average number of non-executed real-time jobs;

$L_{i,\,sb}=E[n_i-k_i]$—average number of *i*-th type fixed servers on stand-by $\left(i=\overline{1,m}\right)$;

$L_{sb}=\sum_{i=1}^{m}L_{i,\,sb}$ —average number of all fixed servers on stand-by;

$\lambda_{av}=\lambda(N-L)$—average rate of servers arriving from maintenance;

$\lambda_{av,\,i}=\lambda p_i(N-L)$—average rate of *i*-th type servers arriving from maintenance $\left(i=\overline{1,m}\right)$;

$W_{i,\,sb}=L_{i,\,sb}/\lambda_{av,\,i}$ —average time spent on stand-by by the *i*-th type server $\left(i=\overline{1,m}\right)$ -Little Theorem;

$W_i=W_{i,\,sb}+1/\mu_i$ —average time of the *i*-th type server being fixed $\left(i=\overline{1,m}\right)$ -Little Theorem.

The use of RTS relies on the principle of availability, which can be given in our case by the following formula $Av=\frac{1}{r}\sum_{i=1}^{m}E[k_i]$.

Another important performance measure is the average cost of system operation during time unit. Denote C_i $(i=1,\cdots,m)$ the cost of time unit activity of the *i*-th type server, and D the penalty paid for the time unit during which one of channels was not served. Then the average cost is defined as follows

$$TC=\sum_{i=1}^{m}C_iE[k_i]+D\left(r-\sum_{i=1}^{m}E[k_i]\right)=Dr-\sum_{i=1}^{m}(D-C_i)E[k_i]$$

6. Conclusions

In this paper, we provided equations for steady-state probabilities, as well as formulas for availability, average cost function and other performance characteristics of multiserver and multichannel RTS with different activity types and ample maintenance facilities. We have examined models with nonpreemptive priorities and without them. Practitioners and researchers can submit their parameters in equations for steady state probabilities and performance characteristics and to make their choice between two models in order to solve real life problems. Further research should address following problems:

- RTS with preemptive priorities;
- RTS in which each server needs more than one kind of maintenance;
- Optimization of RTS involving costs and penalties;
- Multilevel RTS;
- RTS with shortage of maintenance teams.

The use of queueing theory methodology will have significant benefits in analysis of these systems.

References

[1] Liu, C.L. and Layland, J.W. (1973) Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *The Journal of the ACM*, **20**, 46-61. http://dx.doi.org/10.1145/321738.321743

[2] Dhal, S.K. and Liu, C.L. (1978) On a Real-Time Scheduling Problem. *Operations Research*, **26**, 127-140. http://dx.doi.org/10.1287/opre.26.1.127

[3] Xie, M., Dai, Y.S., Poh, K.L. and Lai, C.D. (2004) Optimal Number of Hosts in a Distributed System Based on Cost Criteria. *International Journal of Systems Science*, **35**, 343-353. http://dx.doi.org/10.1080/00207720410001716228

[4] Tawegoum, R., Castelain, E. and Gentina, J.C. (1994) Real-Time Piloting of Flexible Manufacturing Systems. *Euro-*

pean Journal of Operational Research, **78**, 252-261. http://dx.doi.org/10.1016/0377-2217(94)90387-5

[5] Chakravarty, A.K. and Balakrishnan, N. (1998) Reacting in Real-Time to Production Contingencies in a Capacitated Flexible Cell. *European Journal of Operational Research*, **110**, 1-19. http://dx.doi.org/10.1016/S0377-2217(97)00212-9

[6] Glover, F.W. and Laguna, M. (1997) Tabu Search. Kluwer Academic Publishers, Dordrecht. http://dx.doi.org/10.1007/978-1-4615-6089-0

[7] Feo, T.A., Venkatraman, K. and Bard, J.F. (1991) A GRASP for a Difficult Single Scheduling Problem. *Computers and Operations Research*, **18**, 635-643. http://dx.doi.org/10.1016/0305-0548(91)90001-8

[8] Seguin, R., Potvin, J.-Y., Gendreau, M., Crainic, T.G. and Marcotte, P (1997) Real-Time Decision Problems: An Operational Research Perspective. *Journal of the Operational Research Society*, **48**, 162-174. http://dx.doi.org/10.1057/palgrave.jors.2600341

[9] Strosnider, J.K. and Paul, C.J. (1993) A Structured View of Real-Time Problem Solving. *AI Magazine*, **14**, 45-66.

[10] Kreimer, J. and Mehrez, A. (1993) An Optimal Operation Policy for Real-Time N-Server Stand-By System Involving Preventive Maintenance. *European Journal of Operational Research*, **69**, 50-54. http://dx.doi.org/10.1016/0377-2217(93)90089-6

[11] Kreimer, J. and Mehrez, A. (1994) Optimal Real-Time Data Acquisition and Processing by a Multiserver Stand-By System. *Operations Research*, **42**, 24-30. http://dx.doi.org/10.1287/opre.42.1.24

[12] Kreimer, J. and Mehrez, A. (1998) Computation of Availability of a Real-Time System Using Queueing Theory Methodology. *Journal of the Operational Research Society*, **49**, 1095-1100. http://dx.doi.org/10.1057/palgrave.jors.2600610

[13] Kreimer, J. (1999) Real-Time Multiserver and Multichannel Systems with Shortage of Maintenance Crews. *Mathematical and Computer Modelling*, **30**, 169-176. http://dx.doi.org/10.1016/S0895-7177(99)00206-X

[14] Gross, D. and Harris, C.M. (1998) Fundamentals of Queueing Theory. John Wiley, New York.

[15] Kreimer, J. (2000) Real-Time Multiserver System with Two Non-Identical Channels and Limited Maintenance Facilities. *Mathematics and Computers in Simulation*, **53**, 85-94. http://dx.doi.org/10.1016/S0378-4754(00)00171-3

[16] Ianovsky, E. and Kreimer, J. (2003) Optimization of Real-Time Multiserver System with Two Different Channels and Shortage of Maintenance Teams. *Mathematics and Computers in Simulation*, **63**, 615-627. http://dx.doi.org/10.1016/S0378-4754(03)00092-2

[17] Kreimer, J. (2002) Real-Time System with Homogeneous Servers and Nonidentical Channels in Steady State. *Computers and Operations Research*, **29**, 1465-1473. http://dx.doi.org/10.1016/S0305-0548(01)00042-9

[18] Kreimer, J. (2002) Effectiveness Analysis of Real-Time Data Acquisition and Processing Multichannel Systems. *IEEE Transactions on Reliability*, **51**, 91-99. http://dx.doi.org/10.1109/24.994922

[19] Shimshi, Y. and Kreimer, J. (2000) Real-Time Multiserver System with Different kinds of Activities. *Proceedings of the 11th Conference on Industrial Engineering and Management*, *IE&M'*2000, Beer-Sheva, May 2000, 17-21.

[20] Kreimer, J. (2003) Multiserver Single-Channel Real-Time System with Different Kinds of Activities. *Communications in Dependability and Quality Management*, **6**, 91-100.

[21] Ianovsky, E. and Kreimer, J. (2011) An Optimal Routing Policy for Unmanned Aerial Vehicles (Analytical and Cross-Entropy Simulation Approach). *Annals of Operations Research*, **189**, 215-253. http://dx.doi.org/10.1007/s10479-009-0609-1

[22] Rubinstein, R.Y. and Kroese, D.P. (2008) Simulation and the Monte Carlo Method. 2nd Edition, John Wiley & Sons, New York.

[23] Rubinstein, R.Y., Ridder, A. and Vaisman, R. (2014) Fast Sequential Monte Carlo Methods for Counting and Optimization. John Wiley & Sons, New York.

[24] Kroese, D.P., Taimre, T. and Botev, Z.I. (2011) Handbook of Monte Carlo Methods, John Wiley & Sons, New York. http://dx.doi.org/10.1002/9781118014967

[25] Bassan, E. and Kreimer, J. (2008) Multiserver and Multichannel Real-Time Systems with Separate Queues and Preemptive priorities. *Computer Modelling and New Technologies*, **42**, 7-15.

[26] Ianovsky, E. (2005) Analysis and Optimization of Real-Time Systems. PhD Thesis, Ben-Gurion University of the Negev, Beer-Sheva.

Notations

N	number of servers
r	number of channels
m	number of server's activity or quality kinds/types
S_i	i = th type server's operation/service time $\left(i=1,\cdots,m\right)$
μ_i	i = th type server's operation/service time (exp) parameter $\left(i=1,\cdots,m\right)$
R	intenance time
λ	server's maintenance time parameter (exp)
p_i	routing probabilities to i-th kind $\left(i=1,\cdots,m\right)$
n_i	total number of fixed servers of i-th activity type $\left(i=1,\cdots,m\right)$
k_i	number of operating fixed servers of i-th activity type $\left(i=1,\cdots,m\right)$
n_i-k_i	number of fixed servers of i-th activity type $\left(i=1,\cdots,m\right)$ on stand-by
$\left\{\left(n_1,k_1\right),\cdots,\left(n_m,k_m\right)\right\}$	the state of the system
$p\left\{\left(n_1,k_1\right),\cdots,\left(n_m,k_m\right)\right\}$	the corresponding steady state probability
C_i	the cost of time unit activity of the i-th type server $\left(i=1,\cdots,m\right)$
D	penalty cost for the time unit during which one of channels was not served
Av	system availability
TC	an average total cost of system operation per time unit