Prediction of the Optimal Control in a Multi-Server Heterogeneous Queueing System *

Dmitry Efrosinin[1,2] and Natalia Stepanova[3]

[1] Johannes Kepler University Linz, 69 Altenbergerstrasse, 4030 Linz, Austria
[2] Peoples' Friendship University of Russia (RUDN University), 6 Miklukho-Maklaya St, 117198 Moscow, Russian Federation
dmitry.efrosinin@jku.at,efrosinin-dv@rudn.ru
http://www.jku.at,http://www.rudn.ru
[3] Institute of Control Problems, 65 Profsoyuznaya St, 117997 Moscow, Russian Federation
natalia0410@rambler.ru
http://www.ipu.ru

Abstract. In this paper we consider a multi-server heterogeneous queueing system where servers differ in service rates and operating costs. The optimal allocation policy for this system is of threshold type. Gathering data for evaluated optimal threshold levels and system parameters is performed using a policy iteration algorithm. We study the possibility to use these data-sets to provide predictions for optimal thresholds through artificial neural networks. The obtained results are accompanied by heuristic solution based on a fluid approximation. Numerical examples illustrate the quality of provided predictions.

Keywords: Heterogeneous Queueing System· Policy-Iteration Algorithm · Artificial Neural Networks · Heuristic Solution

1 Introduction

Many queueing systems are analyzed for their dynamic and optimal control related to system access, resource allocation, changing service area characteristics and so on. Sets of computerized tools and procedures provide large data-sets which can be useful to expand potential of classical optimization methods. The paper deals with a known model of a multi-server queueing system with controllable allocation of customers between heterogeneous servers which are differentiated by their service and cost attributes. As it is known, see e.g. [1,9], the optimal allocation policy which minimizes the long-run average cost per unit of time for this queueing system belongs to a set of structural policies. For the servers' enumeration (1) the allocation control policy denoted by f is defined through a sequence of threshold levels $1 = q_1 \leq q_2 \leq \cdots \leq q_K < \infty$. According

* The publication has been prepared with the support of the "RUDN University Program 5-100" (recipients D. Efrosinin, mathematical model development).

to this policy, the fastest server must be used whenever it is free and there are waiting customers in the queue. The kth server ($k \geq 2$) is used only if the first $k-1$ servers are busy and the queue length reaches a threshold level $q_k > 0$. In general case, the optimal threshold levels can depend on states of slower server and formally the optimal policy f is not of a pure threshold type. But since the kth threshold value may vary by at most one when the state of slower server changes and it has a weak effect on the average cost, such influence can be neglected. Hence the optimal allocation policy for multi-server heterogeneous queueing system can be treated as a threshold one.

Searching for the optimal values of $q_2, \dots, q_K$ by direct minimizing the average cost function can be expensive, especially when K is large. To calculate the optimal threshold levels we can use a policy iteration algorithm [4,6,10] which constructs a sequence of improved policies that converges to optimal one. Although this algorithm is a powerful tool for solving many optimization tasks, it has significant limitations on dimensionality of the model, number of states, convergence in a heavy traffic case. The contribution of the paper is two-fold. First, we provide a simple heuristic solution (HS) for a sub-optimal policy in order to avoid the search for the optimal one. Second, we investigate the possibility to use the data generated by a policy-iteration algorithm to provide a prediction for the optimal threshold levels with artificial neural networks (NN) [3,7,8]. The trained network can be used then to calculate the optimal thresholds for those system parameters for which alternative numerical methods are difficult or impossible to use, for example, in heavy traffic case, or, in general, to reconstruct the areas of optimality without usage of time-expensive algorithms and procedures. We unsuccessfully tried to find published works where machine learning methods would be used to solve a similar problem and therefore we consider this paper relevant.

This paper is organized as follows. In Section 2 we briefly discuss a mathematical model. Section 3 introduces some heuristic choices for threshold levels, that turn out to be nearly optimal. Section 4 presents results when the trained neural network was ran on verification data of the policy iteration algorithm.

2 Mathematical Model

We remind briefly the model under study. Consider an infinite-capacity $M/M/K$ queueing system with K heterogeneous servers and one common queue, see Figure 1. The customers arrive to the system according to a homogeneous Poisson process with a rate λ. The jth server has an exponentially distributed service time with a rate μ_j. The server j is called an available server if it is idle. The service of customers is assumed to be without preemption, i.e. a customer being served on a server can not change it. The inter-arrival and service times are mutually independent. The system costs include an operating cost $c_j > 0$ per unit of time for busy server j and holding cost $c_0 > 0$ per unit of time for any customer waiting in the queue. Assume that the servers are enumerated in a way

$$\mu_1 \geq \cdots \geq \mu_K,\ c_1\mu_1^{-1} \leq \cdots \leq c_K\mu_K^{-1}, \tag{1}$$

where $c_j \mu_j^{-1}$ stands for the mean operating cost per customer for the jth server.

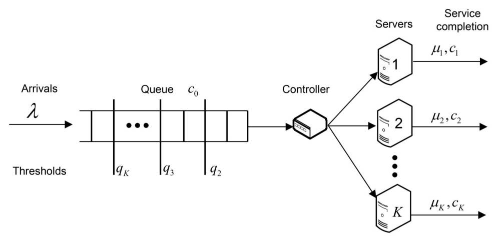

Fig. 1. Controllable multi-server queueing system with heterogeneous servers and operating costs.

The controller or decision maker, which has a full information about system states, dispatchers or allocates customers according to a control policy f either to one of available servers or to queue at a new arrival and service completion epoch if it occurs with a nonempty queue. The system dynamics is common for the systems with one queue and heterogeneous servers. At each arrival epoch the customer joins the queue and the controller can allocate the customer staying at the head of the queue to an available server j. At service completion epochs the controller may decide to allocate the customer from the head of nonempty queue to an available server or leave the customer in the queue. As it was mentioned above, the optimal control policy, which minimizes the long-run average cost per unit of time, belongs to a set of threshold policies defined as a sequence of threshold levels

$$1 = q_1 \leq q_2 \leq \cdots \leq q_K < \infty. \quad (2)$$

According to this policy the first k servers must be occupied whenever there are q customers in the queue and $q_k \leq q \leq q_{k+1} - 1$.

We formulate the above optimization problem as a Markov decision problem associated with a multi-dimensional continuous-time Markov chain $\{X(t)\}_{t \geq 0} = \{Q(t), D_1(t), \ldots, D_K(t)\}_{t \geq 0}$ with a set of admissible actions $A = \{0, 1, \ldots, K\}$ with elements a, where $a = 0$ means the allocation of the customer to the queue and $a = j \neq 0$ – to the jth server. The term $Q(t) \in \mathbb{N}_0$ denotes the number of customers in the queue at time t, $D_j(t) \in \{0, 1\}$ – the number of customers at server j at time t. For any fixed policy f, which is of threshold type with levels $(q_2, \ldots, q_K)$, we wish to guarantee that the process $\{X(t)\}_{t \geq 0}$ is an irreducible, positive recurrent Markov chain with a state space $E^f = \{x = (q(x), d_1(x), \ldots, d_K(x))\} \subseteq \mathbb{N}_0 \times \{0, 1\}^K$ and infinitesimal generator Λ^f. The notations $q(x)$ and $d_j(x)$ will be used further in the paper to specify the components of the vector state $x \in E^f$, where $q(x)$ denotes the queue length in state x and $d_j(x)$ – the state of the jth server in state x. The stability condition is obviously defined through the inequality $\lambda < \sum_{j=1}^{K} \mu_j$. For ergodic Markov chains with costs the long-run average cost per unit of time for the policy f coincides with the corresponding assemble average, i.e.

$$g^f = \limsup_{t \to \infty} \frac{1}{t} V^f(x,t) = \sum_{y \in E^f} c(y) \pi_y^f, \tag{3}$$

where $c(y) = c_0 q(y) + \sum_{j=1}^{K} c_j d_j(y)$ is an immediate cost in state $y \in E^f$,

$$V^f(x,t) = \mathbb{E}^f \Big[\int_0^t \Big(c_0 Q(t) + \sum_{j=1}^{K} c_j D_j(t) \Big) dt | X(0) = x \Big]$$

denotes the total average cost up to time t given initial state is x and $\pi_y^f = \mathbb{P}^f[X(t) = y]$ is a stationary state probability of the process under given policy f. The policy f^* is said to be optimal when for g^f defined in (3) we evaluate

$$g^* = \inf_f g^f = \min_{q_2, \ldots, q_K} g(q_2, \ldots, q_K). \tag{4}$$

Algorithm 1 Policy-iteration algorithm

1: **procedure** PIA($K, W, \lambda, \mu_j, c_j, j = 1, 2, \ldots, K,\ c_0$)
2: $\quad f^{(0)}(x) = \mathrm{argmin}_{j \in J_0(x)} \left\{ \frac{c_j}{\mu_j} \right\}$ ▷ Initial policy
3: $\quad n \leftarrow 0$
4: $\quad g^{(n)} \leftarrow \lambda v^{(n)}(\mathbf{e}_1)$ ▷ Policy evaluation
5: $\quad$ **for** $x = (0, 1, 0, \ldots, 0)$ **to** $(N, 1, 1, \ldots, 1)$ **do**
6:

$$\begin{aligned} v^{(n)}(x) \leftarrow \frac{1}{\lambda + \sum_{j \in J_1(x)} \mu_j} \Big[& c(x) - g^{(n)} + \lambda v^{(n)}(x + \mathbf{e}_{f^{(n)}(x)}) \\ & + \sum_{j \in J_1(x)} \mu_j v^{(n)}(x - \mathbf{e}_j) 1_{\{q(x)=0\}} \\ & + \sum_{j \in J_1(x)} \mu_j v^{(n)}(x - \mathbf{e}_j - \mathbf{e}_0 + \mathbf{e}_{f^{(n)}(x - \mathbf{e}_j - \mathbf{e}_0)}) 1_{\{q(x)>0\}} \Big] \end{aligned}$$

7: $\quad$ **end for**
8: ▷ Policy improvement

$$f^{(n+1)}(x) \leftarrow \mathrm{argmin}_{a \in A(x)} v^{(n)}(x + \mathbf{e}_a)$$

9: $\quad$ **if** $f^{(n+1)}(x) \leftarrow f^{(n)}(x)$, $x \in E_X$ **then return** $f^{(n+1)}(x), v^{(n)}(x), g^{(n)}$
10: $\quad$ **else** $n \leftarrow n + 1$, **go to step 4**
11: $\quad$ **end if**
12: **end procedure**

To evaluate optimal threshold levels and optimized value for the mean number of customers in the system the policy-iteration algorithm 1 is used. Here we use the notations

$$J_0(x) = \{j : d_j(x) = 0\},\ J_1(x) = \{j : d_j(x) = 1\}$$

to specify respectively a set of idle and busy servers in state x, $A(x) = J_0(x) \cup \{0\} \subseteq A$ the subset of admissible actions in state x and $\mathbf{e}_j$ stands for a vector of dimension $K+1$ with 1 in the jth position ($j = 0, 1, \ldots, K$) and 0 elsewhere. We convert the $K+1$-dimensional state space E^f of the Markov decision process ordered in a certain way to a one-dimensional equivalent state space $\mathbb{N}_0$, $\Delta : E^f \to \mathbb{N}_0$, for state $x = (q(x), d_1(x), \ldots, d_K(x)) \in E^f$,

$$\Delta(x) = q(x)2^K + \sum_{i=1}^{K} d_i(x)2^{i-1}. \tag{5}$$

Therefore, in one-dimensional case the changing of the state x due to adding or removing a customer from the queue and due to occupation or departure of a customer from the jth server can be respectively represented in the form,

$$\Delta(x \pm \mathbf{e}_0) = (q(x) \pm 1)2^K + \sum_{i=1}^{K} d_i(x)2^{i-1} = \Delta(x) \pm 2^K,$$
$$\Delta(x \pm \mathbf{e}_j) = q(x)2^K + \sum_{i=1}^{K} d_i(x)2^{i-1} \pm 2^{j-1} = \Delta(x) \pm 2^{j-1}.$$

For further details about derivation of the dynamic programming equation needed to evaluate the optimal policy the interested readers are referred to [1]. The infinite buffer queueing system is approximated by a finite buffer equivalent system in such a way that the loss probability does not exceed some specified small number $\varepsilon > 0$.

Remark 1. For the bounded buffer size W the number of states is

$$|E^f| = 2^K(W+1).$$

If the queue length $q \geq q_K$, all servers must be busy and the system behaves like a $M/M/1$ queueing system with a service rate $\sum_{j=1}^{K} \mu_j$. The stationary state probabilities $\pi_{(q,1,\ldots,1)}, q \geq q_K$, satisfy the difference equation

$$\lambda\pi_{(q-1,1,\ldots,1)} - \Big(\lambda + \sum_{j=1}^{K} \mu_j\Big)\pi_{(q,1,\ldots,1)} + \sum_{j=1}^{K} \mu_j \pi_{(q+1,1,\ldots,1)} = 0,$$

which has a solution in a geometric form, $\pi_{(q,1,\ldots,1)} = \pi_{(q_K,1,\ldots,1)}\rho^{q-q_K}, q \geq q_K$. For details and theoretical substantiation see e.g. [2]. The threshold level q_K can be estimated using HS (7). The buffer size W is chosen in such a way that it satisfies the condition for the loss probability

$$\sum_{q=W}^{\infty} \pi_{(q,1,\dots,1)} = \pi_{q_K} \sum_{q=W}^{\infty} \rho^{q-q_K} \leq \sum_{q=W}^{\infty} \rho^{q-q_K} = \frac{\rho^{W-q_K}}{1-\rho} < \varepsilon,$$

where $\rho = \frac{\lambda}{\sum_{j=1}^{K} \mu_j}$. After simple algebra it implies

$$W > \frac{\log \varepsilon(1-\rho)}{\log(\rho)} + q_K.$$

Example 1. Consider the system $M/M/5$ with $K = 5$ and $\lambda = 15$. All other parameters take the following values

j	0	1	2	3	4	5
c_j	1	5	4	3	2	1
μ_j	-	20	8	4	3	1
$c_j \mu_j^{-1}$	-	0.25	0.50	0.75	0.67	1.00

The buffer size is $W = 80$ which guarantees $W > \frac{\log 0.0001(1-14/36)}{\log(14/36)} + q_5 = 22.2734$ for $\varepsilon = 0.0001$, where $q_5 = 12$ is evaluated by (7). The table of evaluated control actions $f(x)$ for selected system states x is of the form:

System state x	Queue length $q(x)$													
$d = (d_1, d_2, d_3, d_4, d_5)$	0	1	2	3	4	5	6	7	8	9	10	11	12	...
(0,*,*,*,*)	**1**	1	1	1	1	1	1	1	1	1	1	1	1	1
(1,0,*,*,*)	0	0	**2**	2	2	2	2	2	2	2	2	2	2	2
(1,1,0,*,*)	0	0	0	**3**	3	3	3	3	3	3	3	3	3	3
(1,1,1,0,*)	0	0	0	0	**4**	4	4	4	4	4	4	4	4	4
(1,1,1,1,0)	0	0	0	0	0	0	0	0	0	0	0	**5**	5	5
(1,1,1,1,1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Threshold levels q_k, $k = 1, \dots, K = 5$, can be evaluated by comparing the optimal actions

$$f(q, \underbrace{1, \dots, 1}_{k-1}, \underbrace{0, \dots, 0}_{K-k+1}) < f(q+1, \underbrace{1, \dots, 1}_{k-1}, \underbrace{0, \dots, 0}_{K-k+1})$$

for $q = 0, \dots, W-1$. In this example the optimal policy f^* is defined here through a sequence of threshold levels $(q_2, q_3, q_4, q_5) = (3, 4, 5, 12)$ and $g^* = 4.92897$.

The data needed either to verify the heuristic solution or for training and verification of the neural network was generated by a policy-iteration algorithm in form of the list

$$\begin{aligned} S = \Big\{ & (\lambda, \mu_1, \dots, \mu_K, c_0, c_1, \dots, c_K) \to (q_2, \dots, q_K) : \\ & \lambda \in [1, 45], \mu_1, \dots, \mu_K \in [1, 40], c_0 \in [1, 3], c_1, \dots, c_K \in [1, 5], \\ & \lambda < \sum_{j=1}^{K} \mu_j, \mu_1 \geq \dots \geq \mu_K, c_1 \mu_1^{-1} \leq \dots \leq c_K \mu_K^{-1} \Big\}. \end{aligned} \tag{6}$$

Example 2. Some elements of the list S for the $M/M/5$ queueing system are

$$\begin{aligned}
&(1, 20, 8, 4, 2, 1, 1, 1, 1, 1, 1, 1) \rightarrow (2, 5, 13, 30),\\
&(10, 20, 8, 4, 2, 1, 1, 1, 1, 1, 1, 1) \rightarrow (1, 4, 9, 21),\\
&(1, 20, 8, 4, 2, 1, 1, 5, 4, 3, 2, 1) \rightarrow (5, 12, 20, 20),\\
&(10, 20, 8, 4, 2, 1, 1, 5, 4, 3, 2, 1) \rightarrow (3, 8, 13, 13).
\end{aligned}$$

3 Heuristic Solution

As it was mentioned above, the policy iteration algorithm has restrictions on dimensionality of the model, number of states, convergence in a heavy traffic case. In this section we derive a heuristic solution (HS) to estimate threshold levels $q_k, k = 2, \ldots, K$, for the arbitrary K using a simple discrete fluid approximation which is illustrated in Figure 2. Assume that q_k is an optimal threshold to

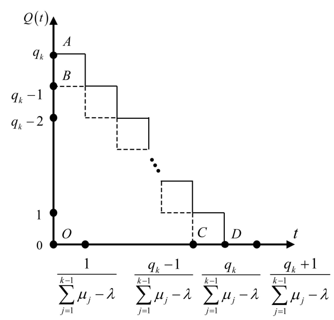

Fig. 2. Fluid approximation.

allocate the customer to server k in state $(q_k - 1, \underbrace{1, \ldots, 1}_{k-1}, \underbrace{0, \ldots, 0}_{K-k+1})$, where the first $k - 1$ servers are busy. Now we compare the queues of the system given initial state is $x_0 = (q_k, \underbrace{1, \ldots, 1}_{k-1}, 0, \underbrace{0, \ldots, 0}_{K-k})$, where the kth server is not used for a new customer, and $y_0 = (q_k - 1, \underbrace{1, \ldots, 1}_{k-1}, 1, \underbrace{0, \ldots, 0}_{K-k})$, where the kth server is occupied by a waiting customer. It is assumed that the stability condition holds. In Figure 2, the queue lengths are labeled by $A = q_k$ and $B = q_k - 1$. If the queue dynamics corresponded to the deterministic fluid, it would decrease at the rate $\sum_{j=1}^{k-1} \mu_j - \lambda$. When this rate is maintained until the queue is empty, it occurs respectively at points $D = \frac{q_k}{\sum_{j=1}^{k-1} \mu_j - \lambda}$ and $C = \frac{q_k - 1}{\sum_{j=1}^{k-1} \mu_j - \lambda}$. The total holding times of customers in a queue with lengths q_k and $q_k - 1$ are equal obviously to the areas

$$F_{AOD} = \frac{q_k(q_k + 1)}{2} \cdot \frac{1}{\sum_{j=1}^{k-1} \mu_j - \lambda} \quad \text{and} \quad F_{BOC} = \frac{q_k(q_k - 1)}{2} \cdot \frac{1}{\sum_{j=1}^{k-1} \mu_j - \lambda}$$

of triangles AOD and BOC. The mean operating cost of the first $k-1$ servers until the queue is empty starting from state x_0 is equal to

$$q_k\Big(\frac{c_1}{\mu_1}\frac{\mu_1}{\sum_{j=1}^{k-1}\mu_j}+\cdots+\frac{c_{k-1}}{\mu_{k-1}}\frac{\mu_{k-1}}{\sum_{j=1}^{k-1}\mu_j}\Big)=q_k\frac{\sum_{j=1}^{k-1}c_j}{\sum_{j=1}^{k-1}\mu_j},$$

where $\frac{\mu_i}{\sum_{j=1}^{k-1}\mu_j}$ is a probability to be served by the ith server, and starting from state y_0 – is equal to $(q_k-1)\frac{\sum_{j=1}^{k-1}c_j}{\sum_{j=1}^{k-1}\mu_j}$. According to a specified deterministic fluid

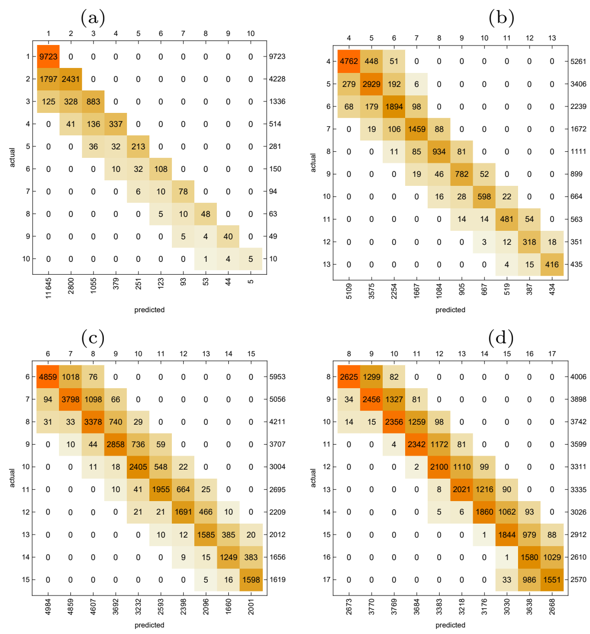

Fig. 3. Confusion matrices (a)–(d) for prediction of q_2, q_3, q_4 and q_5 using HS

schema we formulate

Proposition 1. *The optimal thresholds* $q_k, k=2,\dots,K$, *are defined by*

$$q_k \approx \hat{q}_k = \min\Big\{1, \Big\lfloor \frac{\sum_{j=1}^{k-1}\mu_j - \lambda}{c_0}\Big[\frac{c_k}{\mu_k}-\frac{\sum_{j=1}^{k-1}c_j}{\sum_{j=1}^{k-1}\mu_j}\Big]\Big\rfloor\Big\}. \tag{7}$$

Proof. Denote by $V(x)$ the overall average system cost until the system is empty given initial state is $x\in E^f$. The decision to perform the allocation to the kth server in state $(q_k-1,\underbrace{1,\dots,1}_{k-1},\underbrace{0,\dots,0}_{K-k+1})$ must lead to a reduction of the overall system costs under fluid schema, i.e.

$$V(x_0) - V(y_0) > 0. \tag{8}$$

where

$$V(x_0) = c_0 F_{AOD} + q_k \frac{\sum_{j=1}^{k-1} c_j}{\sum_{j=1}^{k-1} \mu_j} + V(0, \underbrace{1, \dots, 1}_{k-1}, \underbrace{0, \dots, 0}_{K-k+1}), \tag{9}$$

$$\begin{aligned} V(y_0) &= \frac{c_k}{\mu_k} + V(q_k - 1, \underbrace{1, \dots, 1}_{k-1}, 0, \underbrace{0, \dots, 0}_{K-k}) \\ &= \frac{c_k}{\mu_k} + c_0 F_{BOC} + (q_k - 1)\frac{\sum_{j=1}^{k-1} c_j}{\sum_{j=1}^{k-1} \mu_j} + V(0, \underbrace{1, \dots, 1}_{k-1}, \underbrace{0, \dots, 0}_{K-k+1}). \end{aligned}$$

After substitution of (9) into (8) and some simple manipulations we get that the heuristic solution for the optimal threshold q_k is defined then as the integer larger then 1 and the smallest integer (7) satisfying the inequality (8).

Example 3. Consider a queueing system from previous example for $K = 5$. We select randomly from the data-set S (6) a list of system parameters $(\lambda, \mu_1, \dots, \mu_K, c_0, c_1, \dots, c_K)$ and evaluate with HS the corresponding thresholds $q_k, k = 1, \dots, K$. Confusion matrices in Figure 3 visualize the performance of proposed heuristics respectively for the threshold levels (q_2, q_3, q_4, q_5). Each row of these matrices represents the instances in a predicted value while each column represents the instances in an actual value. The overall accuracies, i.e. the metric which describe the closeness of the measurements to a specific value, as well as the accuracies for results with possible deviation of threshold values by ± 1 from the real value are summarized in Table 1.

HS	q_2	q_3	q_4	q_5
Accuracy	0.8430	0.8778	0.7899	0.6282
Accuracy ± 1	0.9861	0.9884	0.9871	0.9769

Table 1. Accuracy for prediction with HS

4 Artificial Neural Networks

Artificial Neural Networks (NN) is a part of a supervised machine learning which is most popular in different problems of data classification, pattern recognition, regression, clustering, time series forecasting. Here we show that the NN can give even more positive results comparing to the HS that indicates the possibility to

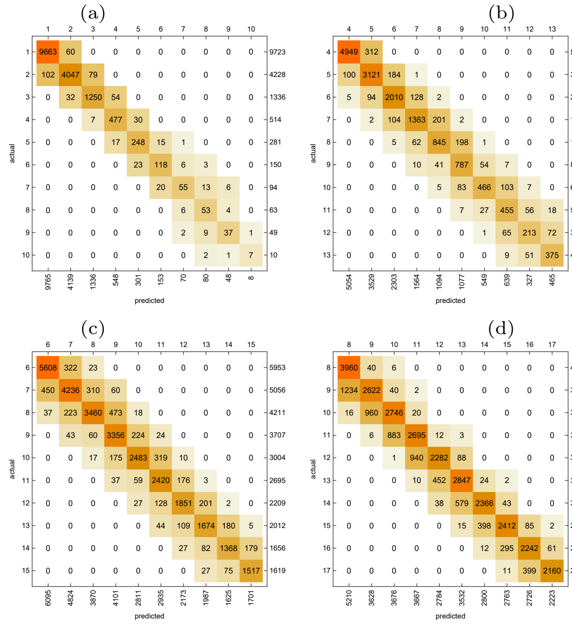

Fig. 4. Confusion matrices (a)–(d) for prediction of q_2, q_3, q_4 and q_5 using NN

use it for predicting the structural control policies. The data-set S (6) is used to explore predictions for the optimal threshold levels through the NN. 70% of the same data S which was not used for HS is referred to as training data and the rest of S – as validation data. We train a multilayer (6-layer) NN using an adaptive moment estimation method [5] and the neural network toolbox in *Mathematica*© of the Wolfram Research. Then we verify the approximated function

$$\hat{q}_k := \hat{q}_k(\lambda, \mu_1, \dots, \mu_K, c_0, c_1, \dots, c_K),$$

which should be accurate enough to be used to predict new output from verification data. The algorithm was ran many times on samples and networks with different sizes. In all cases the results were quite positive and indicate the potential of machine learning methodology for optimization problems in the queueing theory.

Example 4. The results of predictions in framework of some typical example are summarized in form of confusion matrices shown in Figure 4. The overall accuracy of classification and accuracies for the values with deviations are given in Table 2. We can see that the NN methodology exhibits more accurate predictions for the optimal thresholds comparing to the HS. Therefore we may conclude that classical queueing system analysis can and must be supplemented and extended by more active use of machine learning technologies.

NN	q_2	q_3	q_4	q_5
Accuracy	0.9700	0.8785	0.8708	0.7977
Accuracy ± 1	0.9991	0.9951	0.9874	0.9962

Table 2. Accuracy for prediction with NN

5 Conclusion

We combine classic methodology of analyzing controllable queues with a heuristic solution and machine learning to study the possibility to forecast the values of optimal thresholds. When analyzing and comparing the results obtained by algorithms of the Markov decision theory and by supervised learning we may conclude that these methodologies can be seen as complementary rather than competitive.

References

1. Efrosinin, D.: Controlled queueing systems with heterogeneous servers: Dynamic optimization and monotonicity properties of optimal control policies in multiserver heterogeneous queues. VDM Verlag, Saarbr"ucken, Germany (2008)
2. Efrosinin, D., Sztrik, J.: An algorithmic approach to analyzing the reliability of a controllable unreliable queue with two heterogeneous servers. European Journal of Operational Research **271**, 934–952 (2018)
3. Gershenson, C.: Artificial neural networks for beginners, `http://arxiv.org/abs/cs/0308031` (2003)
4. Howard, R.: Dynamic programming and Markov processes. Wiley Series, London, UK (1960)
5. Kingma, D. P., Ba, J. L. Adam: A method for stochastic optimization, `https://arxiv.org/abs/1412.6980` (2015)
6. Puterman, M. L.: Markov decision process. Wiley series in Probability and Mathematical Statistics, London, UK (1994)
7. Rätsch, G.: A brief introduction into machine learning. Friedrich Miescher Laboratory of the Max Planck Society (2004)
8. Russel, S. J., Norvig, P.: Artificial intelligence A modern approach. Prentice-Hall, Inc. (1995)
9. Rykov, V., Efrosinin, D.: On the slow server problem. Automation and Remote Control **70**, 2013–2023 (2010)
10. Tijms, H.C.: Stochastic models. An algorithmic approach. John Wiley and Sons, New-York, USA (1994)