A NUMERICAL SOLUTION FOR THE MULTI-SERVER QUEUE WITH HYPER-EXPONENTIAL SERVICE TIMES

Jos H.A. de SMIT

Twente University of Technology, Enschede, The Netherlands

In this paper we present a numerical method for the queue $GI/H_2/s$, which is based on general results for $GI/H_m/s$. We give a complete description of the algorithm which yields exact results for the steady distributions of the actual waiting time, the virtual waiting time and the number of customers in the system both at arrival epochs and in continuous time.

Multi-server queue, hyper-exponential service times, actual waiting time, virtual waiting time, queue length

1. Introduction

In two recent papers, [1] and [2], we have obtained a general solution for the multi-server queue with hyper-exponential service times $GI/H_m/s$. Here we describe a numerical method based on this general solution. We restrict ourselves to the model $GI/H_2/s$, i.e. the service time distribution G is given by

$$G(x) = p_1(1 - \exp(-b_1 x)) + p_2(1 - \exp(-b_2 x)), \quad x \geqslant 0, \tag{1}$$

with $0 < p_1 < 1$, $p_2 = 1 - p_1$, $b_1 > b_2 > 0$. Let S be a typical service time, denote its mean by $\beta = ES$ and its coefficient of variation by $c_S^2 = \text{Var } S/\beta^2$. Then $\beta = p_1/b_1 + p_2/b_2$ and $c_S^2 > 1$. If we fix β and c_S^2 we can still choose a third parameter r_S which is defined by $r_S = (p_1/b_1)/(p_2/b_2)$.

For many practical situations, in which service times occur with a coefficient of variation larger than 1, the H_2 distribution may offer a satisfactory description. In what follows we shall give a method which yields the steady state distributions of the following quantities:
- the actual waiting time W;
- the virtual waiting time W^*;
- the queue length at arrival epochs (i.e. just before arrivals) Q;
- the queue length in continuous time Q^*;
- the total number of customers in the system at arrival epochs C;
- the total number of customers in the system in continuous time C^*.

No exact results for the complete distributions of W and W^* have been found before. For the special case of Poisson arrivals the moments of W^* and the distributions of Q^* and C^* (which in this case are identical to those of W, Q and C) can be calculated using the method of Takahashi and Takami [4]. In principle this method can be extended to the case of phase-type interarrival times, but in practice this may be difficult since such an extension leads to a very large state space.

For a lucid discussion and typical results of existing exact and approximate numerical methods the reader is referred to van Hoorn [5]. Our method is independent of the form of the interarrival time distribution. Any distribution can be included by changing only a single line in the computer program while the CPU-time needed is hardly effected. For those cases and those results which can be obtained by other methods it seems that our method is more efficient, more accurate and more stable. This may be explained by the fact that we take advantage of our knowledge of the form of the resulting distributions. In

Section 2 we give the relevant results for the queue $GI/H_2/s$, while Section 3 contains some more details about the numerical algorithm. As an illustration we give some typical numerical results for different interarrival time distributions.

2. The queue $GI/H_2/s$

In this section we give results for the queue $GI/H_2/s$, $s > 1$. The proofs can be found in [1] and [2], where these results have been obtained for the more general model $GI/H_m/s$, $s > 1$, $m > 1$. Let T_n be the arrival epoch of the nth customer, $n = 1, 2,\dots$. The interarrival times $A_n = T_n - T_{n-1}$, are i.i.d. with common distribution function F and finite mean α. The coefficient of variation of F is denoted by $c_A^2 = \text{Var } A_n/\alpha^2$ and its Laplace–Stieltjes transform by

$$A(\phi) = E(\exp(-\phi A_n)) = \int_0^\infty \exp(-\phi x)\, dF(x), \quad \text{Re } \phi \geqslant 0.$$

The service times S_n, $n = 1, 2,\dots$, are i.i.d. with common distribution function G, as given by (1). We can interpret the hyper-exponential service time distribution by saying that there are two types of customers; with probability p_i an arriving customer is of type i and requires a service time which is exponentially distributed with parameter b_i, $i = 1, 2$. This interpretation will be used in what follows.

Let W_n be the waiting time of the nth customer and $W_{n,i}$, $i = 1, 2,\dots,s$; the service backlog or the workload of the ith server just before the arrival of the nth customer, i.e. the ith server would become idle at $T_n + W_{n,i}$ (or is already idle at T_n if $W_{n,i} = 0$) if the nth and subsequent customers would not be admitted to the system. Since the queue discipline is first-come-first-served, we may assume that in front of each server there is a queue and that an arriving customer joins the queue of the server with the smallest workload. If there are several servers with this smallest workload the customer selects one of them at random. Hence $W_n = \min_i W_{n,i}$. We say that at $T_n -$ (i.e. just before the arrival of the nth customer) server i is in phase j iff $W_{n,i} > W_n$ and the last customer who joined the queue of server i before T_n is of type j. Let $X_{n,j}$ be the number of servers who at $T_n -$ are in phase j, $j = 1, 2$; then $X_{n,1} + X_{n,2} \leqslant s - 1$. In fact if $W_n > 0$ then $X_{n,1} + X_{n,2} = s - 1$ with probability 1. We have shown in [1] that $\{(W_n, X_{n,1}, X_{n,2}), n = 1, 2,\dots\}$ is a vector Markov process and $(W_n, X_{n,1}, X_{n,2})$ converges weakly to the steady state vector (W, X_1, X_2) iff $\rho = \beta/(\alpha s) < 1$. From now on we assume that this is the case.

In what follows we shall impose five conditions. We have shown that these conditions are 'almost always' satisfied, implying that any model for which they do not hold can be approximated arbitrarily closely by a model for which they do. Let $1(A)$ be the indicator function of the event A, i.e. $1(A) = 1$, if A occurs and $1(A) = 0$, otherwise; and let δ_{ij} be Kronecker's symbol, i.e. $\delta_{ii} = 1$ and $\delta_{ij} = 0$, $i \neq j$. Trinomial coefficients will be denoted by

$$\binom{i}{j;\,k} = \frac{i!}{j!\,k!\,(i-j-k)!}, \quad j \geqslant 0,\; k \geqslant 0,\; j + k \leqslant i.$$

The identity matrix of appropriate dimension is denoted by I.

The distribution of (W, X_1, X_2) can be specified by the following functions:

$$\begin{aligned} Z(i,j) &= P(X_1 = i, X_2 = j), \quad 1 \geqslant 0,\; j \geqslant 0,\; i + j \leqslant s - 2; \\ &= 0, \qquad\qquad\qquad\quad \text{otherwise;} \end{aligned}$$

$$Z_k(\phi) = E(\exp(-\phi W)1(X_1 = k - 1, X_2 = s - k)), \quad k = 1, 2,\dots,s;\; \text{Re } \phi \geqslant 0.$$

These functions satisfy the system of equations given below:

$$H(\phi)Z(\phi) = A(-\phi)Z^* + Z(0) - D(-\phi), \quad \text{Re } \phi = 0, \tag{2}$$

where

$$
Z(\phi) = \begin{pmatrix} Z_1(\phi) \\ \vdots \\ Z_s(\phi) \end{pmatrix}, \qquad Z^* = \begin{pmatrix} Z_1^* \\ \vdots \\ Z_s^* \end{pmatrix}, \qquad D(\phi) = \begin{pmatrix} D_1(\phi) \\ \vdots \\ D_s(\phi) \end{pmatrix},
$$

with

$$
Z_k^* = p_1 Z(k-2, s-k) + p_2 Z(k-1, s-k-1),
$$

and $D_k(\phi)$, $k = 1, 2, \ldots, s$, are functions which are bounded and analytic for Re $\phi \geqslant 0$; and where $H(\phi)$ is an $s \times s$-matrix given by

$$
H_{k,k}(\phi) = 1 - A(-\phi)\left[p_1 \frac{kb_1}{\phi + kb_1 + (s-k)b_2} + p_2 \frac{(s-k+1)b_2}{\phi + (k-1)b_1 + (s-k+1)b_2} \right],
$$

$$
k = 1, 2, \ldots, s;
$$

$$
H_{k,k+1}(\phi) = -A(-\phi) p_2 \frac{kb_1}{\phi + kb_1 + (s-k)b_2}, \quad k = 1, 2, \ldots, s-1;
$$

$$
H_{k,k-1}(\phi) = -A(-\phi) p_1 \frac{(s-k+1)b_2}{\phi + (k-1)b_1 + (s-k+1)b_2}, \quad k = 2, 3, \ldots, s; \tag{3}
$$

$$
H_{k,j}(\phi) = 0, \quad \text{otherwise};
$$

moreover, for $i \geqslant 0, j \geqslant 0, i + j \leqslant s - 2$

$$
Z(i,j) = \delta_{i+j,0} + \sum_{k=i}^{s-2} \sum_{l=j}^{s-2-k} b(i,j,k,l)[p_1 Z(k-1, l) + p_2 Z(k, l-1)]
$$

$$
+ \sum_{k=i+1}^{s-j} \sum_{n_1=i}^{k-1} \sum_{n_2=j}^{s-k} \binom{k-1}{i; n_1-i}\binom{s-k}{j; n_2-j}(-1)^{n_1+n_2+i+j} D_k(n_1 b_1 + n_2 b_2), \tag{4}
$$

where

$$
b(i,j,k,l) = \sum_{n_1=i}^{k} \sum_{n_2=j}^{l} \binom{k}{i; n_1-i}\binom{l}{j; n_2-j}(-1)^{n_1+n_2+i+j} A(n_1 b_1 + n_2 b_2),
$$

$$
0 \leqslant i \leqslant k, 0 \leqslant j \leqslant l, k + l \leqslant s - 1; \tag{5}
$$

$$
= 0, \quad \text{otherwise};
$$

and finally, for $k = 1, 2, \ldots, s$,

$$
D_k((k-1)b_1 + (s-k)b_2) = Z_k(0). \tag{6}
$$

In [1] the system (2)–(6) is solved using a factorization method. Next we give the result for the case considered here:

(i) det $H(\phi)$ can be written in the form

$$
\det H(\phi) = 1 + \sum_{l=0}^{s} \frac{1}{\phi + lb_1 + (s-l)b_2} \sum_{k=1}^{s} [A(-\phi)]^k a(l, k),
$$

where the $a(l, k)$ are constants not depending on ϕ;

(ii) if $\sum_{k=1}^{s}[A(lb_1 + (s-l)b_2)]^k a(l, k) \neq 0$ for $l = 1, 2, \ldots, s$ (Condition 1) then det $H(\phi)$ has exactly $s + 1$ zeros in the left half-plane Re $\phi < 0$, which we denote by $\mu_1, \mu_2, \ldots, \mu_{s+1}$;

(iii) assume that the μ_i, $i = 1, 2, \ldots, s + 1$, are distinct (Condition 2), let B_i be a (nonunique) nonzero

s-dimensional column vector satisfying

$$H(\mu_i)B_i = 0, \tag{7}$$

and let B be the $s \times (s+1)$-matrix whose ith column is B_i;

(iv) define the $(s+1) \times (s+1)$-matrix L by

$$L_{j,k} = \left[p_1 B_{j-1,k} + p_2 B_{j,k} \right] \frac{1}{\mu_k + (j-1)b_1 + (s-j+1)b_2},$$

$$j = 1, 2, \ldots, s+1, \quad k = 1, 2, \ldots, s+1,$$

where $B_{0,k} = B_{s+1,k} = 0$, $k = 1, 2, \ldots, s+1$; and the $(s+1) \times s$-matrix M by

$$M_{j,k} = p_1 \delta_{j,k+1} + p_2 \delta_{j,k};$$

assume det $L \neq 0$ (Condition 3) and define

$$C = L^{-1}M;$$

(v) the $s \times s$-matrix $K(\phi)$ is given by

$$K(\phi) = I + BJ(\phi)C,$$

where $J(\phi)$ is the diagonal $(s+1) \times (s+1)$-matrix with

$$J_{k,k}(\phi) = 1/(\phi - \mu_k), \quad k = 1, 2, \ldots, s+1;$$

(vi) define the $s \times s$-matrix $Q(\phi)$ by

$$Q_{j,k}(\phi) = K_{j,k}(\phi)/A(\phi) - p_1(s-j+1)K_{j-1,k}(\phi) - \left[p_1 j + p_2(s-j+1) \right] K_{j,k}(\phi)$$

$$-p_2 j K_{j+1,k}(\phi), \quad j = 1, 2, \ldots, s, \quad k = 1, 2, \ldots, s,$$

where $K_{0,k}(\phi) = K_{s+1,k}(\phi) = 0$, $k = 1, 2, \ldots, s$, and the $s \times s$-matrix S by

$$S_{j,k} = Q_{j,k}(-(j-1)b_1 - (s-j)b_2);$$

(vii) assume that det $S \neq 0$ (Condition 4) and define

$$G(\phi) = (K(0) - H(\phi)K(\phi))S^{-1};$$

and, for $i \geq 0, j \geq 0, i+j \leq s-2, k = 1, 2, \ldots, s$,

$$c(i,j,k) = \sum_{h=i+1}^{s-j} \sum_{n_1=i}^{h-1} \sum_{n_2=j}^{s-h} \binom{h-1}{i; \, n_1-i} \binom{s-h}{j; \, n_2-j} (-1)^{n_1+n_2+i+j} G_{h,k}(-n_1 b_1 - n_2 b_2);$$

(viii) number the elements of the set $\{(i,j) | i \geq 0, j \geq 0, i+j \leq s-2\}$ from 1 until $\binom{s}{2}$ by assigning to the pair (i,j) the number $n(i,j) = i + 1 + (s+i+j+1)(s-i-j-2)/2$; and define the $\binom{s}{2} \times \binom{s}{2}$-matrix T by

$$T_{k,l} = p_1 \left[b(k_1, k_2, l_1+1, l_2) + c(k_1, k_2, l+1) \right] + p_2 \left[b(k_1, k_2, l_1, l_2+1) + c(k_1, k_2, l) \right],$$

$$k = 1, 2, \ldots, \binom{s}{2}, \quad l = 1, 2, \ldots, s-1,$$

$$= p_1 b(k_1, k_2, l_1+1, l_2) + p_2 b(k_1, k_2, l_1, l_2+1),$$

$$k = 1, 2, \ldots, \binom{s}{2}, \quad l = s, s+1, \ldots \binom{s}{2},$$

where $k = n(k_1, k_2)$ and $l = n(l_1, l_2)$;

(ix) assume that the matrix $I - T$ has rank $\binom{s}{2} - 1$ (Condition 5) and construct the matrix T' from

$I - T$ by replacing the last row by the row vector T^0 given by

$$T_k^0 = 1 + p_1 \sum_{h=1}^{s} R_{h,k+1} + p_2 \sum_{h=1}^{s} R_{h,k}, \quad k = 1, 2, \ldots, s - 1,$$

$$= 1, \qquad\qquad\qquad\qquad\qquad k = s, s + 1, \ldots, \binom{s}{2},$$

where $R = K(0)S^{-1}$;

(x) Let i^0 be the $\binom{s}{2}$-dimensional column vector with the last component equal to 1 and the other components equal to 0, and let Z^0 be the column vector with components

$$Z_1^0, Z_2^0, \ldots, Z_{\binom{s}{2}}^0,$$

satisfying

$$T'Z^0 = i^0, \tag{8}$$

then, for $i \geqslant 0, j \geqslant 0, i + j \leqslant s - 2$, we have

$$Z(i,j) = Z_{n(i,j)}^0,$$

and, for $k = 1, 2, \ldots, s$,

$$Z_k(0) = R_{k,1} p_2 Z_1^0 + \sum_{h=2}^{s-1} R_{k,h} \left[p_1 Z_{h-1}^0 + p_2 Z_h^0 \right] + R_{k,s} p_1 Z_{s-1}^0;$$

(xi) define the column vector V with components $V_1, V_2, \ldots, V_{s+1}$ by

$$V = CK^{-1}(0)Z(0)$$

and, for $j = 1, 2, \ldots, s + 1$,

$$a_j = \frac{V_j}{\mu_j} \sum_{i=1}^{s} B_{ij};$$

(xii) let $A^*(\phi) = (1 - A(\phi)/(\alpha\phi))$, $\phi \neq 0$, and $A^*(0) = 1$, and define, for $i \geqslant 0, j \geqslant 0, i + j \leqslant s - 2$,

$$Z^*(i,j) = \sum_{k=i}^{s-1-j} \sum_{l=j}^{s-1-k} b^*(i,j,k,l) \left[p_1 Z(k-1,l) + p_2 Z(k,l-1) \right]$$

$$+ \sum_{h=i+1}^{s-j} \sum_{n_1=i}^{h-1} \sum_{n_2=j}^{s-h} \binom{h-1}{i; n_1 - i} \binom{s-h}{j; n_2 - j} (-1)^{n_1 + n_2 + i + j}$$

$$\times \sum_{l=1}^{s} \frac{V_l}{n_1 b_1 + n_2 b_2 + \mu_l} \left[h b_1 L_{h,l} + (s - h + 1) b_2 L_{h-1,l} \right]$$

$$\times \left[A^*(-\mu_l) - A^*(n_1 b_1 + n_2 b_2) \right],$$

where

$$b^*(i,j,k,l) = \sum_{n_1=i}^{k} \sum_{n_2=j}^{l} \binom{k}{i; n_1 - i} \binom{l}{j; n_2 - j} (-1)^{n_1 + n_2 + i + j} A^*(n_1 b_1 + n_2 b_2);$$

and define, for $j = 1, 2, \ldots, s + 1$,

$$d_j = \left(1 - 1/A(-\mu_j)\right) a_j / (\alpha\mu_j).$$

Now we have found all the parameters for the steady state distributions which are given by

$$P(W \leqslant x) = 1 + \sum_{j=1}^{s+1} a_j \exp(\mu_j x), \quad x \geqslant 0, \qquad P(W^* \leqslant x) = 1 + \sum_{j=1}^{s+1} d_j \exp(\mu_j x), \quad x \geqslant 0,$$

$$P(Q \leqslant k) = 1 + \sum_{j=1}^{s+1} a_j \left[A(-\mu_j) \right]^{k+1}, \quad k = 0, 1, \ldots,$$

$$P(Q^* \leqslant k) = 1 + \sum_{j=1}^{s+1} d_j \left[A(-\mu_j) \right]^{k+1}, \quad k = 0, 1, \ldots,$$

$$P(C \leqslant k) = \sum_{i+j \leqslant k} Z(i,j), \quad k = 0, 1, \ldots, s-2,$$

$$P(C^* \leqslant k) = \sum_{i+j \leqslant k} Z^*(i,j), \quad k = 0, 1, \ldots, s-2,$$

$$P(C \leqslant k) = 1 + \sum_{j=1}^{s+1} a_j \left[A(-\mu_j) \right]^{k-s+1}, \quad k = s-1, s, s+1, \ldots,$$

$$P(C^* \leqslant k) = 1 + \sum_{j=1}^{s+1} d_j \left[A(-\mu_j) \right]^{k-s+1}, \quad k = s-1, s, s+1, \ldots.$$

3. The numerical algorithm

The solution given by (i)–(xii) in the previous section implies a numerical algorithm for calculating the required parameters. In the case of Poisson arrivals it is apparent that $d_j = a_j$, and it may be shown that $Z^*(i,j) = Z(i,j)$ so that we can omit (xii). The zeros $\mu_1, \mu_2, \ldots, \mu_{s+1}$ turn out to be real. In order to determine them we have to calculate det $H(\phi)$ many times. The following recursive method is accurate and efficient. Take

$$h_0(\phi) = 1, \quad g_1(\phi) = -A(-\phi)\frac{sp_2 b_2}{\phi + sb_2}, \quad h_1(\phi) = 1 - A(-\phi)\left[\frac{p_1 b_1}{\phi + b_1 + (s-1)b_2} + \frac{sp_2 b_2}{\phi + sb_2} \right],$$

and, for $k = 2, 3, \ldots, s$, calculate

$$g_k(\phi) = -A(-\phi)\frac{(s-k+1)p_2 b_2}{\phi + (k-1)b_1 + (s-k+1)b_2}\left[g_{k-1}(\phi) + h_{k-2}(\phi) \right],$$

$$h_k(\phi) = g_k(\phi) + \left[1 - A(-\phi)\frac{kp_1 b_1}{\phi + kb_1 + (s-k)b_2} \right] h_{k-1}(\phi);$$

then det $H(\phi) = h_s(\phi)$.

The system (7) may be solved recursively since $H(\mu_i)$ is a Jacobi matrix and we may choose some convenient value for B_{1i}. For the solution of (8) we may take advantage of the fact that T' has the form

$$T' = \begin{pmatrix} T'_{11} & T'_{12} & & & \\ T'_{21} & T'_{22} & T'_{23} & & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \\ T'_{s-2,1} & T'_{s-2,2} & \cdots & T'_{s-2,s-2} & T'_{s-2,s-1} \\ T'_{s-1,1} & T'_{s-1,2} & \cdots & T'_{s-1,s-2} & T'_{s-1,s-1} \end{pmatrix}$$

where $T'_{k,l}$ is an $(s-k) \times (s-l)$-matrix.

One may apply the following accuracy checks:

(a) $\det K(0) = \prod_{k=1}^{s+1} \left[\dfrac{(k-1)b_1 + (s-k+1)b_2}{-\mu_k} \right]$;

(b) Z^0 should satisfy the last equation of $(I - T)Z^0 = 0$.

The algorithm has been programmed in Algol-60 and run on a DEC-10 computer. The CPU-time needed is hardly effected by the choice of the interarrival time distribution and the parameters β, c_S^2 and r_S. In Table 1 we show how it depends on the number of servers s.

Most of the CPU-time is spent on calculating the coefficients $c(i, j, k)$ (step(vii)). The number of operations in this part of the program is almost proportional to s^6. For $s = 20$ the CPU-times for the successive steps of the algorithm are approximately as follows: (i)–(iv): 2 seconds, (v)–(vi): 6 seconds, (vii): 65 seconds, (viii)–(xi): 15 seconds, and (xii): 7 seconds. For large s the total memory requirements amount to approximately $s^3 + 8s^2$ reals (double precision, 62 bits) and $4s^2$ integers. Up to $s = 15$ the algorithm turns out to be very stable and accurate even for extreme parameter values like $c_S^2 = 10$ and $r_S = 10$ or $r_S = 0.1$. For less extreme cases reasonably accurate results may still be obtained up to $s = 30$.

As an illustration we give some typical results for the model $GI/H_2/15$ with $\rho = 0.9$, $c_S^2 = 5$ and $r_S = 0.5$. For convenience we take $\beta = 1$. The percentile γ_p of a random variable X is defined by $P(X > \gamma_p) = p$. For the hyper-exponential interarrival time distributions we have chosen $r_A = 1$.

The program has been tested extensively. The results for the queue length in the system $M/H_2/s$ are in full agreement with results obtained by Hokstad [3], Marcel Neuts (personal communication) using the matrix geometric method, and Van Hoorn [5] using the method described in [4].

The approximations and simulation results for the waiting times in the system $M/H_2/s$ which were also published in [5] agree closely with our exact results. Known results from queuing theory, such as $L = \lambda W$,

Table 1

s	5	10	15	20	25	30
CPU-time in seconds	0.5	5	28	95	290	690

Table 2
Actual waiting time

GI	c_A^2	EW	Var W	$P(W=0)$	$\gamma_{0.01}$	$\gamma_{0.05}$	$\gamma_{0.10}$
D	0	0.8409	2.1941	0.4531	6.7632	3.9297	2.7099
E_5	0.2	0.8943	2.3966	0.4346	7.0550	4.1200	2.8563
E_2	0.5	0.9777	2.7162	0.4099	7.4924	4.4048	3.0753
M	1	1.1161	3.2913	0.3751	8.2209	4.8783	3.4390
H_2	2	1.3850	4.5571	0.3267	9.6305	5.7922	4.1392
H_2	5	2.1951	9.4414	0.2421	13.7846	8.4791	6.1941

Table 3
Virtual waiting time

GI	c_A^2	EW^*	Var W^*	$P(W^*=0)$	$\gamma_{0.01}$	$\gamma_{0.05}$	$\gamma_{0.10}$
D	0	0.8618	2.2218	0.4242	6.8004	3.9669	2.7471
E_5	0.2	0.9125	2.4193	0.4123	7.0846	4.1496	2.8859
E_2	0.5	0.9887	2.7309	0.3966	7.5108	4.4232	3.0937
M	1	1.1161	3.2913	0.3751	8.2209	4.8783	3.4390
H_2	2	1.3627	4.5224	0.3445	9.5954	5.7570	4.1041
H_2	5	2.1007	9.2662	0.2905	13.6487	8.3431	6.0582

Table 4
Number of customers at arrival epochs

GI	c_A^2	EQ	Var Q	P(Q = 0)	EC	Var C	P(C = 0)
D	0	11.083	394.75	0.4970	24.358	438.46	0.0737×10^{-6}
E_5	0.2	11.866	434.98	0.4787	25.194	480.21	0.1620×10^{-6}
E_2	0.5	13.055	498.89	0.4536	26.454	546.31	0.3800×10^{-6}
M	1	15.067	614.90	0.4174	28.567	665.74	0.9876×10^{-6}
H_2	2	19.006	873.75	0.3657	32.662	930.42	3.5649×10^{-6}
H_2	5	30.982	1895.74	0.2725	44.930	1966.01	17.7352×10^{-6}

Table 5
Number of customers in continuous time

GI	c_A^2	EQ^*	Var Q^*	$P(Q^* = 0)$	EC^*	Var C^*	$P(C^* = 0)$
D	0	11.352	399.97	0.4761	24.852	438.71	0.0232×10^{-6}
E_5	0.2	12.088	439.29	0.4620	25.588	480.45	0.0725×10^{-6}
E_2	0.5	13.200	501.70	0.4434	26.700	546.50	0.2480×10^{-6}
M	1	15.067	614.90	0.4174	28.567	665.74	0.9876×10^{-6}
H_2	2	18.698	867.00	0.3809	32.198	929.49	7.4171×10^{-6}
H_2	5	29.634	1860.77	0.3165	43.133	1957.66	87.0153×10^{-6}

Poisson arrivals see time averages and heavy traffic approximations have been used to check the internal consistency of our method.

References

[1] J.H.A. de Smit, "The queue $GI/M/s$ with customers of different types or the queue $GI/H_m/s$", Adv. Appl. Prob. 15, 392–419 (1983).

[2] J.H.A. de Smith, "The queue $GI/H_m/s$ in continuous time", J. Appl. Prob. 21, (1984), to appear.

[3] P. Hokstad, "Some numerical results and approximations for the many server queue with non-exponential service time", Dept. of Mathematics, University of Trondheim (1982).

[4] Y. Takashashi and Y. Takami, "A numerical method for the steady state probabilities of a $GI/G/c$ queueing system in a general class", J. Oper. Res. Soc. Japan 19, 147–157 (1976).

[5] M.H. van Hoorn, "Algorithms and approximations for queueing systems", Mathematical Centre, Amsterdam (1983).