Annales Univ. Sci. Budapest., Sect. Comp. 18 (1999) 11-23

A QUEUEING MODEL FOR A NON-HOMOGENEOUS POLLING SYSTEM SUBJECT TO BREAKDOWNS

B. Almási (Debrecen, Hungary)

To the memory of Béla Kovács

Abstract. This paper deals with a non-homogeneous finite-source queuing model to describe the performance of a multi-terminal system subject to random breakdowns under Polling service discipline. The model studied here is a closed queueing network which has three service stations, a CPU (single server), terminals (infinite server), a repairman (single server) and finite number of customers (jobs) that have distinct service rates at the service stations. The service stations are not independent, as the repairman repairs the failed terminals and CPU. It can be viewed as a continuation of papers [1,2], which discussed a FIFO (First-In, First-Out) and a PPS serviced queuing model subject to random breakdowns. All random variables are assumed to be independent and exponentially distributed. The system's behaviour can be described by a Markov chain, but the number of states is very large. The purpose of this paper is to give a recursive computational approach to solve the steady-state equations and to illustrate the problem in question using some numerical results.

Acknowledgement. The author is very grateful to Professor M. Arató for his helpful comments.

Research is partially supported by Hungarian National Foundation for Scientific Research under grants OTKA T014974/95 and T016933/95.

1. The model

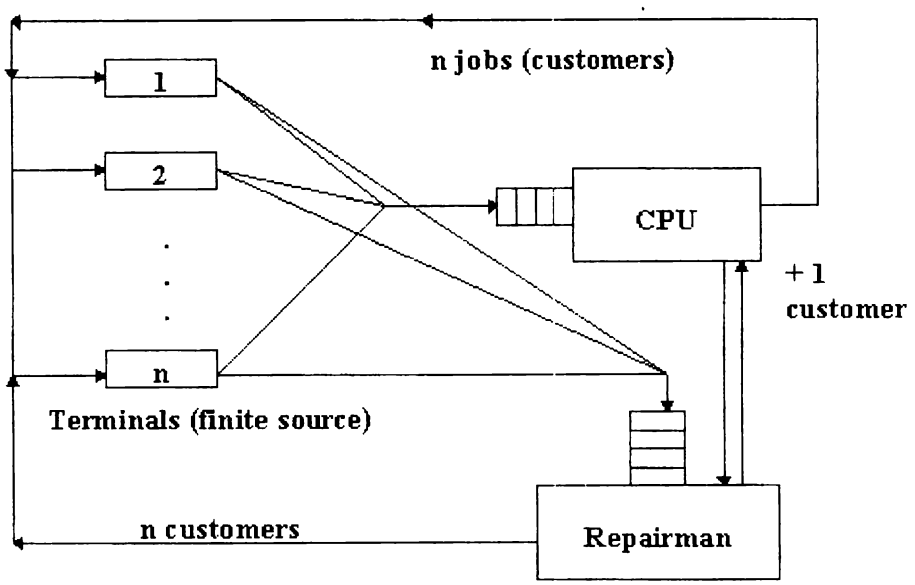

Figure 1.

This paper deals with a terminal system consisting of n terminals connected with a Central Processor Unit (CPU) and a repairman to repair the breakdowns of the CPU and terminals. The model is a closed queueing network model with three service stations and finite number $n+1$ of jobs (Figure 1). The service stations are not independent, as the repairman's queue consists of the failed CPU and terminals. The first service station is the processor, consisting of only one server, where jobs from the terminals may suffer from queueing delay. The required running times of job i are exponentially distributed random variables with means $1/\mu_i$ $(i = 1, \ldots, n)$. The processor serves the jobs one by one according to the job id numbers. There is no delay time when skipping from one job id to the next one (id 1 follows id n in the order). The serviced jobs are transferred to the second service station, which is a collection of n terminals (servers). At the terminals there is no queueing delay for the jobs (as the number of terminals is equal to the number of jobs). The service times of job i at the terminal i (i.e. the thinking times of the user at terminal i) are assumed to be exponentially distributed with means $1/\lambda_i$.

Let us suppose that the CPU is subject to random breakdowns stopping the whole system, and giving a special customer to the third service station

(repairman). The failure-free operation times of the CPU are exponentially distributed random variables with mean $1/\alpha$. The restoration times of the CPU are exponentially distributed with mean $1/\beta$.

The busy terminals are also subjects to random breakdowns not affecting the system's operation, but stopping the work at the terminal and transferring the customer to the repairman's que. The failure-free operation times of busy terminals are supposed to be exponentially distributed random variables with mean $1/\gamma_i$ for terminal i. The repair times of terminal i are exponentially distributed random variables with mean $1/\tau_i$. The breakdowns are serviced by a single repairman according to FIFO discipline among terminals and providing preemptive priority to the failure of CPU. We assume that each terminal sleeps while its job is serviced by the CPU, that is the terminal is inactive while waiting at the CPU, and it cannot break down. All random variables involved here are assumed to be independent of each other.

On the one hand this paper is a generalization of the non-homogeneous model discussed in [9] (which allowed only CPU failures), on the other hand it further generalizes the homogeneous model treated in [10] (which allowed both terminal and CPU failures). This paper is the continuation of [1,2] where the FIFO and PPS disciplines were discussed (instead of Polling) and we build a new non-homogeneous model and solve the steady-state equations recursively by using a similar computational approach as in [1]. In equilibrium the main performance of the system, such as the mean number of jobs residing at the CPU, the mean number of functional terminals, the expected response time of jobs and utilizations are obtained. Finally it is investigated - by using some numerical results - how breakdowns affect the performance characteristics.

2. The mathematical model and a computational approach

Let us introduce the following random variables:

$$X(t) = \begin{cases} 1, & \text{if the CPU is failed at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

$Y(t)$ = the failed terminals' indices at time t in order of their failure, or 0 if there is no failed terminal,

$Z(t)$ = the indices of jobs staying at the CPU at time t the first job id is serviced, the others are in increasing order, starting from the first one, (id 1 follows id n in the order), or $Z(t) = 0$ if there is no job at the CPU.

It is easy to see that the process

$$M(t) = (X(t), Y(t), Z(t)),$$

is a multi-dimensional Markov chain with 3 vector-components and with state space

$$S = (\quad (q; i_1, \ldots, i_k; j_1, \ldots, j_s), \quad q = 0, 1; \quad k = 0, \ldots, n; \quad s = 0, \ldots, n-k),$$

where

$(i_1, \ldots, i_k)$ is a permutation of k objects from the numbers $1, \ldots, n$ or 0, if k=0,

$(j_1, \ldots, j_s)$ is a cyclic permutation of the s element subsets from the remaining $n - k$ numbers or 0, if s=0.

The event $(q; i_1, \ldots, i_k; j_1, \ldots, j_s)$ denotes that the operating system is in state $X(t) = q$, there are k failed terminals with indices $i_1, \ldots, i_k$, and there are s jobs with indices $j_1, \ldots, j_s$ at the CPU $(i_r \neq j_l, \quad r = 1, \ldots, k; \quad l = 1, \ldots, s)$.

The reader can easily verify that the number of states is

$$dim(S) = 2 \sum_{k=0}^{n} \sum_{s=0}^{k} \frac{n!}{(n-k)!(s-1)!}.$$

Let us denote the steady-state distribution of $(M(t), t \geq 0)$ by

$$p(q; i_1, \ldots, i_k; j_1, \ldots, j_s) = \lim_{t \to \infty} p(X(t) = q; Y(t) = i_1, \ldots, i_k; Z(t) = j_1, \ldots, j_s),$$

which exists and is unique (see [5]) because of all the rates are assumed to be positive.

For brevity let us introduce the following notation:

$$R(i_1, \ldots, i_k; j_1, \ldots, j_s) =$$

$$= \begin{cases} \{r, j_s < r < j_1 \quad \text{and} \quad r \neq i_1, \ldots, i_k\} & \text{if } j_s < j_1, \\ \{r, (j_s < r \leq n \quad \text{or} \quad 1 \leq r < j_1) \quad \text{and} \quad r \neq i_1, \ldots, i_k\} & \text{otherwise.} \end{cases}$$

$R(i_1, \ldots, i_k; j_1, \ldots, j_s)$ denotes the set of r positive integers, for which

$$\text{if} \quad (0; i_1, \ldots, i_k; j_1, \ldots, j_s) \in S, \quad \text{then} \quad (0; i_1, \ldots, i_k; r, j_1, \ldots, j_s) \in S.$$

Since we study the steady-state behavior of the Markov chain $M(t)$, following [5], we can start with the statement

$$\text{Average rate of leaving state } (q; i_1, \ldots, i_k; j_1, \ldots, j_s) =$$
$$= \text{Average rate of entering state} (q; i_1, \ldots, i_k; j_1, \ldots, j_s),$$

that is we can build the global balance equations for $p(q; i_1, \ldots, i_k; j_1, \ldots, j_s)$ by using the rules discussed in Section 1:

$$(\alpha + \tau_{i_1} + \mu_{j_1} + \sum_{\substack{r \neq i_1, \ldots, i_k \\ r \neq j_1, \ldots, j_s}} (\lambda_r + \gamma_r)) p(0; i_1, \ldots, i_k; j_1, \ldots, j_s) =$$
$$= \beta p(1; i_1, \ldots, i_k; j_1, \ldots, j_s) + + \sum_{\substack{r \neq i_1, \ldots, i_k \\ r \neq j_1, \ldots, j_s}} \tau_r p(0; r, i_1, \ldots, i_k; j_1, \ldots, j_s) +$$
$$+ \sum_{r \in R(i_1, \ldots, i_k; j_1, \ldots, j_s)} \mu_r p(0; i_1, \ldots, i_k; r, j_1, \ldots, j_s) +$$
$$+ \gamma_{i_k} p(0; i_1, \ldots, i_{k-1}; j_1, \ldots, j_s) + \sum_{r=2}^{s} \lambda_{j_r} p(0; i_1, \ldots, i_k; j_1, \ldots j_{r-1}, j_{r+1}, \ldots, j_s),$$

for all $i_1, \ldots, i_k$; $j_1, \ldots, j_s$; $k = 0, \ldots, n$; $s = 0, \ldots, n - k$,

$$\beta p(1; i_1, \ldots, i_k; j_1, \ldots, j_s) = \alpha p(0; i_1, \ldots, i_k; j_1, \ldots, j_s), \tag{2}$$

for all $i_1, \ldots, i_k$; $j_1, \ldots, j_s$; $k = 0, \ldots, n$; $s = 0, \ldots, n-k$, where the probabilities of meaningless events and coefficients are defined to be zero.

The system of equations will be simpler if we substitute Equation (2) to Equation (1). Namely, we have

$$(\tau_{i_1} + \mu_{j_1} + \sum_{r \neq i_1, \ldots, i_k, j_1, \ldots, j_s} (\lambda_r + \gamma_r)) p(0; i_1, \ldots, i_k; j_1, \ldots, j_s) =$$
$$= \sum_{\substack{r \neq i_1, \ldots, i_k \\ r \neq j_1, \ldots, j_s}} (\tau_r p(0; r, i_1, \ldots, i_k; j_1, \ldots, j_s) +$$
$$+ \sum_{r \in R(i_1, \ldots, i_k; j_1, \ldots, j_s)} \mu_r p(0; i_1, \ldots, i_k; r, j_1, \ldots, j_s) +$$
$$+ \gamma_{i_k} p(0; i_1, \ldots, i_{k-1}; j_1, \ldots, j_s) + \sum_{r=2}^{s} \lambda_{j_r} p(0; i_1, \ldots, i_k; j_1, \ldots j_{r-1}, j_{r+1}, \ldots, j_s),$$

for all $i_1, \ldots, i_k$; $j_1, \ldots, j_s$; $k = 0, \ldots, n$; $s = 0, \ldots, n-k$,

$$\beta p(1; i_1, \ldots, i_k; j_1, \ldots, j_s) = \alpha p(0; i_1, \ldots, i_k; j_1, \ldots, j_s), \tag{4}$$

for all $i_1, \ldots, i_k$; $j_1, \ldots, j_s$; $k = 0, \ldots, n$; $s = 0, \ldots, n-k$.

The purpose of this paper is to solve this system subject to the normalization condition

$$\sum_{q=0}^{1} \sum_{k=0}^{n} \sum_{s=0}^{n-k} p(q, k, s) = 1,$$

where

$$p(q, k, s) = \sum_{(i_1, \ldots, i_k) \in V_n^k} \sum_{(j_1, \ldots, j_s) \in C_{n-k}^s} p(q; i_1, \ldots, i_k; j_1, \ldots, j_s),$$

V_n^k : the set of all $(i_1, \ldots, i_k)$ (as defined above),

C_{n-k}^s : the set of all $(j_1, \ldots, j_s)$ (as defined above).

Such a system of linear equations could easily be solved by standard computational methods, e.g. by Gauss-elimination. But it we do not forget that the unknowns are probabilities and therefore - since the state space is very large - the round off errors may have considerable effect on them (see [6,7,11]) and when using computer programs to solve the system of equations, the whole matrix of the equations cannot be stored in a personal computer if $n \geq 3$. It is more efficient to use a recursive computational method to determine the steady-state probabilities, as described in the following section (as it was proposed by Tomkó [4]).

3. The recursive solution

Let $\underline{Y}(m)$ be the vector of the stationary probabilities for the states where the operating system is working, there are k failed terminals, and $s = m - k$

jobs are waiting at the CPU ($(k = 0, ..., m)$, $m = 0, ..., n$). That is

$$\underline{Y}(m) = \begin{pmatrix} p(0;1,\ldots,m-1,m;0) \\ p(0;1,\ldots,m-1,m+1;0) \\ \\ p(0;n,\ldots,n-m+1;0) \\ p(0;1,\ldots,m-1;m) \\ p(0;1,\ldots,m-1;m+1) \\ \\ p(0;n,\ldots,n-m+2;n-m+1) \\ \\ p(0;0;n-m+1,\ldots,n) \end{pmatrix}$$

In words, the elements of $\underline{Y}(m)$ are written in lexicographically increasing order of indices

1.for $k = m$ and $s = 0$,

2.for $k = m - 1$ and $s = 1$,

$m + 1$. for $k = 0$ and $s = m$.

Similarly, let $\underline{Z}(m)$ be the vector of stationary probabilities alike $\underline{Y}(m)$, but for the states where the CPU is failed. From the definition it is obvious that the dimension of $\underline{Y}(m)$ and $\underline{Z}(m)$ is $d(m) = \sum_{s=0}^{m} \frac{n!}{(n-m)!(m-s-1)!}$. Using these notations Equations (3),(4) can be written in matrix form as

(i) $\underline{Y}(0) = C(0)\underline{Y}(1)$,

(ii) $\underline{Y}(j) = C(j)\underline{Y}(j+1) + D(j)\underline{Y}(j-1), \quad j = 1, \ldots, n-1$,

(iii) $\underline{Y}(n) = D(n)\underline{Y}(n-1)$,

(iv) $\underline{Z}(j) = F(j)\underline{Y}(j), \quad j = 0, \ldots, n$.

The dimensions of the matrices are $\left(d(j) = \sum_{s=0}^{j} \frac{n!}{(n-j)!(j-s-1)!}\right)$:

$$F(j) : d(j) \times d(j), \quad C(j) : d(j) \times d(j+1), \quad D(j) : d(j) \times d(j-1).$$

The elements of all the matrices can be obtained from the Equations (3), (4). For example we can use Equation (4) to obtain the elements of matrix $F(k+s)$ $(k+s = 0, \ldots, n)$: the element $p(1; i_1, \ldots, i_k; j_1, \ldots, j_s)$ of $Z(k+s)$ can be obtained from the element $p(0; i_1, \ldots, i_k; j_1, \ldots, j_s)$ of $Y(k+s)$ by multiplying it with $\frac{\alpha}{\beta}$. That is, the matrix $F(k+s)$ contains non-zero elements only in its main diagonal, and this non-zero element is the constant value $\frac{\alpha}{\beta}$.

Similarly, we can use the second line of Equation (3) to build the matrix $C(k+s)$, and the third line to determine the matrix $D(k+s)$ $(k+s = 0, \ldots, n)$. Applying these notations we can state our main result:

Theorem. *The solution of the Equations (i) - (iv) can be given in the form*

$$\begin{aligned} \underline{Y}(j) &= L(j)\underline{Y}(j-1), \qquad j = 1, \ldots, n, \\ \underline{Z}(j) &= F(j)\underline{Y}(j), \qquad\quad j = 0, \ldots, n, \end{aligned} \tag{5}$$

where $L(n) = D(n)$, $L(j) = (I - C(j)L(j+1))^{-1}D(j)$, $j = 1, \ldots, n-1$, *so the system of equations can be solved uniquely up to a multiplicative constant, which can be obtained from the normalization condition.*

The proof of the theorem is the same as it was in [1], only the contents of the matrices are different in our case.

Applying (5) we can start the recursion with any initial value denoted by $\underline{Y}'(0)$ and the non-normalized $p'(q; i_1, \ldots, i_k; j_1, \ldots, j_s)$ elements of $\underline{Y}'(m)$, $\underline{Z}'(m)$ $(m = 0, \ldots, n)$ can be obtained. We can calculate the steady-state probabilities from $\underline{Y}'(m)$, $\underline{Z}'(m)$ $(m = 0, \ldots, n)$ by using the normalization condition as follows

$$\underline{Y}(m) = \frac{\underline{Y}'(0)}{\sum_{q=0}^{1} \sum_{k=0}^{n} \sum_{s=0}^{n-k} \sum_{i_1, \ldots, i_k \in V_n^k} \sum_{j_1, \ldots, j_s \in C_{n-k}^s} p'(q; i_1, \ldots, i_k; j_1, \ldots, j_s)} \underline{Y}'(m),$$

$$\underline{Z}(m) =$$

$$= \frac{\underline{Y}'(0)}{\sum_{q=0}^{1} \sum_{k=0}^{n} \sum_{s=0}^{n-k} \sum_{i_1,\ldots,i_k \in V_n^k} \sum_{j_1,\ldots,j_s \in C_{n-k}^s} p'(q; i_1, \ldots, i_k; j_1, \ldots, j_s)} \underline{Z}'(m),$$

$m = 0, \ldots, n.$

4. Performace measures

We derive the steady-state characteristics from the steady-state probabilities because the model is too complicated to derive the characteristics directly from the parameters $(n, \alpha, \beta, \ldots)$. Some of these characteristics will be calculated in Tables 1-3 (for $n = 4$ and $n = 3$) as examples. We can use these numerical results to investigate how parameters influence the characteristics.

We employ the following usual notation: $\delta(i,j) = 1$, if $i = j$ (and 0 otherwise).

The steady-state characteristics:

(i) Mean number of jobs residing at the CPU

$$\overline{n}_j = \sum_{i=0}^{1} \sum_{k=0}^{n} \sum_{s=0}^{n-k} sp(i,k,s).$$

(ii) Mean number of functional terminals

$$\overline{n}_g = n - \sum_{i=0}^{1} \sum_{k=0}^{n} \sum_{s=0}^{n-k} kp(i,k,s).$$

(iii) Mean number of busy terminals

$$\overline{n}_b = \sum_{k=0}^{n} \sum_{s=0}^{n-k} (n-k-s)p(0,k,s).$$

(iv) Utilization of repairman

$$U_r = \sum_{k=0}^{n} \sum_{s=0}^{n-k} p(1,k,s) + \sum_{k=1}^{n} \sum_{s=0}^{n-k} p(0,k,s).$$

(v) Utilization of CPU

$$U_{CPU} = \sum_{k=0}^{n-1} \sum_{s=1}^{n-k} p(0,k,s).$$

(vi) Utilization of terminal i $(i = 1, \ldots, n)$

$$U_i = \sum_{k=0}^{n} \sum_{s=0}^{n-k} \sum_{r=1}^{k} \sum_{v=1}^{s} \sum_{i_1,\ldots,i_k \in V_n^k} \sum_{j_1,\ldots,j_s \in C_{n-k}^s} (1 - \delta(i,i_r) - \delta(i,j_v)) \times$$

$$\times p(0; i_1, \ldots, i_k; j_1, \ldots, j_s).$$

(vii) Expected response time of jobs for terminal i $(i = 1, \ldots, n)$

$$T_i = \frac{\sum_{q=0}^{1} \sum_{k=0}^{n} \sum_{s=0}^{n-k} \sum_{r=1}^{s} \sum_{i_1,\ldots,i_k \in V_n^k} \sum_{j_1,\ldots,j_s \in C_{n-k}^s} \delta(i,j_r) p(q; i_1, \ldots, i_k; j_1, \ldots, j_s)}{\lambda_i U_i}.$$

The reader can easily verify the validity of the formulas above. The proof of the last one can be found in [3].

5. Numerical results

The algorithm described above was implemented on an IBM PC. We show several examples to illustrate how breakdowns influence the characteristics. The running times were at about 8 seconds for $n = 4$. If we compare these results to the ones described in [1,2] we can see how scheduling strategy influences the characteristics.

Case 1. Failure-free system.

System parameters and characteristics:

$$\begin{array}{lll} n = 4 & \alpha = 0.0001 & \beta = 9999.0 \\ \overline{n}_j = 2.185 & \overline{n}_g = 4.0 & U_{CPU} = 0.903 \end{array}$$

Terminal parameters and characteristics:

i	λ_i	μ_i	γ_i	τ_i	U_i	T_i
1	0.300	0.600	0.0001	9999.0	0.471	3.741
2	0.400	0.700	0.0001	9999.0	0.415	3.526
3	0.200	0.500	0.0001	9999.0	0.552	4.055
4	0.500	0.900	0.0001	9999.0	0.377	3.303

Table 1.

This case will be the starting point of our investigation. It can be used to approximate a failure-free system described in [1] using FIFO discipline instead of polling. The global system characteristics are very similar to the results of [1]. The response times of the terminals are smaller for $i = 1, 2$; and bigger, for $i = 3, 4$.

Case 2. Terminal failure.

System parameters and characteristics:

$$\begin{array}{lll} n = 4 & \alpha = 0.0001 & \beta = 9999.0 \\ \overline{n}_j = 1.254 & \overline{n}_g = 2.58 & U_{CPU} = 0.663 \end{array}$$

Terminal parameters and characteristics:

i	λ_i	μ_i	γ_i	τ_i	U_i	T_i
1	0.300	0.600	0.2200	0.5000	0.342	2.993
2	0.400	0.700	0.1700	0.3400	0.331	2.685
3	0.200	0.500	0.1600	0.3000	0.391	3.303
4	0.500	0.900	0.3200	0.4500	0.265	2.516

Table 2.

In this example we can see how terminal failures influence the performance measures. The response times and the number of good terminals are less than in Case 1. That is, the system works as if there were less terminals. The effect

of the terminal failures seems to be greater using FIFO discipline at the CPU (see [1]): the terminal utilizations are smaller, the response times are greater in our case than it was in [1].

Case 3. CPU failure.

System parameters and characteristics:

$$\begin{array}{lll} n = 4 & \alpha = 0.25 & \beta = 0.45 \\ \overline{n}_j = 1.254 & \overline{n}_g = 2.58 & U_{CPU} = 0.663 \end{array}$$

Terminal parameters and characteristics:

i	λ_i	μ_i	γ_i	τ_i	U_i	T_i
1	0.300	0.600	0.2200	0.5000	0.220	4.656
2	0.400	0.700	0.1700	0.3400	0.213	4.177
3	0.200	0.500	0.1600	0.3000	0.251	5.138
4	0.500	0.900	0.3200	0.4500	0.171	3.915

Table 3.

If we compare these results with Case 1, it can be seen, that the failure of the CPU increases the response times and decreases the utilizations, as one can expect.

References

[1] **Almási B. and Sztrik J.,** A queueing model for a non-homogeneous terminal system subject to breakdowns, *Computers and Mathematics with Applications,* **25** (4) (1993), 105-111.

[2] **Almási B.,** A queueing model for a processor-shared multi-terminal system subject to breakdowns, *Acta Cybernetica,* **10** (4) (1993), 273-282.

[3] **Almási B.,** Response time for finite heterogeneous nonreliable queueing systems, *Computers and Mathematics with Applications,* **31** (11) (1996), 55-59.

[4] **Csige L. and Tomkó J.,** The machine interference for exponentially distributed operating and repair times, *Alk. Mat. Lapok,* **8** (1982), 107-124. (in Hungarian)

[5] **Goodman R.,** *Introduction to stochastic models,* Benjamin/Cummings, California, 1988.

[6] *Computer performance modeling handbook,* ed. S.S.Lavenberg, Academic Press, New York, 1983.

[7] **Snyder P.M. and Stewart W.J.,** Explicit and iterative approaches to solving queueing models, *Oper. Res.,* **33** (1985), 183-202.

[8] **Sztrik J.,** A probability model for priority processor-shared multiprogrammed computer systerns, *Acta Cybernetica,* **7** (1986), 329-340.

[9] **Sztrik J.,** On the heterogeneous machine interference with limited server's availability, *European Journal of Op. Res.,* **28** (1987), 321-328.

[10] **Sztrik J. and Gál T.,** A recursive solution of a queueing model for a multi-terminal system subject to breakdowns, *Performance Evaluation,* **11** (1990), 1-7.

[11] **Tijms H.C.,** *Stochastical modelling and analysis. A computational approach,* J.Wiley and Sons, New York, 1986.

B. Almási
Department of Mathematics and Informatics
Kossuth Lajos University
P.O.B. 12.
H-4010 Debrecen, Hungary