Markov decision process and linear programming based control of MAP/MAP/N queues

András Mészáros[1,3] and Miklós Telek[1,2]

[1] Budapest University of Technology and Economics,
[2] MTA-BME Information systems research group,
[3] Inter-University Center for Telecommunications and Informatics Debrecen,
{meszarosa,telek}@hit.bme.hu

Abstract. We investigate the control problem of the optimal choice of idle server (if any) for arriving customer in order to minimize the mean system time (waiting time + service time). The considered MAP/MAP/N queue consists of a common infinite buffer and multiple identical servers with MAP service processes whose phases (internal states) are known. Customers arrive according to a MAP (whose phase is also known) and are served with work conserving policy. Idle servers preserve their phases.

We transform the obtained infinite state optimization problem to a finite state one and apply two optimization procedures, policy iteration of finite state MDP and linear programming.

Keywords: Markov arrival process, Markov decision process, MAP/MAP/N queue.

1 Introduction

Suboptimal control of multi-server systems may result in lower utilisation and, consequently, higher system time, therefore finding the optimal control scheme in these systems may be critical. Different types of queueing systems have been analysed from the point of view of optimal control. Earlier works typically consider Poisson arrival process and exponential service time, see e.g. [11] for a survey. The work dealing with problems closest to our topic is probably that of Efrosinin [5], which analyses several types of queueing systems including the MAP/PH/K/B-K structure. Efrosinin uses Markov decision processes (MDPs) to investigate various multi-server systems with a common finite queue and independent service times. In our work we consider an infinite queue with correlated arrival and service times, characterised by Markov arrival processes (MAPs).

The direct Markov chain description of the infinite queueing system contains infinite states, for which the classical MDP and linear programming (LP) solution techniques cannot be applied. With the use of the matrix analytic methodology, however, we have derived two finite state formalizations of the optimization problem which can be used to find the optimal policy of the infinite system employing finite MDP solvers and LP, respectively. According to [6], the finite MDP formalization of the problem ensures that a pure stationary optimal control policy exists. The LP formalization is an alternative description of the optimization problem which is more efficient in certain cases.

The rest of the paper is organised as follows. In Section 2 we present the necessary theoretical background on MAPs and MDPs. In Section 3 we provide the matrix analytic description of the MAP/MAP/N system. Based on this description we give the finite state MDP model and the LP description of the problem in Section 4 and 5. Section 6 presents some numerical results, finally Section 7 concludes the paper.

2 Background

In the following we will use the form $\boldsymbol{M}$ to denote matrices without and $\boldsymbol{M_k}$ with an index. For their elements in position (i, j) we will use notations $\boldsymbol{M}_{i,j}$ and $\boldsymbol{M_k}_{i,j}$ respectively. Furthermore we will use $\mathbb{1}$ to denote a column vector of 1s and $\underline{e_i}$ to denote a column vector for which $\underline{e_i}_j = \delta_{i,j}$, where $\delta_{i,j}$ is the Kronecker delta.

2.1 Markov arrival processes

The standard description of a MAP is given using the square matrix pair $(\boldsymbol{D_0}, \boldsymbol{D_1})$, where $\boldsymbol{D_0}+\boldsymbol{D_1}$ is the infinitesimal generator of the background CTMC [8]. $\boldsymbol{D_0}$ describes state transitions without arrival and $\boldsymbol{D_1}$ with an arrival. The average arrival intensity of a MAP is $\mu = \frac{1}{\underline{v}(-\boldsymbol{D_0})^{-1}\mathbb{1}}$, where $\underline{v}$ is the solution of the system of linear equation $\underline{v}\left((-\boldsymbol{D_0})^{-1}\boldsymbol{D_1} - \boldsymbol{I}\right) = 0$ and $\underline{v}\mathbb{1} = 1$.

2.2 Markov Decision Processes

Definition 1 *Let us consider a process $X(k)$ on a discrete time Markov chain with state space S, a set of decisions A, a set of transition probability matrices $P = \{\boldsymbol{P_a},\ a \in A\}$ such that $\boldsymbol{P_a}_{i,j} = Pr(X(k+1) = j|X(k) = i, a_k = a),\ \forall i, j \in S,\ a \in A,\ k \in \mathrm{N}$ and a set of cost functions $C = \{C_a(s),\ a \in A,\ s \in S\}$. We say that the tuple (S, A, P, C) is a Markov decision process.*

MDPs are powerful tools for optimal control of Markovian systems [11]. The previous definition stands only for discrete time homogeneous MDPs and can be generalised to continuous time and heterogeneous cases, but the above definition is sufficient in the current discussion. We also note that S can be finite or infinite, however the common algorithms are only applicable for the finite case.

Any function $\pi(s)$ that assigns an $a \in A$ to every $s \in S$ is called a strategy. The standard problem of MDPs is to find an optimal strategy, i.e. a $\pi(s)^*$ that minimizes a given objective function. The objective function used in this paper is the average cost per step in steady state, thus the optimal policy is

$$\pi^* = \arg\min_{\pi} \mathrm{E}_{\pi}\left[\lim_{k\to\infty}\frac{1}{k}\sum_{i=1}^{k} C_{\pi(X(i))}(X(i))\right], \tag{1}$$

or equivalently

$$\pi^* = \arg\min_{\pi} \sum_{s\in S} \alpha_{\pi}(s) C_{\pi(s)}(s), \tag{2}$$

where $\alpha_{\pi}(s)$ is the steady state probability of being in state s for policy π.

The previous description stands for pure strategies (i.e. we always make the same decision in a state with 1 probability). In general a convex combination of pure strategies (called mixed strategies) can be considered as well, however, as shown in [6], there always exists a pure strategy that gives the optimum for the average cost per step problem.

3 Infinite state description of the queueing system

The service processes of the N servers of a MAP/MAP/N queue are stochastically identical, but in the considered control problem apart of the pair of matrices characterizing the MAP service process the phase of the individual servers are also known and the available idle servers (if any) are distinguished based on this information at customer arrival. The goal is to find the policy for assigning the arriving customer with the optimal idle server.

The natural structured representation of the MDP characterizing the MAP/MAP/N queue with this control option is presented below for $N = 2$. Extension to more servers is quite straightforward, but would needlessly complicate the description. We refer to the totality of states that have k customers in the system as level k and denote state i of level k as (k, i). Levels for which the number of customers in the system is higher than the number of servers are called regular the others are called irregular. Let the MAP describing the inter-arrival times be of size n_a and defined by $(\boldsymbol{A_0}, \boldsymbol{A_1})$ and the MAP describing the service times be of size n_s and defined by $(\boldsymbol{S_0}, \boldsymbol{S_1})$. We recall again that we consider only work conserving schemes. Thus the MAP/MAP/2 queue can be described as a continuous time Markov chain that has the standard structure of a quasi birth-death (QBD) process [8] with infinitesimal generator

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{L}_0 & \boldsymbol{F}_0 & \boldsymbol{0} & \cdots & & \\ \boldsymbol{B}_1 & \boldsymbol{L}_1 & \boldsymbol{F}_1 & \boldsymbol{0} & \cdots & \\ \boldsymbol{0} & \boldsymbol{B}_2 & \boldsymbol{L} & \boldsymbol{F} & \boldsymbol{0} & \\ \vdots & \boldsymbol{0} & \boldsymbol{B} & \boldsymbol{L} & \boldsymbol{F} & \ddots \\ & & \ddots & \ddots & \ddots & \ddots \end{pmatrix}, \tag{3}$$

where

$$\boldsymbol{L_0} = \boldsymbol{A_0} \otimes \boldsymbol{I}(n_s^2), \qquad \boldsymbol{F_0} = \left(\boldsymbol{\Delta}(\boldsymbol{A_1} \otimes \boldsymbol{I}(n_s^2) \mid (\boldsymbol{I}(n_a) - \boldsymbol{\Delta})(\boldsymbol{S_1} \otimes \boldsymbol{I}(n_s^2))\right),$$

$$\boldsymbol{F_1} = \begin{pmatrix} \boldsymbol{A_1} \otimes \boldsymbol{I}(n_s^2) \\ \hline \boldsymbol{A_1} \otimes \boldsymbol{I}(n_s^2) \end{pmatrix}, \qquad \boldsymbol{B_1} = \begin{pmatrix} \boldsymbol{I}(n_a n_s) \otimes \boldsymbol{S_1} \\ \hline \boldsymbol{I}(n_a) \otimes \boldsymbol{S_1} \otimes \boldsymbol{I}(n_s) \end{pmatrix},$$

$$\boldsymbol{L_1} = \begin{pmatrix} \boldsymbol{A_0} \otimes \boldsymbol{I}(n_s^2) + \boldsymbol{I}(n_a n_s) \otimes \boldsymbol{S_0} \mid & 0 \\ \hline 0 & \mid \boldsymbol{A_0} \otimes \boldsymbol{I}(n_s^2) + \boldsymbol{I}(n_a) \otimes \boldsymbol{S}_0 \otimes \boldsymbol{I}(n_s) \end{pmatrix},$$

$$\boldsymbol{B_2} = \left(\boldsymbol{I}(n_a n_s) \otimes \boldsymbol{S_1} \mid \boldsymbol{I}(n_a) \otimes \boldsymbol{S_1} \otimes \boldsymbol{I}(n_s) \right), \qquad \boldsymbol{L} = \boldsymbol{A_0} \oplus \boldsymbol{S_0} \oplus \boldsymbol{S_0},$$

$$\boldsymbol{F} = \boldsymbol{A_1} \otimes \boldsymbol{I}(n_s^2), \qquad \boldsymbol{B} = \boldsymbol{I}(n_a) \otimes \boldsymbol{S_1} \otimes \boldsymbol{I}(n_s) + \boldsymbol{I}(n_a n_s) \otimes \boldsymbol{S_1},$$

and $\boldsymbol{I}(x)$ is the identity matrix of size x.

As the system is work conserving (i.e. servers can only be idle, if the queue is empty), decisions have to be made at the arrival of a new customer on levels $i = 0, \ldots, N-2$, that is, when 2 or more servers are empty at the time of an arrival. For the considered $N = 2$ case this means only the 0th level. The specific control here is determined by diagonal matrix $\boldsymbol{\Delta}$. Assuming that the size of the arrival MAP is n_a and the size of the service MAP is n_s $\boldsymbol{\Delta}$ has the following special structure (for N=2):

$$\boldsymbol{\Delta}_{i,i} = \begin{cases} p_{j,k,k} = 0.5, & \text{if } i = (j-1) * n_s^2 + (k-1) * n_s + k, \\ p_{j,k,l}, & \text{if } i = (j-1) * n_s^2 + (k-1) * n_s + l, \\ 1 - p_{j,k,l}, & \text{if } i = (j-1) * n_s^2 + (l-1) * n_s + k, \end{cases}$$

where $j = 1, \ldots, n_a$ and $k, l = 1 \ldots, n_s$, with $k < l$. Parameter $p_{j,k,l}$ is the probability, that we choose the first server if the MAP of the arrival is in phase j, the MAP of the first server is in phase k, and the MAP of second server is in phase l. From this $0 \le p_{j,k,l} \le 1$. If both servers are in the same state we choose both with the same 0.5 probability. Otherwise, the only constraint is that $p_{j,k,l} = 1 - p_{j,l,k}$ for any given j, k, l set. This constraint corresponds to the assumption that the probability of choosing the server in phase k does not depend on whether it is labeled first or second.

The steady state solution of the system is partitioned according to the levels as $\boldsymbol{\alpha} = \left(\boldsymbol{\alpha_0}\ \boldsymbol{\alpha_1}\ \boldsymbol{\alpha_2} \ldots \right)$. Due to the level independent behaviour of (3) for $i \ge 2$ we have

$$\boldsymbol{\alpha_i} = \boldsymbol{\alpha_2} \boldsymbol{R}^{i-2}, \tag{4a}$$

where $\boldsymbol{R}$ is the minimal non-negative solution of the quadratic matrix equation [8]

$$0 = \boldsymbol{F} + \boldsymbol{R}\boldsymbol{L} + \boldsymbol{R}^2 \boldsymbol{B}.$$

Matrix $\boldsymbol{R}$ can be determined using efficient numerical methods [8]. Based on (3) and using matrix $\boldsymbol{R}$, the irregular part of the steady state distribution is the solution of the linear system

$$\left(\boldsymbol{\alpha_0}\ \boldsymbol{\alpha_1}\ \boldsymbol{\alpha_2} \right) \begin{pmatrix} \boldsymbol{L_0} & \boldsymbol{F_0} & 0 \\ \boldsymbol{B}_1 & \boldsymbol{L_1} & \boldsymbol{F_1} \\ 0 & \boldsymbol{B}_2 & \boldsymbol{L} + \boldsymbol{R}\boldsymbol{B} \end{pmatrix} = 0, \tag{4b}$$

with normalization condition

$$\boldsymbol{\alpha_0}\mathbb{1} + \boldsymbol{\alpha_1}\mathbb{1} + \boldsymbol{\alpha_2}\left(\boldsymbol{I} - \boldsymbol{R}\right)^{-1}\mathbb{1} = 1. \tag{4c}$$

Using the steady state distribution (4c), the mean number of customers in the system can be expressed as

$$\begin{aligned}\mathrm{E}\left(n\right) &= \sum_{i=0}^{\infty} i\boldsymbol{\alpha_i}\mathbb{1} = \boldsymbol{\alpha_1}\mathbb{1} + \sum_{i=2}^{\infty} i\boldsymbol{\alpha_2}\boldsymbol{R}^{i-2}\mathbb{1} \qquad (5)\\ &= \boldsymbol{\alpha_1}\mathbb{1} + 2\boldsymbol{\alpha_2}\left(\boldsymbol{I} - \boldsymbol{R}\right)^{-1}\mathbb{1} + \boldsymbol{\alpha_2}\boldsymbol{R}\left(\boldsymbol{I} - \boldsymbol{R}\right)^{-2}\mathbb{1},\end{aligned}$$

and, applying Little's law, the mean system time can be calculated as

$$T = \frac{\mathrm{E}\left(n\right)}{\lambda}, \tag{6}$$

where λ is the expected value of the inter-arrival time. Based on the connection between T and $\mathrm{E}(n)$ it is clear that optimizing one is equivalent with optimizing the other. In the following we will use $\mathrm{E}(n)$ as objective function in the optimization.

These equations can be easily extended for the $N > 2$ case. Doing so we get

$$\mathrm{E}\left(n\right) = \sum_{i=1}^{N-1} i\boldsymbol{\alpha_i}\mathbb{1} + N\boldsymbol{\alpha_N}\left(\boldsymbol{I} - \boldsymbol{R}\right)^{-1}\mathbb{1} + \boldsymbol{\alpha_N}\boldsymbol{R}\left(\boldsymbol{I} - \boldsymbol{R}\right)^{-2}\mathbb{1}. \tag{7}$$

Equation (5) and (6) is relatively simple, however in the expression of the $\boldsymbol{\alpha_i}$ vectors terms including $p_{j,k,l}^{-1}$ and $(1 - p_{j,k,l})^{-1}$ will appear. This makes the straightforward optimisation a non-linear problem.

4 Finite state MDP formalization of the problem

In this section we present a finite state MDP formalization of the queueing system control problem. This formalization is based on the following observations:

- Decisions have to be made only on levels $0, \dots, N-2$.
- The objective function of the optimization is (7), which has a similar form to the objective function of the MDP (2) and contains only $\alpha_i,\ i = 1, \dots N$.

Using these our goal is to make an MDP for which (2) (the objective function of the MDP) is identical to (7) (the objective function of the optimization problem). To achieve this we use the following method.

In the first step we apply the simple transformation: $\boldsymbol{P} = \frac{1}{\gamma}\boldsymbol{Q} + \boldsymbol{I}$, where $\gamma = \max_{i,j} |\boldsymbol{Q}_{i,j}|$, i.e., the absolute value of the element of $\boldsymbol{Q}$ with the largest absolute value. This ensures that $\boldsymbol{P}$ is a valid DTMC transition matrix. Furthermore $\alpha\boldsymbol{Q} = 0$ (where α is the steady state probability vector of $\boldsymbol{Q}$), thus $\alpha\boldsymbol{P} = \alpha\frac{1}{\gamma}\boldsymbol{Q} + \alpha\boldsymbol{I} = \alpha$, consequently α is the steady state probability vector of the new DTMC as well. Note that this transformation is the same as the one used in randomization [7]. It is easy to see that this DTMC defines the S, A, P sets of an infinite state MDP, where decisions correspond to possible server choices upon arrival. For example in the two server example: $A = \{1, 2\}$, and if $i = (j-1) * n_s^2 + (k-1) * n_s + l$, then $p_{j,k,l} = 1$ if $\pi(i) = 1$ and $p_{j,k,l} = 0$ if $\pi(i) = 0$, in other words $\boldsymbol{P_1}$ is $\boldsymbol{P}$ with $p_{j,k,l} = 1$, $\forall j, k, l$ and $\boldsymbol{P_2}$ is $\boldsymbol{P}$ with $p_{j,k,l} = 0$, $\forall j, k, l$. Note that the resulting DTMC is a discrete time QBD with $\boldsymbol{B_i}' = \frac{1}{\gamma}\boldsymbol{B_i}$, $\boldsymbol{F_i}' = \frac{1}{\gamma}\boldsymbol{F_i}$ and $\boldsymbol{L_i}' = \frac{1}{\gamma}\boldsymbol{L_i} + \boldsymbol{I}$ for all i, where $\boldsymbol{M_i}'$ denotes the discrete time pair of the continuous time QBD's $\boldsymbol{M_i}$ matrix.

In the second step we change the infinite state regular part to a finite set of states and transitions while keeping the steady state probabilities of the irregular part unaffected. This step will be discussed in more detail shortly.

In the third step we assign the costs to the states based on objective function of the original optimization problem (7). This assignment is fairly straightforward. The cost of state i of level k is

$$C_{(k,i)} = \begin{cases} k, & \text{for } k = 0, 1, \ldots, N-1, \\ \underline{e_i}^T \left(N \left(\boldsymbol{I} - \boldsymbol{R}\right)^{-1} + \boldsymbol{R} \left(\boldsymbol{I} - \boldsymbol{R}\right)^{-2} \right) \mathbb{1}, & \text{for } k = N, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

We would like to stress that the dynamic behaviour of the MDP, i.e., the cost collected in a state of the MDP, does not have to be proportional to the waiting time accumulated in the corresponding state of the CTMC. The only important thing is that the irregular part of the MDP has the same steady state probabilities as the irregular part of original CTMC. Consequently, using the appropriate costs, the MDP has the same objective function - thus the same optimum - as the original problem.

The main question of the above procedure is how to carry out step two, i.e., how to substitute the infinite state regular part of the DTMC so that the steady state probabilities of the irregular part remain the same. For this we need to introduce two new matrices, $\boldsymbol{G}$ and $\boldsymbol{H}$. $\boldsymbol{G}_{i,j}$ shows the probability that, starting from (k, i), $k > N$ we reach $k-1$ and the first time this occurs we arrive in $(k-1, j)$, i.e.,

$$\boldsymbol{G}_{i,j} = Pr(\tau < \infty,\ X(\tau) = (k-1, j) | X(0) = (k, i)), \quad (9)$$

where τ is the time of the first arrival to level $k-1$. $\boldsymbol{H}_{i,j}$ is the the expected time (number of steps) of reaching level $k-1$ $(k > N)$ if we start in (k, i) supposing we arrive in phase $(k-1, j)$, multiplied by $\boldsymbol{G}_{i,j}$, i.e.,

$$\boldsymbol{H}_{i,j} = \mathrm{E}[\tau I_{X(\tau)=(k-1,j)} | X(0) = (k, i)], \quad (10)$$

where again τ is the time of the first arrival to level $k-1$ and I is the indicator function. It can be shown that $\boldsymbol{G}$ and $\boldsymbol{H}$ are the solutions of

$$\boldsymbol{B'} + (\boldsymbol{L'} - \boldsymbol{I})\boldsymbol{G} + \boldsymbol{F'}\boldsymbol{G}^2 = 0 \quad (11)$$

and

$$\boldsymbol{G} + (\boldsymbol{L'} - \boldsymbol{I})\boldsymbol{H} + \boldsymbol{F'}\boldsymbol{G}\boldsymbol{H} + \boldsymbol{F'}\boldsymbol{H}\boldsymbol{G} = 0, \tag{12}$$

respectively. Equation (11) can be solved numerically using efficient numerical methods, while (12) is a Sylvester equation, which is linear in the elements of $\boldsymbol{H}$ thus can be solved analytically if $\boldsymbol{G}$ is known. More details, including the derivation of the equations and the applicable numerical procedures for (11) can be found in [8].

Using $\boldsymbol{G}$ and $\boldsymbol{H}$ a finite state equivalent of the infinite QBD can be given. The irregular part of the DTMC consists of the 0th to Nth level. These are left unchanged during the transformation process. The regular part is substituted by M^2 states, where M is the size of one level. The probability of the event that the process, starting from (N, i) goes up to any phase of level $N+1$ in the next step and reaches level N again in phase j for the first time (after possibly multiple transitions on higher levels) is

$$b_{i,j} = \sum_{k=1}^{M} \boldsymbol{F'}_{i,k}\boldsymbol{G}_{k,j}. \tag{13}$$

Let the random variable $\tau_{i,j}$ be $\tau_{i,j} = \left(\tau I_{X(\tau)=(k-1,j)} | X(0) = (k,i)\right) / \boldsymbol{G}_{i,j}$. Note that $\mathrm{E}[\tau_{i,j}] = \frac{\boldsymbol{H}_{i,j}}{\boldsymbol{G}_{i,j}}$. For solving the optimization problem we only have to know the steady state probabilities of states on the irregular levels, therefore we do not need to distinguish states of the regular levels, thus we can modify the system the following way. From level $0, \ldots, N$ to level $0, \ldots, N$ transitions happen as before. We substitute the regular part (levels $N+1$ and above) of the DTMC with a level of M^2 states denoted by $s_{i,j},\ i,j = 1, \ldots, M$. From state (N, i) transition to state $s_{i,j}$ happens with $b_{i,j}$ probability. If the process reaches $s_{i,j}$ it transitions to (N, j) after staying in $s_{i,j}$ for $\tau_{i,j}$ time. This structure has the following interpretation. The instant the process would enter the regular part of the DTMC we determine the first state it arrives to upon first reaching the irregular part again. Instead of moving on to a state of the regular part the process moves to an intermediate state where it stays for the random time which is the same as the time needed in the original DTMC to go back to level N conditional on the fact that it arrives to state (N, j). It is clear, that the substitution does not make a difference from the irregular part's point of view. This new structure, however, is not Markovian as the distributions of transition times from $s_{i,j}$ are not memoryless. Processes where transition probabilities are according to a transition matrix, but transition times may have a general distribution are called semi-Markov processes. For semi-Markov processes the steady state distribution depends only on the expected value of the transition times. For proof see e.g. [3]. Thus, without affecting the steady state probabilities of the irregular part we can change $\tau_{i,j}$ to the geometrically distributed $\tau'_{i,j}$ if $\mathrm{E}[\tau'_{i,j}] = \mathrm{E}[\tau_{i,j}] = \frac{\boldsymbol{H}_{i,j}}{\boldsymbol{G}_{i,j}}$. This geometrical distribution can be achieved using a feedback in $s_{i,j}$ with probability $q_{i,j}$ and transition to (N, j) with probability $1 - q_{i,j}$, where $q_{i,j} = \frac{\boldsymbol{H}_{i,j} - \boldsymbol{G}_{i,j}}{\boldsymbol{H}_{i,j}}$. Now the modified system is a DTMC and its irregular part has the same steady state probability distribution as the original CTMC. In the following we will denote the transition matrix of this DTMC by $\boldsymbol{P^*}$. Matrix $\boldsymbol{P^*}$ coupled with the previously defined costs and actions form a finite state MDP that can be optimized using standard methods (e.g. linear programming, value iteration, policy iteration) and has the same optimal strategy and optimal cost as the original infinite continuous time system.

Using the description of Section 3, the size of level $0, \ldots,\ N$ is $n_a n_s^N$ each, while the size of level $N+1$ is $(n_a n_s^N)^2$ consequently the size of $\boldsymbol{P^*}$ is $N n_a n_s^N + (n_a n_s^N)^2$. To improve the speed of the optimization this size has to be reduced, which can be done using two methods. The first one is the reduction of the original QBD, the second one is the reduction of the part used for substituting the regular part of the QBD.

We will not discuss the first method in detail just present its basic idea. It is easy to see, that the labelling of the servers is arbitrary, i.e., while in the description in Section 3 the phase of every server is followed individually, it is enough to keep track of the number of empty and busy servers at each point (and the phase of the arrival process of course). Consequently the irregular levels (where the buffer is empty) can be described using a set of $2n_s+1$ numbers. The first element of the set indicates the phase of the arrival and can be between 1 and n_s, the next n_s elements show the number of empty servers in each of the service phases, the final n_s elements show the number of occupied servers in each of the service phases. As there are a total of N servers, the sum of the last $2n_s$ elements of the set is N. E.g. if the queueing system has three servers ($N=3$), the service process is described by a size 3 MAP and the arrival process by a size 4 MAP, then $(4|0,1,1|0,0,1)$ denotes the state where the arrival process is in phase 4, there are two empty servers, one in phase 2 one in phase 3, and there is one working server in phase 3. It can be easily seen that there are a total of $n_a \left(\!\binom{2n_s}{N}\!\right)$ different configurations for the set, where $\left(\!\binom{n}{k}\!\right) = \binom{n+k-1}{k}$ is the k combination of n with repetition. Using the above idea a more efficient QBD description can be constructed where each state of the irregular part corresponds to a specific configuration of the set. This construction is done by combining multiple equivalent states into one, which is called lumping and is a standard method for state space reduction. The previously described method of making a finite state MDP from the infinite QBD can still be applied without any changes. Using the same thought process the finite state substitute of the regular part can be reduced to $\left(n_a \left(\!\binom{n_s}{N}\!\right)\right)^2$ states.

The second improvement can be made by realizing that there are a few constraints that the regular part of the original CTMC and its substitute has to satisfy. If these requirements are met, the size of the substituted part and its exact structure are not important. These constraints come from $\alpha\boldsymbol{Q} = 0$, $\sum_{i=0}^{N-1} \boldsymbol{\alpha_i}\mathbb{1} + \boldsymbol{\alpha_N}\left(\boldsymbol{I}-\boldsymbol{R}\right)^{-1}\mathbb{1} = 1$ and the general Markovian constraints of a CTMC. For simplicity's sake we discuss the $N=2$ case, from which the general case can be easily derived. Let us consider the modified CTMC with generator

$\hat{\boldsymbol{Q}}$ that has the following structure

$$\hat{\boldsymbol{Q}} = \begin{pmatrix} \boldsymbol{L}_0 & \boldsymbol{F}_0 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B}_1 & \boldsymbol{L}_1 & \boldsymbol{F}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_2 & \boldsymbol{L} & \boldsymbol{F}^* \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}^* & \boldsymbol{L}^* \end{pmatrix}. \tag{14}$$

We have to keep the steady state probabilities of level $0, 1, 2$ the same, but the new part can have a different α_3^*, $\alpha_3^* \neq \alpha_3$ steady state probability vector. First of all, from $\alpha^* \mathbb{1} = \alpha \mathbb{1} = 1$ using $\alpha_3 = \alpha_2 \boldsymbol{R}$ we get

$$\alpha_3^* \mathbb{1} = \alpha_2 \boldsymbol{R} (\boldsymbol{I} - \boldsymbol{R})^{-1} \mathbb{1}. \tag{15}$$

From $\alpha \boldsymbol{Q} = 0$ we have $\alpha_1 \boldsymbol{F_1} + \alpha_2 \boldsymbol{L} + \alpha_3 \boldsymbol{B} = 0$ and from $\alpha^* \hat{\boldsymbol{Q}} = 0$ we have $\alpha_1 \boldsymbol{F_1} + \alpha_2 \boldsymbol{L} + \alpha_3^* \boldsymbol{B^*} = 0$. By subtracting these equations from each other and using $\alpha_i = \alpha_2 \boldsymbol{R}^{i-2}$ we get

$$\alpha_3^* \boldsymbol{B^*} = \alpha_2 \boldsymbol{R} \boldsymbol{B}. \tag{16}$$

Furthermore from $\alpha^* \hat{\boldsymbol{Q}} = 0$ and $\hat{\boldsymbol{Q}} \mathbb{1} = 0$ we get

$$\alpha_2 \boldsymbol{F^*} + \alpha_3^* \boldsymbol{L^*} = 0, \tag{17}$$

$$(\boldsymbol{B_2} \boldsymbol{L} + \boldsymbol{F^*}) \mathbb{1} = 0, \tag{18}$$

$$(\boldsymbol{B^*} + \boldsymbol{L^*}) \mathbb{1} = 0. \tag{19}$$

Finally, we have the standard sign constraints, i.e. all elements of $\boldsymbol{B^*}$, $\boldsymbol{L^*}$, $\boldsymbol{F^*}$ are non-negative, except for the diagonal of $\boldsymbol{L^*}$ which is strictly negative.

These constraints give a constrained linear equation system if we first tie α_3^*. We can assign values to α_3^* randomly or e.g. make all the elements equal. This system can be solved using linear programming if a solution does exist. In general it is not guaranteed that a constrained linear equation system has a solution, however the method presented previously in this section gives one where the size of $\boldsymbol{L^*}$ is $M^2 \times M^2$ if M is the size of the last irregular level. Thus one possible method is to start by trying to find a solution to the above equations with an $\boldsymbol{L^*}$ of size $n \times n$, $n = 1$ (the size of $\boldsymbol{B^*}$ and $\boldsymbol{F^*}$ are $n \times 1$ and $1 \times n$ respectively) and increase the size until the system can be solved or $n = M^2$ is reached. After this the optimization problem can be solved (after transforming the CTMC with generator $\hat{\boldsymbol{Q}}$ into a DTMC) using standard MDP methods.

5 Linear programming solution of the problem

In this section we give a method for solving the queueing system control optimization problem using linear programming (LP). First we mention that linear programming is one of the classical ways to optimize a finite state MDP. As such, the last step of the optimization after the transformation of the infinite state MDP to a finite state one could be using LP to find the optimal strategy. In this section, however, we make use of the flexibility of LP to describe the problem without forming a finite MDP, although the description will be based on the LP formalization of MDP optimization. Consequently we first present the LP formalization of the general average cost MDP problem. We will follow the same thought process as in [1] with different notations. After that we make the necessary customization for the problem at hand.

Let us take an MDP with the previously introduced (S, A, P, C) notation, but consider also mixed strategies. In case of a mixed strategy decision $a \in A$ is made at state $i \in S$ with probability $u_{i,a}$, i.e., $u(i,a) = Pr(a_k = a | X(k) = i)$, with $\sum_{a \in A} u(i,a) = 1,\ \forall i \in S$. The goal is to optimize $u(i,a)$ for all $i \in S,\ a \in A$ according to the given objective function. Let us define $\boldsymbol{U}$ such that $\boldsymbol{U}_{i,a} = u(i,a),\ \forall\ i \in S$ and $a \in A$ and $\boldsymbol{P_u}$ such that $\boldsymbol{P_u}_{i,j} = \sum_{a \in A} u(i,a) \boldsymbol{P_a}_{i,j}$. Now the objective function in (2) changes to $\sum_{i \in S} \alpha(u)_i \sum_{a \in A} \boldsymbol{U}_{i,a} C_a(i)$. From these the optimization problem can be given in a form that is similar to the standard LP form as

$$\begin{aligned} \min \sum_{i \in S} \alpha_i \sum_{a \in A} U_{i,a} C_a(i), \\ \text{s.t. } \alpha(\boldsymbol{P_u} - \boldsymbol{I}) = 0, \\ \boldsymbol{U}\mathbb{1} = \mathbb{1}, \end{aligned} \tag{20}$$

where the variables are the elements of α and $\boldsymbol{U}$. This problem is non-linear because $\boldsymbol{P_u}$ depends on the elements of $\boldsymbol{U}$, thus the products of the elements of α and $\boldsymbol{U}$ appear in the constraints, however, it can be linearised by introducing new variables $x_{i,a} = \alpha_i u_{i,a}$. That is, $x_{i,a}$ is the steady state probability that the process is in state i and decision a is made. Let us introduce matrix $\boldsymbol{X}$ with $\boldsymbol{X}_{i,a} = x_{i,a},\ \forall\ i \in S$ and $a \in A$ and denote its ith column by $\boldsymbol{X}_{*,i}$. Notice that $(\boldsymbol{X}\mathbb{1})^T = \alpha$. As such the optimization problem can now be defined as an LP as

$$\begin{aligned} \min \sum_{i \in S} \sum_{a \in A} \boldsymbol{X}_{i,a} C_a(i), \\ \text{s.t. } \sum_{a \in A} \boldsymbol{X}^T_{*,a} \boldsymbol{P_a} - (\boldsymbol{X}\mathbb{1})^T = \underline{0}, \\ \sum_{i \in S} \sum_{a \in A} \boldsymbol{X}_{i,a} = 1, \\ \boldsymbol{X}_{i,a} \geq 0, \quad \forall i \in S,\ \forall a \in A. \end{aligned} \tag{21}$$

Here $\boldsymbol{X}_{i,a} \boldsymbol{P_a}_{i,j}$ is the steady state probability that the process is in state i, decision a is made, and as a result the process transitions to state j.

As mentioned before, the above introduced LP could be used to solve the finite state MDP of Section 3. Instead we make use of the fact that the LP optimization is a general purpose tool unlike methods that are particularly developed to solve MDPs, e.g. value and policy iteration. The MAP/MAP/N optimization problem can be formalized using LP by noticing that equations (4b) and (4c) (more precisely their N server generalizations) are sufficient constraints to solve the optimization problem. To be consistent with the previous discrete time description we use the discrete time counterpart of (4c), i.e., $\boldsymbol{P} = \frac{1}{\gamma}\boldsymbol{Q} + \boldsymbol{I}$ just as in Section 3. Thus, for the $N = 2$ case for example

$$\alpha = \begin{pmatrix}\boldsymbol{\alpha_0} \; \boldsymbol{\alpha_1} \; \boldsymbol{\alpha_2}\end{pmatrix}, \tag{22}$$

$$\boldsymbol{P} = \frac{1}{\gamma}\begin{pmatrix} \boldsymbol{L_0} & \boldsymbol{F_0} & 0 \\ \boldsymbol{B'_1} & \boldsymbol{L'_1} & \boldsymbol{F'_1} \\ 0 & \boldsymbol{B'_2} & \boldsymbol{L'} + \boldsymbol{RB'} \end{pmatrix} + \boldsymbol{I}. \tag{23}$$

These can be substituted into (21). The only difference is in the last normalization constraint. This changes to $\boldsymbol{\alpha_0}\mathbb{1} + \boldsymbol{\alpha_1}\mathbb{1} + \boldsymbol{\alpha_2}\left(\boldsymbol{I} - \boldsymbol{R}\right)^{-1}\mathbb{1} = 1$ according to (4c). Finally the cost vector is the same as in Section 3, thus, when the index of the last state of level i is k_i, the LP problem is

$$\begin{aligned}
&\min \sum_{i\in S}\sum_{a\in A}\boldsymbol{X}_{i,a}C_a(i),\\
&\text{s.t.} \sum_{a\in A}\boldsymbol{X}^T_{*,a}\boldsymbol{P_a} - (\boldsymbol{X}\mathbb{1})^T = \underline{0},\\
&\sum_{i=1}^{k_1}\sum_{a\in A}\boldsymbol{X}_{i,a} + \sum_{i=k_1+1}^{k_2}\sum_{j=k_1+1}^{k_2}\sum_{a\in A}\boldsymbol{X}_{i,a}(\boldsymbol{I}-\boldsymbol{R})^{-1}_{i-k_1,j-k_1} = 1,\\
&\boldsymbol{X}_{i,a} \geq 0, \quad \forall i \in S,\ \forall a \in A.
\end{aligned} \tag{24}$$

As in the previous section $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ can be obtained by using substitution $p_{j,k,l} = 1$ and $p_{j,k,l} = 0$ respectively $\forall j, k, l$. If, for a given $i \in S$ and $a_1, a_2,\ a_1 \neq a_2$ we get $u_{i,a_1}, u_{i,a_2} > 0$, then the optimization gives a mixed strategy as optimum. In that case choosing either of the decisions with 1 probability gives the same optimum. The significance of enabling mixed strategies is that it makes the optimization an ordinary LP problem instead of the integer LP problem that results from considering only pure strategies.

6 Numerical experiments

6.1 Computational complexity

The computational complexity of building up the QBD, finding its $\boldsymbol{G}$ and $\boldsymbol{H}$ matrices and transforming the problem to finite state is negligible compared to the one of the solution of the resulting MDP or LP problem, therefore we only consider complexity of the solution the finite MDP and LP.

The complexity of the basic MDP solution methods is summarized in [9]. Let $|S|$ be the number of states and $|A|$ be the number of possible decisions. We upper bound the number of decisions by $|A| = n_s$, which means that there is an empty server in every phase for all the states of the MDP. Policy iteration has a complexity $O(|S|^3)$ per step and requires $O(|A|)$ steps in the average case.

As seen in Section 4 $|S| = n_a \left(\binom{2n_s}{N}\right) + \left(n_a \left(\binom{n_s}{N}\right)\right)^2$ - or less, if the substitute of the regular part can be further reduced. The second term of the expression is usually higher than the first one, for the computationally tractable cases the difference is $0-2$ orders of magnitude.

For the LP approach the number of variables is $|X| = n_a \left(\binom{2n_s}{N}\right) n_s$ and the number of constraints (from (24)) is $n_c = |X| + n_a \left(\binom{2n_s}{N}\right) + 1$. There is a vast number of algorithms for solving LP problems. The simplest and most frequently used are probably the simplex and the revised simplex methods. For the latter the average computational cost is $O(|X|^3)$ (see e.g. [10]). Using more involved methods this cost can be reduced. See [4] for example for a comprehensive summary.

In general the two methods have similar computational costs. The MDP approach is better if the substitute of the regular part can be reduced, while the LP approach is more efficient if this is not possible and a more involved LP solver is used.

6.2 Numerical examples

M/MAP(2)/2 systems First we discuss the simplest interesting case, the M/MAP(2)/2 queue, in which arrivals happen according to a Poisson process, and the service is carried out by two servers that have the same order 2 MAP service time. The first part of this segment is the reiteration of the results of the corresponding section in [2].

In the M/MAP(2)/2 queue there is one simple question to be answered: If both servers are idle, one of them is in phase 1 and the other one is in phase 2, which server has to process the next arriving customer to have a minimal average system time? In other words, what is the optimal value of $p_{1,1,2}$? It seems natural, and from the first part of the paper we already know, that a pure strategy is optimal, i.e., $p_{1,1,2}$ is either 1 or 0. The intuitive answer is to choose the server which can serve the customer faster. This means that we compare the mean service time starting from phase 1 and phase 2, i.e., $\underline{e_1}^T(-\boldsymbol{S_0})^{-1}\mathbb{1}$ and $\underline{e_2}^T(-\boldsymbol{S_0})^{-1}\mathbb{1}$, and if the first expression is smaller, we choose the server in phase 1 ($p_{1,1,2} = 1$), otherwise the one in phase 2 ($p_{1,1,2} = 0$).

This greedy decision can be motivated by the fact that we would like to serve the customer as fast as possible to have an idle queue as soon as possible. For the examined system, however, the numerical results show that the opposite choice is better, i.e., it is better to choose the server which serves the customer slower. This counter-intuitive result can be interpreted the following way. If we use the faster server for the first customer, the probability of finishing the service before a new arrival is high, as the mean service time of the faster state is smaller than the mean inter-arrival time of a new customer. Upon service there is a chance that the server moves to the slower state, leaving the system with two servers in the phase with higher service time. In this state there is a higher chance that more than 2 consecutive customers arrive before the first customer can be served, which leads to a higher average system time. In other words, assigning the customer with the faster server leads to a more deteriorated state after service completion, while assigning the customer with the server in the slower phase, there is a chance that the server will move to the faster state upon service, thus the state of the system improves. One can think of this effect as the repair of the server at the cost of a slower service. Our extensive numerical investigations suggest that choosing the server with higher service time is optimal for MAP(2) servers regardless of their other characteristics and the intensity of arrivals.

A natural question to ask is what does the possible gain, i.e., the magnitude of the difference between the worst and best strategies, depend on. In other words, how can we characterize systems where the best strategy is significantly better then the worst. In the following we denote by g_a the absolute value of the maximum gain, and by g_r its ratio to the cost of the best strategy. That is, $g_a = \mathrm{E}_{\pi}^{-}(n) - \mathrm{E}_{\pi}^{+}(n)$, $g_r = \frac{\mathrm{E}_{\pi}^{-}(n) - \mathrm{E}_{\pi}^{+}(n)}{\mathrm{E}_{\pi}^{+}(n)}$, where $\mathrm{E}_{\pi}(n)$ is the mean number of customers for policy π, and π^{+} and π^{-} are the optimal and worst strategy respectively. According to our experiments, classic statistical measures of the service MAP such as its autocorrelation or the moments of its marginal distribution cannot be used to characterize the gain in the general case. This can be understood by knowing that different MAPs can have the same exact statistical properties. We have found that it is best to consider the following simple characteristics:

- The ratio of the mean arrival (λ) and service rate (μ (see Section 2.1 for computation)), $r_1 = \frac{\lambda}{\mu}$. (To have a stable queue $r_1 < N$ has to hold.)
- The ratio of mean service times of the server starting from the different phases of the MAP, i.e. $r_2 = \frac{\max(m_1,m_2)}{\min(m_1,m_2)}$, with $m_1 = \underline{e_1}^T(-\boldsymbol{S_0})^{-1}\mathbb{1}$, $m_2 = \underline{e_2}^T(-\boldsymbol{S_0})^{-1}\mathbb{1}$.
- The steady state probability vector of the phases embedded to the arrivals of the service MAP, $\underline{v} = \{v_1, v_2\}$ (see Section 2.1).

Intuitively the higher r_1, the bigger the difference is. This is true for the absolute gain g_a but not for the relative gain g_r. Figure 1 shows the relative gain versus the arrival intensity for the M/MAP(2)/2 queue with service MAP

$$\boldsymbol{S}_0 = \begin{pmatrix} -1/10 & 1/20 \\ 0 & -100 \end{pmatrix}, \quad \boldsymbol{S}_1 = \begin{pmatrix} 1/20 & 0 \\ 5 & 95 \end{pmatrix}, \tag{25}$$

It can be seen that g_r has a maximum around $\lambda \approx 0.44$. Based on our experiments this behaviour is typical, i.e., the relative gain is the highest for medium load. Intuitively, the higher r_2, the bigger the g_r is, as there is a more significant difference between the possible decisions. This time the intuition is correct. Finally we found that g_r can get higher if the MAP is more "balanced", i.e. both elements of the $\underline{v}$ embedded probability vector are high enough.

The above observations for the M/MAP(2)/2 case are reflected in Figure 2, where 1000 service MAPs with a given mean and completely random elements were taken and their relative gains g_r were plotted against flexibility f that represents the randomness due to the embedded stationary vector and the difference of the mean service times starting from different initial states as

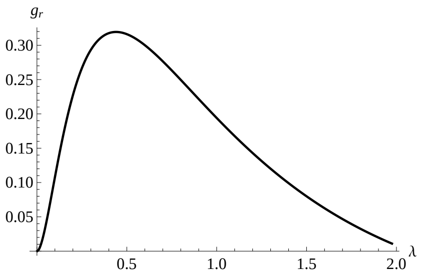

Fig. 1. Relative gain as the function of arrival intensity for a given service process

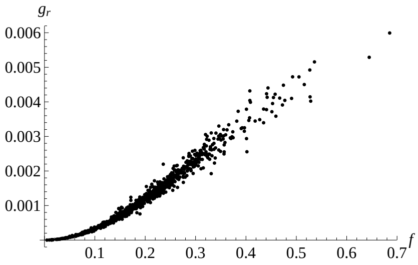

Fig. 2. Relative gain in mean system time as the function of flexibility f

$f(r_2, \underline{v}) = \sqrt{(r_2 - 1) * v_1 * v_2}$. The correlation between f and g_r was ≈ 0.98. We examined more special types of random MAPs as well, the correlation between f and g_r was always > 0.9.

More complex systems Intuitive understanding of the optimal control of more complex systems becomes increasingly hard. Again we refer back to [2], where it was demonstrated that for M/MAP(3)/2 queues already the optimal strategy cannot be explained as simply as for the M/MAP(2)/2 case. For example let us take the service MAP with

$$\boldsymbol{S}_0 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -2.3 & 0 \\ 0 & 0 & -100 \end{pmatrix}, \quad \boldsymbol{S}_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 2.3 \\ 100 & 0 & 0 \end{pmatrix}.$$

For $\lambda = 1.5$ the optimal strategy is to always prioritize the server in phase 1 and choose the server in phase 3 over the one in phase 2. For $\lambda = 1.2$ the priority of phase 2 and phase 3 are swapped. The $r_1, r_2, \underline{v}$ factors introduced for the M/MAP(2)/2 queue can still be used to roughly evaluate the system.

For $N > 2$ using simple intuitive rules gets even harder. In these cases the optimal server choice can even depend on the phase of the servers that are occupied at the time of a new arrival. For example let us consider an M/MAP(3)/3 queue with arrival intensity $\lambda = 3.6$ and service MAP

$$\boldsymbol{S}_0 = \begin{pmatrix} -0.76 & 0 & 0 \\ 0 & -10 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad \boldsymbol{S}_1 = \begin{pmatrix} 0 & 0.76 & 0 \\ 0 & 0 & 10 \\ 1 & 0 & 0 \end{pmatrix}.$$

If the queue is on level 1, the empty servers are in phase 1 and 3 and the occupied server is in phase 1, the optimal decision is to process the new customer by the server in phase 3. However if the occupied server is in phase 3, the optimal choice is the server in phase 1.

Finally, we have to stress that the intuitive explanations of the optimal control are only conjectures based on numerical experiments, and that in spite of the relatively efficient optimization techniques the proposed methods can only be used for relatively small systems (depending on the size service and arrival MAPs for $N = 2, \ldots,\ 10$ servers) due to the multiplicative increase of the state space.

Ackonwledgement

The authors gratefully acknowledge the support of the TÁMOP-4.2.2C-11/1/KONV-2012-0001 and the OTKA K101150 projects.

7 Conclusion

In this paper we presented two procedures for finding the optimal policy in MAP/MAP/N queues. Both procedures are based on the matrix analytic methods, which make an efficient treatment possible. Using these procedures we demonstrated some of the characteristics of the MAP/MAP/N systems. We showed that even the simplest queues have counter intuitive behaviour and illustrated the lack of simple intuitive rules in case of more complex systems (e.g. $N \geq 3$), which makes the use of a computational approach is necessary.

References

1. D. Bello and G. Riano. Linear programming solvers for Markov decision processes. In *IEEE Systems and Information Engineering Design Symposium*, pages 90–95, 2006.
2. L. Bodrog, M. Gribaudo, G. Horváth, A. Mészáros, and M. Telek. Control of queues with map servers: experimental results. In *8th International Conference on Matrix-Analytic Methods in Stochastic Models*, 2014.
3. E. Çinlar. Exceptional paper-Markov renewal theory: A survey. *Management Science*, 21(7):727–752, 1975.
4. G. B. Dantzig and M. N. Thapa. *Linear programming 2: Theory and Extensions.* Springer-Verlag, 2003.
5. D. Efrosinin. *Controlled Queueing Systems with Heterogeneous Servers.* PhD thesis, University of Trier, 2004.
6. J. Filar and K. Vrieze. *Competitive Markov Decision Processes.* Springer, 1997.
7. D. Gross and D. R. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research*, 32(2):343–361, 1984.
8. G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling.* ASA-SIAM Series on Statistics and Applied Probability. Society for Industrial and Applied Mathematics, 1999.
9. M. L. Littman, T. L. Dean, and L. P. Kaelbling. On the complexity of solving Markov decision problems. In *Proc. of 11th conf. on Uncertainty in artificial intelligence*, pages 394–402. Morgan Kaufmann Publishers Inc., 1995.
10. V. Pan. On the complexity of a pivot step of the revised simplex algorithm. *Computers & mathematics with applications*, 11(11):1127–1140, 1985.
11. S. Stidham Jr and R. Weber. A survey of Markov decision models for control of networks of queues. *Queueing Systems*, 13(1-3):291–314, 1993.