Infinite Markov decision processes with decision independent subset of states[*]

András Mészáros
MTA-BME Information Systems Research Group
1117, Magyar Tudosok krt. 2, Budapest, Hungary
meszarosa@hit.bme.hu

Miklós Telek
Dept. of Networked Systems and Services
Budapest University of Technology and Economics
1117, Magyar Tudosok krt. 2, Budapest, Hungary
telek@hit.bme.hu

October 3, 2018

Abstract

In this work, we examine Markov Decision Processes (MDPs) that are composed of a finite subset of states with decisions and a (potentially infinite) subset of states without decisions. We show that, if some parameters of these MDPs can be computed efficiently, then a transformation to a condensed representation is possible, in which only the states with decisions are kept, and the rewards and transition rates are modified such that the optimal policy is the same in the original and the modified MDP.

When the subset of states without a decision is infinite, the analysis is based on some structural regularity of the process. Two practically important structures are considered the M/G/1-type and the G/M/1-type.

Keywords: Markov decision process, equivalent representation, state space reduction, Markov chain with regular structure.

[*] This project is partially sponsored by the OTKA K123914 project.

1 Introduction

Markov decision processes (MDPs) give a powerful yet simple tool to formalize and solve decision problems and are commonly used in a wide variety of fields from machine learning [11] through telecommunication [2] to finance [3]. One of the biggest issues when using MDPs is known as state space explosion, that is, the number of states of the MDP usually grows exponentially with the number of variables of the analysed system. Because of this, the number of states in the MDP can easily increase to a point where the classical MDP solving methods cannot be used. To overcome this problem, some techniques have been developed. One possibility is to prove that the optimal policy in the MDP has threshold form, which means that the optimal decision in any state can be determined based on whether a certain parameter is above a fixed value (threshold). In this case, finding the threshold is enough to solve the optimization problem. For instance, accepting requests to a queue may be optimal until the queue length reaches a certain value. See e.g. [12] or [8] for more examples. Another noteworthy approach is presented in [6], where more efficient solution methods for MDPs with factored representations are considered. These cannot be applied, however for MDPs that are infinite or cannot be factorized.

Apart from exact optimal solutions, one can get a quasi-optimal solution by using certain approximation techniques. One possible approach is the truncation of the state space. This may happen based on the physical model (e.g. the size of the buffer is constrained), as in [17] and [10] for example. Alternatively, one can truncate based on only mathematical considerations as discussed by [1]. Another interesting method is presented in [16], where a so-called deterministic simulative model is introduced. The essence of this model is that the original MDP is transformed in such a way, that transitions of the new model all become deterministic.

In this work, we propose an efficient reduction method that can be used for MDPs which are composed of a finite subset of states with decisions and a finite or infinite subset of states without decisions. More specifically we give a method that compresses the MDP to the size of the subset with decisions. The presented method requires that some important parameters of the MDP can be calculated efficiently. We also show how to use the presented reduction method for infinite state MDPs with QBD, M/G/1-type and G/M/1-type structures, which are the most prevalent classes of infinite MDPs in queueing problems.

The problem of reducing the size of the state space of such systems has been considered in [14], but the solution proposed there was suboptimal in the sense that the reduced state space was much larger than the set of states with a decision. The solution proposed here is optimal in this sense.

The rest of the paper is organized as follows. The paper starts with an example in Section 2 to motivate the forthcoming analysis. Section 3 provides a summary of MDPs. The parameters of the MDP compression method are provided in Section 4, while the compression method is stated and proved in Section 5. The special cases of infinite MDPs with M/G/1 type and G/M/1 type structures are provided in Section 6, whose reward parameters a computed in Section 7.

2 A motivating example

The problem considered in this paper has a strong practical motivation which is detailed in [5, 14]. Here we summarize the problem and the related model for completeness.

In case of multi-server systems with identical but state dependent servers, it is an interesting optimization problem to properly assign new incoming jobs with one of the free servers, if more than one server is idle at customer arrival. Consider a simple queueing systems with multiple MAP servers, where the incoming customers can be freely assigned with service unit in case of more than one available free servers.

In particular, [14] considers the optimal control of MAP/MAP/n queues. The simplest version of such models is the M/MAP/2 queue where customers arrive according to a Poisson process with rate λ, the service process of each server is a MAP with two states characterized by the matrix pair $(\boldsymbol{S_0}, \boldsymbol{S_1})$. The associated state transition structure is

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

where

$$\boldsymbol{L}_0 = -\lambda \boldsymbol{I} \otimes \boldsymbol{I}, \boldsymbol{F}_0(a) = \lambda \left((\boldsymbol{I} \otimes \boldsymbol{I})\, \boldsymbol{P}(a), \quad (\boldsymbol{I} \otimes \boldsymbol{I})\, (\boldsymbol{I} - \boldsymbol{P}(a)) \right),$$

$$\boldsymbol{B}_1 = \begin{pmatrix} \boldsymbol{I} \otimes \boldsymbol{S}_1 \\ \boldsymbol{S}_1 \otimes \boldsymbol{I} \end{pmatrix}, \boldsymbol{L}_1 = \begin{pmatrix} -\lambda \boldsymbol{I} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{S}_0 & 0 \\ 0 & -\lambda \boldsymbol{I} \otimes \boldsymbol{I} + \boldsymbol{S}_0 \otimes \boldsymbol{I} \end{pmatrix},$$

$$\boldsymbol{F}_1 = \lambda \begin{pmatrix} \boldsymbol{I} \otimes \boldsymbol{I} \\ \boldsymbol{I} \otimes \boldsymbol{I} \end{pmatrix}, \boldsymbol{B}_2 = \begin{pmatrix} \boldsymbol{I} \otimes \boldsymbol{S}_1 \\ \boldsymbol{S}_1 \otimes \boldsymbol{I} \end{pmatrix}, \boldsymbol{L} = -\lambda \boldsymbol{S}_0 \oplus \boldsymbol{S}_0,$$

$$\boldsymbol{F} = \boldsymbol{A}_1 \otimes \boldsymbol{I} \otimes \boldsymbol{I}, \boldsymbol{B} = \boldsymbol{I} \otimes \boldsymbol{S}_1 + \boldsymbol{S}_1 \otimes \boldsymbol{I}\ .$$

and the matrix which is responsible for the decision upon customer arrival to the idle system is

$$\boldsymbol{P}(a_1) = \mathrm{diag}\,(1/2, 1, 0, 1/2) \quad \text{and} \quad \boldsymbol{P}(a_2) = \mathrm{diag}\,(1/2, 0, 1, 1/2)\,.$$

According to matrix $\boldsymbol{P}(a)$, at a customer arrival to the empty system the customer is directed to the server in phase 1 by choosing action a_1 and to the server in phase 2 by choosing action a_1, if the servers are in different phases. If idle servers are in the same phase, the service units are chosen evenly.

The associated reward matrix

$$\boldsymbol{C}(a) = \begin{pmatrix} \boldsymbol{I} & & & & \\ & \frac{1}{2}\boldsymbol{I} & & & \\ & & \boldsymbol{0} & & \\ & & & \boldsymbol{0} & \\ & & & & \ddots \end{pmatrix},$$

which is decision independent, intends to maximize the server idle time.

For curiosity, we note that the counter intuitive conclusion gained by the analysis of this model in [5] is that it worth to choose the slower server, because it results in a better system state for higher levels of system saturation.

3 Theoretical background

3.1 Markov Decision Processes

In the paper, we consider continuous time, time-homogeneous, non-discounted, MDPs with the following definition.

Definition 3.1. *Let $X(t)$ be a continuous time Markov chain with state space S, A a set of decisions, α_0 an initial probability vector, $\boldsymbol{Q}(a)$ a decision dependent generator matrix satisfying $\boldsymbol{Q}(a)\underline{\boldsymbol{1}} = 0$ for $\forall a \in A$ (where $\underline{\boldsymbol{1}}$ is the column vector of 1s with appropriate size), $\boldsymbol{C}(a)$ is a decision dependent diagonal reward rate matrix. We say that the tuple $(S, A, \alpha_0, \boldsymbol{Q}(a), \boldsymbol{C}(a))$ is a continuous time Markov decision process. The generator of the MDP, $\boldsymbol{Q}(a)$,*

For such MDPs the usual optimization problem is to find a policy (state-decision mapping) $\pi^*(s) \in \{\pi(s) : S \to A\}$ such that

$$\pi^* = \arg\max_{\pi} \mathrm{E}_\pi \left[\lim_{T\to\infty} \frac{1}{T} \int_{t=0}^{T} \boldsymbol{C}_{X(t),X(t)}(\pi(X(t))dt \right].$$

Throughout the paper we assume that for every policy the Markov model is composed of exactly one communicating block and potentially one transient block. In this case, the optimal policy can also be expressed as

$$\pi^* = \arg\max_{\pi} \alpha(\pi)\boldsymbol{C}(\pi)\underline{\boldsymbol{1}}, \tag{1}$$

where $\alpha(\pi)$ is the steady state probability vector for policy π, that satisfies,

$$\alpha(\pi)\boldsymbol{Q}(\pi) = \underline{\mathbf{0}}, \quad \alpha(\pi)\underline{\mathbf{1}} = 1,$$

where $\underline{\mathbf{0}}$ is the column vector of 0s of appropriate size [9].

The definition of MDPs does not constrain the sign of the elements of the $\boldsymbol{C}(a)$ reward matrix, however offsetting $\boldsymbol{C}(a)$ with a constant value c does not change the optimal policy since for any policy π

$$\alpha(\pi)\left(\boldsymbol{C}(\pi) + c\boldsymbol{I}\right)\underline{\mathbf{1}} = \alpha(\pi)\boldsymbol{C}(\pi)\underline{\mathbf{1}} + c,$$

that is, the optimal policy is the same for an MDP with reward matrix $\boldsymbol{C}(a)$ and $\boldsymbol{C'}(a) = \boldsymbol{C}(a) + c\boldsymbol{I}, \forall c \in \mathbb{R}$, therefore in the following we assume that $\min_{a,i} \boldsymbol{C}(a)_{ii} > 0$, that is, the reward rate in every state is positive for every decision.

3.2 Basic transformation of MDPs

Our main goal is to examine MDPs for which the state space S can be partitioned into two disjoint subsets S_U and S_D ($S_U \cup S_D = S$,$S_U \cap S_D = \emptyset$) where S_U is finite and contains all the states in which decisions can be made and S_D is potentially infinite and contains only states where decisions are not made, or decisions have the same effect (i.e., $\boldsymbol{Q}_{ij}(a_k) = \boldsymbol{Q}_{ij}(a_\ell), \forall i, j \in S_D, k, \ell \in A$). Without loss of generality we assume that the states in S_U have lower indexes than the states in S_D (i.e., $i < j, \forall i \in S_U, j \in S_D$), thus the $\boldsymbol{Q}(\pi)$ generator matrix and the $\boldsymbol{C}(\pi)$ reward-rate matrix have the following block structure

$$\boldsymbol{Q}(\pi) = \begin{pmatrix} \boldsymbol{Q_U}(\pi) & \boldsymbol{Q_{UD}}(\pi) \\ \boldsymbol{Q_{DU}} & \boldsymbol{Q_D} \end{pmatrix}, \quad \boldsymbol{C}(\pi) = \begin{pmatrix} \boldsymbol{C_U}(\pi) & 0 \\ 0 & \boldsymbol{C_D} \end{pmatrix}, \tag{2}$$

where π in the argument indicates that the respective part of the matrix depends on the actual policy. In the rest of the paper we assume that S_D is transient with finite sojourn time, consequently $\boldsymbol{Q_D}$ is non-singular and the (i, j) element of $(-\boldsymbol{Q_D})^{-1}$ is the mean time spent in state $j \in S_D$ before leaving S_D starting from $i \in S_D$.

3.2.1 Transformation of MDPs with no decisions in S_D

Let $\alpha(\pi)$ be the stationary probability vector of the Markov chain with generator $\boldsymbol{Q}(\pi)$. Then $\alpha(\pi)$ is the solution of the linear system $\alpha(\pi)\boldsymbol{Q}(\pi) = 0$ with normalizing equation $\alpha(\pi)\underline{\mathbf{1}} = 1$. Let $\alpha_U(\pi)$ and $\alpha_D(\pi)$ be the parts of vector $\alpha(\pi)$

associated with subsets S_U and S_D, respectively. Using (2), the partitioned form of the linear system is

$$\begin{aligned}\alpha_U(\pi)\boldsymbol{Q_U}(\pi) + \alpha_D(\pi)\boldsymbol{Q_{DU}} &= 0\\ \alpha_U(\pi)\boldsymbol{Q_{UD}}(\pi) + \alpha_D(\pi)\boldsymbol{Q_D} &= 0,\end{aligned} \tag{3}$$

from which we obtain a linear system for α_U

$$\alpha_U(\pi)(\boldsymbol{Q_U}(\pi) - \boldsymbol{Q_{UD}}(\pi)\boldsymbol{Q_D}^{-1}\boldsymbol{Q_{DU}}) = \alpha_U(\pi)\boldsymbol{Q_c}(\pi) = 0, \tag{4}$$

where

$$\boldsymbol{Q_c}(\pi) = \boldsymbol{Q_U}(\pi) + \boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}\boldsymbol{Q_{DU}}. \tag{5}$$

The Markov chain with state space S_U and generator $\boldsymbol{Q_c}(\pi)$ is referred to as censored Markov chain. It is obtained from the original Markov chain by "switching off the clock when the Markov chain visits S_D and switching on the clock when the Markov chain visits S_U" [13].

The censored Markov chain defines the stationary probability of the states in S_U through (4) apart from a normalizing constant, because $\sum_{i \in S_u} \alpha_i(\pi) = \alpha_U(\pi)\underline{\mathbf{1}}_U$ is not known based on (4). We can also express $\alpha_D(\pi)$ from (3) as

$$\alpha_D(\pi) = \alpha_U(\pi)\boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}, \tag{6}$$

from which

$$1 = \alpha(\pi)\underline{\mathbf{1}} = \alpha_D(\pi)\underline{\mathbf{1}} + \alpha_U(\pi)\underline{\mathbf{1}} = \alpha_U(\pi)(\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}\underline{\mathbf{1}}). \tag{7}$$

Using (6), we can rewrite (1) as

$$\begin{aligned}\pi^* &= \arg\max_{\pi} \alpha(\pi)\boldsymbol{C}(\pi)\underline{\mathbf{1}}\\ &= \arg\max_{\pi} \alpha_U(\pi)\boldsymbol{C_U}(\pi)\underline{\mathbf{1}} + \alpha_D(\pi)\boldsymbol{C_D}(\pi)\underline{\mathbf{1}}\\ &= \arg\max_{\pi} \alpha_U(\pi)\big(\boldsymbol{C_U}(\pi)\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}\boldsymbol{C_D}\underline{\mathbf{1}}\big)\end{aligned} \tag{8}$$

4 Parameters for the MDP compression method

In this section, we compute some measures of interest that are needed for the MDP compression method.

Let us consider a continuous time MDP with partition S_U (states with decision) and S_D (states with no decision). For $i \in S_U$, let $\rho_{S_U \setminus i}$ be the time to visit a state in S_U different from i, that is

$$\rho_{S_U \setminus i} = min(t | X(t) \in S_U \setminus i).$$

Based on the decision dependent state partitioning we define the $\boldsymbol{P}(\pi)$ matrix and the $\tau(\pi)$ and $c(\pi)$ vectors by their elements as follows:

$$\begin{aligned} \boldsymbol{P}_{ij}(\pi) &= Pr(X(\rho_{S_U \setminus i}) = j \mid X(0) = i), && (9)\\ \tau_i(\pi) &= E[\rho_{S_U \setminus i} \mid X(0) = i], && (10)\\ c_i(\pi) &= E[\int_{t=0}^{\rho_{S_U \setminus i}} C_{X(t)X(t)} dt \mid X(0) = i], && (11) \end{aligned}$$

where $i, j \in S_U$ and $i \neq j$. That is, assuming policy π, $\boldsymbol{P}_{ij}(\pi)$ is the probability that the process starting from $i \in S_U$ first enters to $S_U \setminus i$ in state j, $\tau_i(\pi)$ is the expected time to the first visit in $S_U \setminus i$ and $c_i(\pi)$ is the expected reward accumulated until this visit. $\boldsymbol{P}(\pi)_{ii} = 0$ by definition. The following theorem expresses $\boldsymbol{P}(\pi)$, $\tau(\pi)$ and $c(\pi)$ based on the partitioned description of the MDP.

Theorem 1. *$\boldsymbol{P}(\pi)$, $\tau(\pi)$, and $c(\pi)$ can be obtained as*

$$\boldsymbol{P}(\pi) = (-\boldsymbol{diagm}\langle \boldsymbol{Q_c}(\pi)\rangle)^{-1}(\boldsymbol{Q_c}(\pi) - \boldsymbol{diagm}\langle \boldsymbol{Q_c}(\pi)\rangle) \tag{12}$$

$$\tau(\pi) = (-\boldsymbol{diagm}\langle \boldsymbol{Q_c}(\pi)\rangle)^{-1}(\underline{\boldsymbol{1}} + \boldsymbol{Q_{UD}}(\pi)\boldsymbol{A}\underline{\boldsymbol{1}}), \tag{13}$$

$$c(\pi) = (-\boldsymbol{diagm}\langle \boldsymbol{Q_c}(\pi)\rangle)^{-1}(\boldsymbol{C_U}(\pi)\underline{\boldsymbol{1}} + \boldsymbol{Q_{UD}}(\pi)\boldsymbol{M}\underline{\boldsymbol{1}}), \tag{14}$$

*where $\boldsymbol{Q_c}(\pi) = \boldsymbol{Q_U}(\pi) + \boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}\boldsymbol{Q_{DU}}$, $\boldsymbol{A} = (-\boldsymbol{Q_D})^{-2}\boldsymbol{Q_{DU}}$, $\boldsymbol{M} = (-\boldsymbol{Q_D})^{-1}\boldsymbol{C_D}(-\boldsymbol{Q_D})^{-1}\boldsymbol{Q_{DU}}$ and **diagm**$\langle\rangle$ is the operator that creates a diagonal matrix from an input matrix by setting all its non-diagonal elements to zero.*

Proof. Although the formulas for $\boldsymbol{P}(\pi)$ and $\tau(\pi)$ can be derived alternatively by an easier approach, we will use a unified approach, for all three measures. Let ρ_{S_U} be the first time when the process visits S_U. We define matrix $\boldsymbol{G}(t)$ such that for $i \in S_D$ and $j \in S_U$, $\boldsymbol{G}_{ij}(t) = Pr(X(\rho_{S_U}) = j, \rho_{S_U} < t | X(0) = i)$ and matrix $\boldsymbol{g}(t)$ as $\boldsymbol{g}(t) = \frac{d}{dt}\boldsymbol{G}(t)$. That is, $\boldsymbol{G}_{ij}(t)$ is the probability that the process starting from state $i \in S_D$ will visit S_U before time t and the first visit will be to state $j \in S_U$. We can express $\boldsymbol{g}_{ij}(t)$ based on the first state transition as

$$\boldsymbol{g}_{ij}(t) = -\boldsymbol{Q_D}_{ii} \mathrm{e}^{\boldsymbol{Q_D}_{ii} t} \frac{\boldsymbol{Q_{DU}}_{ij}}{-\boldsymbol{Q_D}_{ii}} + \int_{\tau=0}^{t} -\boldsymbol{Q_D}_{ii} \mathrm{e}^{\boldsymbol{Q_D}_{ii}\tau} \sum_{k,k\neq i} \frac{\boldsymbol{Q_D}_{ik}}{-\boldsymbol{Q_D}_{ii}} \boldsymbol{g}_{kj}(t-\tau) d\tau. \tag{15}$$

Here $-\boldsymbol{Q_D}_{ii}\mathrm{e}^{\boldsymbol{Q_D}_{ii}\tau}$ corresponds to the density that the first state transition happens at time τ, $\frac{\boldsymbol{Q_{DU}}_{ij}}{-\boldsymbol{Q_D}_{ii}}$ is the probability that the process goes directly to state j at the first transition and $\sum_{k,k\neq i} \frac{\boldsymbol{Q_D}_{ik}}{-\boldsymbol{Q_D}_{ii}} \boldsymbol{g}_{kj}(t-\tau)$ is the probability density that the process goes to some other state in Q_D and it enters S_U in state j at time $t-\tau$.

For the Laplace transform of $\boldsymbol{g}_{ij}(t)$, $\boldsymbol{g^*}_{ij}(s) = \int_t e^{-st}\boldsymbol{g_{ij}}(t)dt$ we get

$$\boldsymbol{g^*}_{ij}(s) = \frac{-\boldsymbol{Q_D}_{ii}}{s-\boldsymbol{Q_D}_{ii}} \left(\frac{\boldsymbol{Q_{DU}}_{ij}}{-\boldsymbol{Q_D}_{ii}} + \sum_{k\in S_D, k\neq i} \frac{\boldsymbol{Q_D}_{ik}}{-\boldsymbol{Q_D}_{ii}} \boldsymbol{g^*}_{kj}(s) \right).$$

By multiplying both sides by $s - \boldsymbol{Q_D}_{ii}$ and adding $\boldsymbol{Q_D}_{ii}\boldsymbol{g^*}_{ii}(s)$ we obtain

$$s\boldsymbol{g^*}_{ij}(s) = \boldsymbol{Q_{DU}}_{ij} + \sum_{k\in S_D} \boldsymbol{Q_D}_{ik}\boldsymbol{g^*}_{kj}(s),$$

which can be written in matrix form as

$$s\boldsymbol{g^*}(s) = \boldsymbol{Q_{DU}} + \boldsymbol{Q_D}\boldsymbol{g^*}(s), \tag{16}$$

from which

$$\boldsymbol{g^*}(s) = (s\boldsymbol{I} - \boldsymbol{Q_D})^{-1}\boldsymbol{Q_{DU}}. \tag{17}$$

Since $\boldsymbol{Q_{DU}}$ and $\boldsymbol{Q_D}$ are policy independent, $\boldsymbol{g^*}(s)$ is policy independent as well. We define $\boldsymbol{G}_{ij} = \int_{t=0}^{\infty} \boldsymbol{g}_{ij}(t)dt = Pr(X(\rho_{S_U}) = j \mid X(0) = i \in S_D)$, which is the probability that the process starting from state $i \in S_D$ enters S_U in state j. From $\boldsymbol{G} = \int_{t=0}^{\infty} \boldsymbol{g}(t)dt = \lim_{s\to 0} \boldsymbol{g^*}(s)$ and (17) we have

$$\boldsymbol{G} = (-\boldsymbol{Q_D})^{-1}\boldsymbol{Q_{DU}}. \tag{18}$$

We note here that, since the generator of the MDP is stationary, and $\boldsymbol{Q_D}$ is transient, that is, $Pr(\rho_{S_U} < \infty \mid X(0) = i) = 1, \forall i \in S_D$, thus $\boldsymbol{G}\underline{\mathbf{1}} = \underline{\mathbf{1}}$.

From the moment generating property of the Laplace transform [7] we also have

$$\boldsymbol{A}_{ij} \overset{def}{=} \mathrm{E}[\rho_{S_U} I_{\{\rho_{S_U}<\infty, X(\rho_{S_U})=j\}} | X(0) = i] = -\frac{d}{ds}\boldsymbol{g^*}_{ij}(s)\big|_{s=0}.$$

By differentiating (16) according to s in $s = 0$ we obtain $\boldsymbol{G} = -\boldsymbol{Q_D}\boldsymbol{A}$, from which

$$\boldsymbol{A} = (-\boldsymbol{Q_D})^{-1}\boldsymbol{G} = (-\boldsymbol{Q_D})^{-2}\boldsymbol{Q_{DU}}. \tag{19}$$

Similar to $\boldsymbol{G}_{ij}(t)$ we define $\boldsymbol{K}_{ij}(r)$ as

$$\boldsymbol{K}_{ij}(r) = Pr\left(X(\rho_{S_U}) = j, \int_{t=0}^{\rho_{S_U}} \boldsymbol{C_D}_{X(t),X(t)}dt < r | X(0) = i \in S_D \right), \tag{20}$$

that is, $\boldsymbol{K}_{ij}(r)$ is the probability that the process starting from state i will visit S_U before reward r is accumulated, and the first visit will be to state $j \in S_U$. In a similar fashion to $\boldsymbol{g}_{ij}(t)$ we express $\boldsymbol{k}_{ij}(r) = \frac{d}{dr}\boldsymbol{K}_{ij}(r)$ based on the first state transition as

$$\begin{aligned}\boldsymbol{k}_{ij}(r) =& \frac{-\boldsymbol{Q}_{\boldsymbol{D}ii}}{\boldsymbol{C}_{\boldsymbol{D}ii}} \mathrm{e}^{\frac{\boldsymbol{Q}_{\boldsymbol{D}ii}}{\boldsymbol{C}_{\boldsymbol{D}ii}} r} \frac{\boldsymbol{Q}_{\boldsymbol{DU}ij}}{-\boldsymbol{Q}_{\boldsymbol{D}ii}} \\ &+ \int_{u=0}^{r} \frac{-\boldsymbol{Q}_{\boldsymbol{D}ii}}{\boldsymbol{C}_{\boldsymbol{D}ii}} \mathrm{e}^{\frac{\boldsymbol{Q}_{\boldsymbol{D}ii}}{\boldsymbol{C}_{\boldsymbol{D}ii}} u} \sum_{k \in S_D, k \neq i} \frac{\boldsymbol{Q}_{\boldsymbol{D}ik}}{-\boldsymbol{Q}_{\boldsymbol{D}ii}} \boldsymbol{k}_{kj}(r-u) du.\end{aligned} \tag{21}$$

Compared to (15), the difference is that instead of the elapsed time we consider the reward accumulated up to the first state transition, which is exponentially distributed with rate $\frac{-\boldsymbol{Q}_{\boldsymbol{D}ii}}{\boldsymbol{C}_{\boldsymbol{D}ii}}$. Following the same steps as in the case of $\boldsymbol{g}_{ij}(t)$ we get that the Laplace transform of $\boldsymbol{k}(r)$, $\boldsymbol{k}^*(s)$, satisfies

$$s\boldsymbol{C}_{\boldsymbol{D}}\boldsymbol{k}^*(s) = \boldsymbol{Q}_{\boldsymbol{DU}} + \boldsymbol{Q}_{\boldsymbol{D}}\boldsymbol{k}^*(s). \tag{22}$$

Just like $\boldsymbol{g}^*(s)$, $\boldsymbol{k}^*(s)$ is also independent of the actual policy, since it depends on the process behaviour during a visit in S_D. From (22) we have

$$\boldsymbol{K} \overset{def}{=} \lim_{r\to\infty} \boldsymbol{K}(r) = \int_{r=0}^{\infty} \boldsymbol{k}(r)dr = \lim_{s\to 0} \boldsymbol{k}^*(s) = (-\boldsymbol{Q}_{\boldsymbol{D}})^{-1}\boldsymbol{Q}_{\boldsymbol{DU}},$$

where

$$\boldsymbol{K}_{ij} = Pr(X(\rho_{S_U}) = j \,|X(0) = i \in S_D) = \boldsymbol{G}_{ij}.$$

Similar to $\boldsymbol{A}$, we introduce

$$\boldsymbol{M}_{ij} \overset{def}{=} \mathrm{E}\left[I_{\{X(\gamma_U)=j\}} \int_{t=0}^{\rho_U} \boldsymbol{C}_{\boldsymbol{D}X(t),X(t)} dt \middle| X(0) = i \in S_D \right] = -\frac{d}{ds}\boldsymbol{k^*}_{ij}(s)\bigg|_{s=0},$$

which can be obtained from the derivative of (22) in $s = 0$,from which

$$\boldsymbol{C}_{\boldsymbol{D}}\boldsymbol{K} = -\boldsymbol{Q}_{\boldsymbol{D}}\boldsymbol{M},$$

from which

$$\boldsymbol{M} = (-\boldsymbol{Q}_{\boldsymbol{D}})^{-1}\boldsymbol{C}_{\boldsymbol{D}}(-\boldsymbol{Q}_{\boldsymbol{D}})^{-1}\boldsymbol{Q}_{\boldsymbol{DU}}. \tag{23}$$

Having the $\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{M}$ matrices, describing the behaviour of the process in S_D, we are ready to express $\boldsymbol{P}(\pi)$, $\tau(\pi)$ and $c(\pi)$. The required derivations follow the same pattern. In each cases the formulas will be broken up into three terms according to the following three cases:

- Case 1: the process moves to state $j \in S_U \setminus i$ during the first state transition.

- Case 2: the process first moves to $k \in S_D$, spends some time in S_D, then enters S_U in state $j \neq i$.
- Case 3: the process first moves to $k \in S_D$, spends some time in S_D, then enters S_U in state i. This case adds a recursive term to the formulas.

We start with $\boldsymbol{P}_{ij}(\pi)$ for $i \neq j$, which gives the probability of entering set $S_U \setminus i$ in state $j \in S_U$ when the process starts in $i \in S_U$:

$$\begin{aligned}\boldsymbol{P}_{ij}(\pi) &= Pr(\rho_{S_U \setminus i} < \infty, X(\rho_{S_U \setminus i}) = j \mid X(0) = i) = \\ &\underbrace{\frac{\boldsymbol{Q_U}_{ij}(\pi)}{-\boldsymbol{Q_U}_{ii}(\pi)}}_{Case\ 1} + \underbrace{\sum_{k \in S_D} \frac{\boldsymbol{Q_{UD}}_{ik}(\pi)}{-\boldsymbol{Q_U}_{ii}(\pi)} \boldsymbol{G}_{kj}}_{Case\ 2} + \underbrace{\sum_{k \in S_D} \frac{\boldsymbol{Q_{UD}}_{ik}(\pi)}{-\boldsymbol{Q_U}_{ii}(\pi)} \boldsymbol{G}_{ki} \boldsymbol{P}_{ij}(\pi)}_{Case\ 3},\end{aligned}$$

from which

$$-\left(\boldsymbol{Q_U}_{ii}(\pi) + \sum_{k \in S_D} \boldsymbol{Q_{UD}}_{ik}(\pi) \boldsymbol{G}_{ki}\right) \boldsymbol{P}_{ij}(\pi) = \boldsymbol{Q_U}_{ij}(\pi) + \sum_{k \in S_D} \boldsymbol{Q_{UD}}_{ik}(\pi) \boldsymbol{G}_{kj}. \tag{24}$$

As defined before let **diagm**$\langle\rangle$ be the operator that creates a diagonal matrix from an input matrix such that all non-diagonal elements of the original matrix are set to zero. Using this notation we can write (24) in matrix form as

$$\begin{aligned}\boldsymbol{P}(\pi) &= (-\textbf{diagm}\langle \boldsymbol{Q_U}(\pi) + \boldsymbol{Q_{UD}}(\pi)\boldsymbol{G}\rangle)^{-1} \\ &\quad \cdot (\boldsymbol{Q_U}(\pi) + \boldsymbol{Q_{UD}}(\pi)\boldsymbol{G} - \textbf{diagm}\langle \boldsymbol{Q_U}(\pi) + \boldsymbol{Q_{UD}}(\pi)\boldsymbol{G}\rangle) \\ &= (-\textbf{diagm}\langle \boldsymbol{Q_c}(\pi)\rangle)^{-1}(\boldsymbol{Q_c}(\pi) - \textbf{diagm}\langle \boldsymbol{Q_c}(\pi)\rangle)\end{aligned} \tag{25}$$

where $-\textbf{diagm}\langle \boldsymbol{Q_c}(\pi)\rangle$ in the second term of the right side ensures that the diagonal of $\boldsymbol{P}(\pi)$ is equal to zero according to the definition of $\boldsymbol{P}_{ii}(\pi)$ and we used that $\boldsymbol{Q_c}(\pi) = \boldsymbol{Q_U}(\pi) + \boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}\boldsymbol{Q_{DU}} = \boldsymbol{Q_U}(\pi) + \boldsymbol{Q_{UD}}(\pi)\boldsymbol{G}$ (which comes from the definition of $\boldsymbol{Q_c}(\pi)$ in (5) and the definition of $\boldsymbol{G}$ in (19)).

The formula for $\tau_i(\pi)$, which describes the expected time before entering $S_U \setminus i$ when the process starts in $i \in S_U$, has a similar structure:

$$\tau_i(\pi) = E\left[\rho_{U\setminus i} | X(0) = i, \exists \rho_{S_U\setminus i}\right] = \underbrace{\sum_{j\in S_U\setminus i} \frac{\boldsymbol{Q_U}_{ij}(\pi)}{-\boldsymbol{Q_U}_{ii}(\pi)} \frac{1}{-\boldsymbol{Q_U}_{ii}(\pi)}}_{Case\ 1} +$$

$$\underbrace{\sum_{j\in S_U\setminus i} \sum_{k\in S_D} \frac{\boldsymbol{Q_{UD}}_{ik}(\pi)}{-\boldsymbol{Q_U}_{ii}(\pi)} \left(\boldsymbol{G}_{kj} \frac{1}{-\boldsymbol{Q_U}_{ii}(\pi)} + \boldsymbol{A}_{kj}\right)}_{Case\ 2} +$$

$$\underbrace{\sum_{k\in S_D} \frac{\boldsymbol{Q_{UD}}_{ik}(\pi)}{-\boldsymbol{Q_U}_{ii}(\pi)} \left(\boldsymbol{G}_{ki} \frac{1}{-\boldsymbol{Q_U}_{ii}(\pi)} + \boldsymbol{A}_{ki} + \boldsymbol{G}_{ki}\tau_i(\pi)\right)}_{Case\ 3}$$

$$= \frac{1}{-\boldsymbol{Q_U}_{ii}(\pi)} + \sum_{j\in S_U} \sum_{k\in S_D} \frac{\boldsymbol{Q_{UD}}_{ik}(\pi)}{-\boldsymbol{Q_U}_{ii}(\pi)} \boldsymbol{A}_{kj} + \sum_{k\in S_D} \frac{\boldsymbol{Q_{UD}}_{ik}(\pi)}{-\boldsymbol{Q_U}_{ii}(\pi)} \boldsymbol{G}_{ki}\tau_i(\pi), \quad (26)$$

where we used that $\boldsymbol{Q_D}$ is transient, thus $\sum_{j\in S_U} \boldsymbol{G}_{kj} = 1, \forall k \in S_D$. The first term (Case 1) is the mean time until the first transition multiplied by the probability of Case 1. In the second term (Case 2) $\sum_{j\in S_U\setminus i} \sum_{k\in S_D} \frac{\boldsymbol{Q_{UD}}_{ik}(\pi)}{-\boldsymbol{Q_U}_{ii}(\pi)} \boldsymbol{G}_{kj} \frac{1}{-\boldsymbol{Q_U}_{ii}(\pi)}$ is the time until the first transition multiplied by the probability of Case 2 and summed for all $j \in S_U \setminus i$ and $\sum_{j\in S_U\setminus i} \sum_{k\in S_D} \frac{\boldsymbol{Q_{UD}}_{ik}(\pi)}{-\boldsymbol{Q_U}_{ii}(\pi)} \boldsymbol{A}_{kj}$ is the remaining time until visiting state j multiplied by the probability of Case 2 and summed for all $j \in S_U \setminus i$. We note that in this part multiplication by $\boldsymbol{G}_{kj}$ is not necessary, because $\boldsymbol{A}_{kj}$ already contains the probability $\boldsymbol{G}_{kj}$. The third term contains analogous components to the second term, but for Case 3, when the process enters S_U in state i, we also have the added $\sum_{k\in S_D} \frac{\boldsymbol{Q_{UD}}_{ik}(\pi)}{-\boldsymbol{Q_U}_{ii}(\pi)} \boldsymbol{G}_{ki}\tau_i(\pi)$, because if the process returns to state i, an additional $\tau_i(\pi)$ time is needed to visit some state in $S_U \setminus i$. We can express $\tau_i(\pi)$ from (26) as

$$-\left(\boldsymbol{Q_U}_{ii}(\pi) + \sum_{k\in S_D} \boldsymbol{Q_{UD}}_{ik}(\pi)\boldsymbol{G}_{ki}\right)\tau_i(\pi) = 1 + \sum_{j\in S_U} \sum_{k\in S_D} \boldsymbol{Q_{UD}}_{ik}(\pi)\boldsymbol{A}_{kj}. \quad (27)$$

Using $\boldsymbol{G} = (-\boldsymbol{Q_D})^{-1}\boldsymbol{Q_{DU}}$ from (18), the matrix form of (27) is

$$\tau(\pi) = (-\mathbf{diagm}\langle \boldsymbol{Q_U}(\pi) + \boldsymbol{Q_{UD}}(\pi)\boldsymbol{G}\rangle)^{-1}(\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)\boldsymbol{A}\underline{\mathbf{1}}), \quad (28)$$

which is identical with (13), the corresponding equation of Theorem 1.

Finally we have a very similar expression for the elements of the $c(\pi)$ vector:

$$
\begin{aligned}
c_i(\pi) = E[\int_{t=0}^{\rho_{S_U\setminus i}} \boldsymbol{C}_{X(t)X(t)} \mid X(0)=i] = \underbrace{\sum_{j\in S_U\setminus i} \frac{\boldsymbol{Q}_{\boldsymbol{U}ij}(\pi)}{-\boldsymbol{Q}_{\boldsymbol{U}ii}(\pi)} \frac{\boldsymbol{C}_{\boldsymbol{U}ii}(\pi)}{-\boldsymbol{Q}_{\boldsymbol{U}ii}(\pi)}}_{Case\ 1} + \\
\underbrace{\sum_{j\in S_U\setminus i} \sum_{k\in S_D} \frac{\boldsymbol{Q}_{\boldsymbol{UD}ik}(\pi)}{-\boldsymbol{Q}_{\boldsymbol{U}ii}(\pi)} \left(\boldsymbol{G}_{kj} \frac{\boldsymbol{C}_{\boldsymbol{U}ii}(\pi)}{-\boldsymbol{Q}_{\boldsymbol{U}ii}(\pi)} + \boldsymbol{M}_{kj}\right)}_{Case\ 2} + \\
\underbrace{\sum_{k\in S_D} \frac{\boldsymbol{Q}_{\boldsymbol{UD}ik}(\pi)}{-\boldsymbol{Q}_{\boldsymbol{U}ii}(\pi)} \left(\boldsymbol{G}_{ki} \frac{\boldsymbol{C}_{\boldsymbol{U}ii}(\pi)}{-\boldsymbol{Q}_{\boldsymbol{U}ii}(\pi)} + \boldsymbol{M}_{ki} + \boldsymbol{G}_{ki} c_i(\pi)\right)}_{Case\ 3} \\
= \frac{\boldsymbol{C}_{\boldsymbol{U}ii}(\pi)}{-\boldsymbol{Q}_{\boldsymbol{U}ii}(\pi)} + \sum_{j\in S_U} \sum_{k\in S_D} \frac{\boldsymbol{Q}_{\boldsymbol{UD}ik}(\pi)}{-\boldsymbol{Q}_{\boldsymbol{U}ii}(\pi)} \boldsymbol{M}_{kj} + \sum_{k\in S_D} \frac{\boldsymbol{Q}_{\boldsymbol{UD}ik}(\pi)}{-\boldsymbol{Q}_{\boldsymbol{U}ii}(\pi)} \boldsymbol{G}_{ki} c_i(\pi). \quad (29)
\end{aligned}
$$

This expression follows the same logic as (26), the only difference is that instead of times we have rewards accumulated over those times, thus the $\frac{1}{-\boldsymbol{Q}_{\boldsymbol{U}ii}}(\pi)$ terms in (26) are replaced by $\frac{\boldsymbol{C}_{\boldsymbol{U}ii}(\pi)}{-\boldsymbol{Q}_{\boldsymbol{U}ii}}(\pi)$, $\boldsymbol{A}$ is changed to $\boldsymbol{M}$ and $\tau_i(\pi)$ is changed to $c_i(\pi)$. We can express $c_i(\pi)$ from (29) as

$$
-\left(\boldsymbol{Q}_{\boldsymbol{U}ii}(\pi) + \sum_{k\in S_D} \boldsymbol{Q}_{\boldsymbol{UD}ik}(\pi) \boldsymbol{G}_{ki}\right) c_i(\pi) = \boldsymbol{C}_{\boldsymbol{U}ii}(\pi) + \sum_{j\in S_U} \sum_{k\in S_D} \boldsymbol{Q}_{\boldsymbol{UD}ik}(\pi) \boldsymbol{M}_{kj}.
$$

Using the same substitutions as in (12) and (13), the matrix form of this equation is identical to (14), the final equation of Theorem 1. □

5 Compression of partitioned MDPs

In this section, we present the main contribution of the paper, the compressed representation of partitioned MDPs. We discuss the idea behind the compression, and provide analytical proof for the equivalence of the original and the compressed forms.

Theorem 2. *Let $(S, A, \boldsymbol{Q}(\pi), \boldsymbol{C}(\pi))$ be an MDP with irreducible $\boldsymbol{Q}(\pi)$, where the generator and reward-rate matrices can be partitioned according to (2). The MDP defined by $(S', A', \boldsymbol{Q'}(\pi), \boldsymbol{C'}(\pi))$ and the one defined by $(S, A, \boldsymbol{Q}(\pi), \boldsymbol{C}(\pi))$ have the same optimal policy, where $S' = S_U$, $A' = A$,*

$$
\boldsymbol{Q'}_{ij}(\pi) = \begin{cases} -\frac{1}{\tau_i(\pi)}, & \textit{if } i = j, \\ \frac{\boldsymbol{P}_{ij}(\pi)}{\tau_i(\pi)}, & \textit{otherwise,} \end{cases} \quad \textit{and} \quad \boldsymbol{C'}_{ij}(\pi) = \begin{cases} \frac{c_i(\pi)}{\tau_i(\pi)}, & \textit{if } i = j, \\ 0, & \textit{otherwise.} \end{cases}
$$

Remark. *The core idea of the compression is the following. When the process exits a state $i \in S_U$, it can take multiple trajectories. It can either directly transition to another state in $S_U \setminus \{i\}$, or it can go through a number of transitions in $S_D \cup \{i\}$ before transitioning to a state in $S_U \setminus \{i\}$. When the decision is made in state i, the distribution of the next state reached in $S_U \setminus \{i\}$, the distribution of the time until reaching this state, and the distribution of the accumulated reward until reaching this state can be calculated (and no further knowledge of decisions in other states is needed). The theorem states that, to optimise the long-term expected reward rate, we can change the distribution of the time needed to reach $S_U \setminus \{i\}$ and the associated accumulated reward to exponentially distributed variables; as long as their expected values match with the expected value of the original distributions, the optimal policy will not change.*

Proof. Let us denote by $\textbf{diagv}\langle v \rangle$ the diagonal matrix created from the elements of vector v, let $\boldsymbol{S}(\pi) = \textbf{diagv}\langle \tau(\pi) \rangle$ and $\boldsymbol{Z}(\pi) = -\textbf{diagm}\langle Q_c(\pi) \rangle$. Then, using the formula for $\boldsymbol{P}(\pi)$ from (12) we can write

$$\begin{aligned}
\boldsymbol{Q'}(\pi) &= \textbf{diagv}\langle \tau(\pi) \rangle^{-1}(-\boldsymbol{I} + \boldsymbol{P}(\pi)) \\
&= \boldsymbol{S}^{-1}(\pi)(-\boldsymbol{I} + (-\textbf{diagm}\langle \boldsymbol{Q_c}(\pi) \rangle)^{-1}(\boldsymbol{Q_c}(\pi) - \textbf{diagm}\langle \boldsymbol{Q_c}(\pi) \rangle)) \qquad (30) \\
&= \boldsymbol{S}^{-1}(\pi)\boldsymbol{Z}^{-1}(\pi)\boldsymbol{Q_c}(\pi).
\end{aligned}$$

and using the formula for $c(\pi)$ from (14) we have

$$\begin{aligned}
\boldsymbol{C'}(\pi) &= \textbf{diagv}\langle \tau(\pi) \rangle^{-1}\textbf{diagm}\langle c(\pi) \rangle \\
&= \boldsymbol{S}^{-1}(\pi)(-\textbf{diagm}\langle \boldsymbol{Q_c}(\pi) \rangle)^{-1}(\boldsymbol{C_U}(\pi)\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)\boldsymbol{M}\underline{\mathbf{1}}) \\
&= \boldsymbol{S}^{-1}(\pi)\boldsymbol{Z}^{-1}(\pi)(\boldsymbol{C_U}(\pi)\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}\boldsymbol{C_D}(-\boldsymbol{Q_D})^{-1}\boldsymbol{Q_{DU}}\underline{\mathbf{1}}) \\
&= \boldsymbol{S}^{-1}(\pi)\boldsymbol{Z}^{-1}(\pi)(\boldsymbol{C_U}(\pi)\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}\boldsymbol{C_D}\underline{\mathbf{1}}),
\end{aligned} \tag{31}$$

where we used that $(-\boldsymbol{Q_D})^{-1}\boldsymbol{Q_{DU}}\underline{\mathbf{1}} = \underline{\mathbf{1}}$ for stationary MDPs. The $\alpha'(\pi)$ stationary probability vector of the compressed MDP has to satisfy the

$$\begin{aligned}
\alpha'(\pi)\boldsymbol{Q'}(\pi) &= 0 \\
\alpha'(\pi)\underline{\mathbf{1}} &= 1
\end{aligned}$$

system of equations. The first equation can be transformed as

$$0 = \alpha'(\pi)\boldsymbol{Q'}(\pi) = \alpha'(\pi)\boldsymbol{S}^{-1}(\pi)\boldsymbol{Z}^{-1}(\pi)\boldsymbol{Q_c}(\pi) = \alpha''(\pi)\boldsymbol{Q_c}(\pi), \tag{32}$$

where $\alpha''(\pi) = \alpha'(\pi)\boldsymbol{S}^{-1}(\pi)\boldsymbol{Z}^{-1}(\pi)$. The second equation can be transformed as

$$\begin{aligned}1 &= \alpha'(\pi)\underline{\mathbf{1}}\\ &= \alpha'(\pi)\boldsymbol{S}^{-1}(\pi)\boldsymbol{Z}^{-1}(\pi)\boldsymbol{S}(\pi)\boldsymbol{Z}(\pi)\underline{\mathbf{1}}\\ &= \alpha''(\pi)(-\textbf{diagm}\langle Q_c(\pi)\rangle)\textbf{diagv}\langle\tau(\pi)\rangle\underline{\mathbf{1}}\\ &= \alpha''(\pi)(-\textbf{diagm}\langle Q_c(\pi)\rangle)(-\textbf{diagm}\langle Q_c\rangle(\pi))^{-1}(\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)\boldsymbol{A}\underline{\mathbf{1}})\\ &= \alpha''(\pi)(\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)\boldsymbol{A}\underline{\mathbf{1}}).\end{aligned}$$

Using the formula for $\boldsymbol{A}$ from (18) and using that $(-\boldsymbol{Q_D})^{-1}\boldsymbol{Q_{DU}}\underline{\mathbf{1}} = \underline{\mathbf{1}}$ we can further transform the expression as

$$1 = \alpha''(\pi)(\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}\underline{\mathbf{1}}). \tag{33}$$

Thus, from (32) and (33) we obtain the

$$0 = \alpha''(\pi)\boldsymbol{Q_c}(\pi)\ ,\ 1 = \alpha''(\pi)(\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}\underline{\mathbf{1}})$$

system of linear equations for $\alpha''(\pi)$ that completely determine $\alpha''(\pi)$. However these are the same as the linear equations for $\alpha_U(\pi)$, (4) and (7), thus $\alpha''(\pi) = \alpha_U(\pi)$.

The mean reward rate of the compressed MDP can be given using (31) as

$$\begin{aligned}\alpha'(\pi)\boldsymbol{C'}(\pi)\underline{\mathbf{1}} &= \alpha'(\pi)\boldsymbol{S}^{-1}(\pi)\boldsymbol{Z}^{-1}(\pi)(\boldsymbol{C_U}(\pi)\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}\boldsymbol{C_D}\underline{\mathbf{1}})\\ &= \alpha_U(\pi)(\boldsymbol{C_U}(\pi)\underline{\mathbf{1}} + \boldsymbol{Q_{UD}}(\pi)(-\boldsymbol{Q_D})^{-1}\boldsymbol{C_D}\underline{\mathbf{1}})\end{aligned}$$

where we used that $\alpha''(\pi) = \alpha_U(\pi) = \alpha'(\pi)\boldsymbol{S}^{-1}(\pi)\boldsymbol{Z}^{-1}(\pi)$. This, however, is the same as the mean reward rate for the original MDP, as can be seen from (8), thus the average reward rate of the original and the compressed MDP is the same for any given policy, thus their optimal policies are also the same. □

6 Compression of partitioned MDPs with special structures

In the previous section, we presented the general formulas of the proposed MDP compression method. In this section, we discuss the application of the method in some special, practically important cases. The general formulas for the compression method rely on the calculation of the $\boldsymbol{G}$, $\boldsymbol{A}$, and $\boldsymbol{M}$ matrices. For MDPs with finite S_D the calculation of these matrices can be done based on (18), (19) and (23), which are based on the computation of the inverse of $\boldsymbol{Q_D}$. If the S_D subset is infinite, however, the calculation of these matrices is not trivial and it has to rely on the structural regularity of the MDP. We discuss two important cases where the calculation of $\boldsymbol{G}$, $\boldsymbol{A}$, and $\boldsymbol{M}$ matrices are possible, which are the cases when the structures of the MDP in S_D are spatially homogeneous M/G/1-type and G/M/1-type.

During the analysis of matrix $\boldsymbol{G}$ and $\boldsymbol{A}$ we are going to utilize some known results of M/G/1-type and G/M/1-type processes, while the analysis presented for matrix $\boldsymbol{M}$ was not discussed in the literature to the best of the authors' knowledge.

6.1 The M/G/1 type process

An MDP is of M/G/1 type (with the considered $\{S_U, S_D\}$ partitioning) if its generator matrix has the following block structure

$$\boldsymbol{Q}(\pi) = \left(\begin{array}{c|cccccc} \bar{\boldsymbol{L}}(\pi) & \bar{\boldsymbol{F}}_{\boldsymbol{1}}(\pi) & \bar{\boldsymbol{F}}_{\boldsymbol{2}}(\pi) & \dots & & & \\ \hline \boldsymbol{B} & \boldsymbol{L} & \boldsymbol{F}_{\boldsymbol{1}} & \boldsymbol{F}_{\boldsymbol{2}} & \dots & & \\ & \boldsymbol{B} & \boldsymbol{L} & \boldsymbol{F}_{\boldsymbol{1}} & \boldsymbol{F}_{\boldsymbol{2}} & \dots & \\ & & \ddots & \ddots & \ddots & \ddots & \end{array} \right). \tag{34}$$

The MDP can be partitioned to levels according to the blocks so that each row in (34) corresponds to a separate level. In the following we assume that the reward rate matrix also has some level based regularity, thus

$$\boldsymbol{C}(\pi) = \left(\begin{array}{c|ccc} \boldsymbol{C}_{\boldsymbol{1}}(\pi) & & & \\ \hline & \boldsymbol{C}_{\boldsymbol{2}} & & \\ & & \boldsymbol{C}_{\boldsymbol{3}} & \\ & & & \ddots \end{array} \right), \tag{35}$$

where $\boldsymbol{C}_{\boldsymbol{n}} = f(\boldsymbol{C}_{\boldsymbol{0}}, n)$ for $n = 2, 3, \dots,$. We discuss some specific cases in Section 7.2.

In the following we denote the ith state of the nth level by $v_{n,i}$, where $i \in \Phi = \{1, 2, \dots, \phi\}$ and $n \in \{1, 2, \dots\}$. Using the level based partitioning the process cannot descend more than one level during a single state transition (i.e., a transition cannot happen from level n to level $n-k,\ k \geq 2$), and any downward transition between neighbouring levels is described by the $\boldsymbol{B}$ matrix. We define the levels such that level 1 is identical with S_U and the rest of the levels are in S_D, thus the S_U, S_D based decomposition of the $\boldsymbol{Q}(\pi)$ and $\boldsymbol{C}(\pi)$ is

$$\boldsymbol{Q_U}(\pi) = \bar{\boldsymbol{L}}(\pi), \quad \boldsymbol{Q_{UD}}(\pi) = \begin{pmatrix} \bar{\boldsymbol{F}}_{\boldsymbol{1}}(\pi) & \bar{\boldsymbol{F}}_{\boldsymbol{2}}(\pi) & \dots \end{pmatrix},$$
$$\boldsymbol{Q_{DU}} = \begin{pmatrix} \boldsymbol{B} \\ 0 \\ \vdots \end{pmatrix}, \quad \boldsymbol{Q_D} = \begin{pmatrix} \boldsymbol{L} & \boldsymbol{F_1} & \boldsymbol{F_2} & \dots & \\ \boldsymbol{B} & \boldsymbol{L} & \boldsymbol{F_1} & \boldsymbol{F_2} & \dots \\ & \ddots & \ddots & \ddots & \ddots \end{pmatrix},$$
$$\boldsymbol{C_U}(\pi) = \boldsymbol{C_1}(\pi), \quad \boldsymbol{C_D} = \begin{pmatrix} \boldsymbol{C_2} & & \\ & \boldsymbol{C_3} & \\ & & \ddots \end{pmatrix}.$$

To compute the $\boldsymbol{G}$, $\boldsymbol{A}$, and $\boldsymbol{M}$ matrices for the infinite S_D we define the $\hat{\boldsymbol{G}}$, $\hat{\boldsymbol{A}}$, and $\hat{\boldsymbol{M}}_{\boldsymbol{n}}$ matrices that are similar, however, instead of trajectories from S_D to S_U they describe trajectories from level n to level $n-1$ inside S_D. Matrix $\hat{\boldsymbol{G}}$ is known to be the characteristic matrix of the M/G/1-type process [15] and is well discussed in the literature, unlike the analysis of matrix $\hat{\boldsymbol{M}}_{\boldsymbol{n}}$.

Theorem 3. *For an MDP with M/G/1-type structure in S_D according to* (34) *and* (35) *the parameters of the reduced MDP representation are*

$$\boldsymbol{P}(\pi) = \left(-\boldsymbol{diagm}\langle \boldsymbol{Q_c}(\pi)\rangle\right)^{-1} \left(\boldsymbol{Q_c}(\pi) - \boldsymbol{diagm}\langle \boldsymbol{Q_c}(\pi)\rangle\right), \tag{36}$$
$$\tau(\pi) = \left(-\boldsymbol{diagm}\langle \boldsymbol{Q_c}(\pi)\rangle\right)^{-1} \left(\underline{\boldsymbol{1}} + \sum_{i=1}^{\infty} \bar{\boldsymbol{F}}_{\boldsymbol{i}}(\pi) \sum_{\ell=0}^{i} \hat{\boldsymbol{G}}^{\ell} \hat{\boldsymbol{A}} \hat{\boldsymbol{G}}^{i-\ell} \underline{\boldsymbol{1}}\right), \tag{37}$$
$$c(\pi) = \left(-\boldsymbol{diagm}\langle \boldsymbol{Q_c}(\pi)\rangle\right)^{-1} \left(\boldsymbol{C_1}(\pi)\underline{\boldsymbol{1}} + \sum_{i=1}^{\infty} \bar{\boldsymbol{F}}_{\boldsymbol{i}}(\pi) \sum_{\ell=0}^{i} \hat{\boldsymbol{G}}^{\ell} \hat{\boldsymbol{M}}_{\boldsymbol{i+1-\ell}} \hat{\boldsymbol{G}}^{i-\ell} \underline{\boldsymbol{1}}\right), \tag{38}$$

where $\boldsymbol{Q_c}(\pi) = \bar{\boldsymbol{L}}(\pi) + \sum_{i=1}^{\infty} \bar{\boldsymbol{F}}_{\boldsymbol{i}}(\pi) \hat{\boldsymbol{G}}^{i}$, $\hat{\boldsymbol{G}}$ is the minimal non-negative solution of

$$\boldsymbol{0} = \boldsymbol{B} + \boldsymbol{L}\hat{\boldsymbol{G}} + \sum_{m=1}^{\infty} \boldsymbol{F_m} \hat{\boldsymbol{G}}^{m+1}, \tag{39}$$

$\hat{\boldsymbol{A}}$ is the solution of the linear matrix equation

$$\hat{\boldsymbol{G}} = \boldsymbol{L}\hat{\boldsymbol{A}} + \sum_{m=1}^{\infty} \boldsymbol{F_m} \sum_{\ell=0}^{m} \hat{\boldsymbol{G}}^{\ell} \hat{\boldsymbol{A}} \hat{\boldsymbol{G}}^{m-\ell}, \tag{40}$$

and $\hat{\boldsymbol{M}}_{\boldsymbol{n}}$ is the solution of

$$\boldsymbol{C_n}\hat{\boldsymbol{G}} = \boldsymbol{L}\hat{\boldsymbol{M}}_{\boldsymbol{n}} + \sum_{m=1}^{\infty} \boldsymbol{F_m} \sum_{\ell=0}^{m} \hat{\boldsymbol{G}}^{\ell} \hat{\boldsymbol{M}}_{\boldsymbol{n+m-\ell}} \hat{\boldsymbol{G}}^{m-\ell}. \tag{41}$$

Remark. *Efficient numerical methods are available for the solution of* (39) *e.g., in [4]. The solution of* (40) *can be achieved e.g., with the use of the column stacking* vec *operator, for which* $vec(\boldsymbol{ABC}) = (\boldsymbol{C}^T \otimes \boldsymbol{A})vec(\boldsymbol{B})$. *Applying* vec *for* (40) *gives*

$$vec(\hat{\boldsymbol{G}}) = (\boldsymbol{I} \otimes \boldsymbol{L})\, vec(\hat{\boldsymbol{A}}) + \sum_{m=1}^{\infty} \sum_{\ell=0}^{m} \left(\hat{\boldsymbol{G}}^{m-\ell T} \otimes \boldsymbol{F_m} \hat{\boldsymbol{G}}^{\ell} \right) vec(\hat{\boldsymbol{A}}),$$

from which

$$vec(\hat{\boldsymbol{A}}) = \left((\boldsymbol{I} \otimes \boldsymbol{L}) + \sum_{m=1}^{\infty} \sum_{\ell=0}^{m} \left(\hat{\boldsymbol{G}}^{m-\ell T} \otimes \boldsymbol{F_m} \hat{\boldsymbol{G}}^{\ell} \right) \right)^{-1} vec(\hat{\boldsymbol{G}}).$$

The solution of (41) *is more difficult in general. It is discussed in Section 7.2 for some special* $\boldsymbol{C_n}$ *(and* $\boldsymbol{F_n}$*) series.*

We also note that the expressions in the theorem can be further simplified based on the fact that the characteristic matrix of a positive recurrent M/G/1 type process is a stochastic matrix, that is $\hat{\boldsymbol{G}}\underline{\boldsymbol{1}} = \underline{\boldsymbol{1}}$. *When applying this simplification it is enough to compute vectors* $\hat{\boldsymbol{A}}\underline{\boldsymbol{1}}$ *and* $\hat{\boldsymbol{M}}_{\boldsymbol{i}}\underline{\boldsymbol{1}}$ *instead of matrices* $\hat{\boldsymbol{A}}$ *and* $\hat{\boldsymbol{M}}_{\boldsymbol{i}}$.

Proof. Similar to the proof of Theorem 1 we apply a unified approach for the analysis of all required measures. For the analysis of the level process in S_D, we define $\hat{\boldsymbol{G}}_{ij}(t) = Pr(X(\rho_{n-1}) = v_{n-1,j}, \rho_{n-1} < t | X(0) = v_{n,i})$, where ρ_{n-1} is the time of the first visit to level $n-1$, i.e., $\hat{\boldsymbol{G}}_{ij}(t)$ is the probability that the process, starting from state i of level n reaches level $n-1$ before time t and the first visit is to state j on this level. We also define the multi level version of this measure, $\hat{\boldsymbol{G}}_{\boldsymbol{m}ij}$, that describe trajectories from level n to level $n-m$,

$$\hat{\boldsymbol{G}}_{\boldsymbol{m}ij}(t) = Pr(X(\rho_{n-m}) = v_{n-1,j}, \rho_{n-m} < t | X(0) = v_{n,i}).$$

Furthermore, $\hat{\boldsymbol{g}}(t) = \frac{d}{dt}\hat{\boldsymbol{G}}(t)$ and $\hat{\boldsymbol{g}}_{\boldsymbol{m}}(t) = \frac{d}{dt}\hat{\boldsymbol{G}}_{\boldsymbol{m}}(t)$. By definition, we have $\hat{\boldsymbol{g}}_{\boldsymbol{1}}(t) = \hat{\boldsymbol{g}}(t)$ and by the fact that the first visit from level n to level $n-m$ can be decomposed into the first visit from level n to level $n-1$ and then the first visit from level $n-1$ to level $n-m$ we also have $\hat{\boldsymbol{g}}_{\boldsymbol{m}}(t) = (\hat{\boldsymbol{g}} * \hat{\boldsymbol{g}}_{\boldsymbol{m-1}})(t)$, $\forall m \geq 2$, where $*$ is the convolution operator, i.e., $(\boldsymbol{a}*\boldsymbol{b})(t) = \int_{\tau=0}^{t} \boldsymbol{a}(\tau)\boldsymbol{b}(t-\tau)d\tau$. Similar to $\boldsymbol{g}_{ij}(t)$, we can express $\hat{\boldsymbol{g}}_{ij}(t)$ based on the first transition from state $v_{n,i}$ as

$$\hat{\boldsymbol{g}}_{ij}(t) = -\boldsymbol{L}_{ii} \mathrm{e}^{\boldsymbol{L}_{ii}t} \frac{\boldsymbol{B}_{ij}}{-\boldsymbol{L}_{ii}} + \int_{\tau=0}^{t} -\boldsymbol{L}_{ii} \mathrm{e}^{\boldsymbol{L}_{ii}\tau} \left(\sum_{\substack{k\in\Phi,\\ k\neq i}} \frac{\boldsymbol{L}_{ik}}{-\boldsymbol{L}_{ii}} \hat{\boldsymbol{g}}_{kj}(t-\tau) + \sum_{m=1}^{\infty} \sum_{\substack{k\in\Phi,\\ k\neq i}} \frac{\boldsymbol{F}_{\boldsymbol{m}ik}}{-\boldsymbol{L}_{ii}} \hat{\boldsymbol{g}}_{\boldsymbol{m+1}kj}(t-\tau) \right) d\tau. \quad (42)$$

The Laplace transform of (42) gives

$$\hat{\boldsymbol{g}}^{*}{}_{ij}(s) = \frac{-\boldsymbol{L}_{ii}}{s-\boldsymbol{L}_{ii}}\left(\frac{\boldsymbol{B}_{ij}}{-\boldsymbol{L}_{ii}} + \sum_{\substack{k\in\Phi,\\ k\neq i}} \frac{\boldsymbol{L}_{ik}}{-\boldsymbol{L}_{ii}}\hat{\boldsymbol{g}}^{*}{}_{kj}(s) + \sum_{m=1}^{\infty}\sum_{k\in\Phi}\frac{\boldsymbol{F}_{\boldsymbol{m}ik}}{-\boldsymbol{L}_{ii}}\hat{\boldsymbol{g}}^{*}_{\boldsymbol{m+1}_{kj}}(s)\right).$$

Multiplying both sides by $s-\boldsymbol{L}_{ii}$ and adding $\boldsymbol{L}_{ii}\hat{\boldsymbol{g}}^{*}{}_{ii}(s)$ we obtain

$$s\hat{\boldsymbol{g}}^{*}{}_{ij}(s) = \boldsymbol{B}_{ij} + \sum_{k\in\Phi}\frac{\boldsymbol{L}_{ik}}{-\boldsymbol{L}_{ii}}\hat{\boldsymbol{g}}^{*}{}_{kj}(s) + \sum_{m=1}^{\infty}\sum_{k\in\Phi}\frac{\boldsymbol{F}_{\boldsymbol{m}ik}}{-\boldsymbol{L}_{ii}}\hat{\boldsymbol{g}}^{*}_{\boldsymbol{m+1}_{kj}}(s),$$

which can be written in matrix form, using $\hat{\boldsymbol{g}}^{*}_{\boldsymbol{m}}(s) = \hat{\boldsymbol{g}}^{*}(s)^{m}$, as

$$s\hat{\boldsymbol{g}}^{*}(s) = \boldsymbol{B} + \boldsymbol{L}\hat{\boldsymbol{g}}^{*}(s) + \sum_{m=1}^{\infty}\boldsymbol{F}_{\boldsymbol{m}}\hat{\boldsymbol{g}}^{*}(s)^{m+1}. \tag{43}$$

Similar to the case of $\boldsymbol{g}^{*}(s)$, using the final value theorem we have that

$$\begin{aligned}\hat{\boldsymbol{G}}_{ij} &\overset{def}{=} Pr(X(\rho_{n-1}) = v_{n-1,j}|X(0)=v_{n,i})\\ &= \lim_{t\to\infty} G(t) = \int_{t=0}^{\infty}\hat{\boldsymbol{g}}_{ij}(t)dt = \lim_{s\to 0}\hat{\boldsymbol{g}}^{*}{}_{ij}(s),\end{aligned}$$

i.e., $\hat{\boldsymbol{G}}_{ij}$ is the probability that the process, starting from state i of level n reaches level $n-1$ ($n>2$) in state j. Substituting $s=0$ into (43) we get

$$\boldsymbol{0} = \boldsymbol{B} + \boldsymbol{L}\hat{\boldsymbol{G}} + \sum_{m=1}^{\infty}\boldsymbol{F}_{\boldsymbol{m}}\hat{\boldsymbol{G}}^{m+1}, \tag{44}$$

Which is the well-known matrix equation for computing the characteristic matrix of an M/G/1 type process [15]. Similar to matrix $\boldsymbol{A}$ we define

$$\hat{\boldsymbol{A}}_{ij} \overset{def}{=} \mathrm{E}[\rho_{n-1}I_{\{X(\rho_{n-1})=j\}}|X(0)=i] = -\frac{d}{ds}\hat{\boldsymbol{g}}^{*}{}_{ij}(s)\big|_{s=0},$$

which we obtain from the moment generating property of the Laplace transform.

Taking the derivative of (43) according to s in $s=0$ we get

$$\hat{\boldsymbol{G}} = \boldsymbol{L}\hat{\boldsymbol{A}} + \sum_{m=1}^{\infty}\boldsymbol{F}_{\boldsymbol{m}}\sum_{\ell=0}^{m}\hat{\boldsymbol{G}}^{\ell}\hat{\boldsymbol{A}}\hat{\boldsymbol{G}}^{m-\ell}, \tag{45}$$

which is the linear equation for computing $\hat{\boldsymbol{A}}$ based on $\hat{\boldsymbol{G}}$.

To obtain the matrix of mean accumulated rewards till the first visit to level $n-m$, starting from level n, similar to $\boldsymbol{K}(r)$ we define $\hat{\boldsymbol{K}}_{\boldsymbol{n,m}}(r)$ by its i,j element as

$$\hat{\boldsymbol{K}}_{\boldsymbol{n,m}_{ij}}(r) = Pr\left(X(\rho_{n-m}) = v_{n-m,j}, \int_{t=0}^{\rho_{n-m}} \boldsymbol{C}_{X(t),X(t)} dt < r | X(0) = v_{n,i}\right), \tag{46}$$

that is, $\hat{\boldsymbol{K}}_{\boldsymbol{n,m}_{ij}}(r)$ is the probability that the process starting from state i of level n will visit level $n-m$ before reward r is accumulated and the first visit to level $n-m$ will be in state j. We also define $\hat{\boldsymbol{k}}_{\boldsymbol{n,m}_{ij}}(r) = \frac{d}{dr}\hat{\boldsymbol{K}}_{\boldsymbol{n,m}_{ij}}(r)$. Element $\hat{\boldsymbol{k}}_{\boldsymbol{n,1}_{ij}}(r)$ can be expressed very similar to $\boldsymbol{k}_{ij}(r)$ in (21). Using notation $\hat{\boldsymbol{k}}_{\boldsymbol{n}}(r) = \hat{\boldsymbol{k}}_{\boldsymbol{n,1}}(r)$ and $\hat{\boldsymbol{k}}_{\boldsymbol{n,m}}(r) = (\hat{\boldsymbol{k}}_{\boldsymbol{n,n-1}} * \hat{\boldsymbol{k}}_{\boldsymbol{n-1,m-1}})(r),\ \forall n \geq m+1, m \geq 1$ we can write

$$\hat{\boldsymbol{k}}_{\boldsymbol{n,1}_{ij}}(r) = \frac{-\boldsymbol{L}_{ii}}{\boldsymbol{C}_{\boldsymbol{n}ii}} \mathrm{e}^{\frac{\boldsymbol{L}_{ii}}{\boldsymbol{C}_{\boldsymbol{n}ii}} r} \frac{\boldsymbol{B}_{ij}}{-\boldsymbol{L}_{ii}} + \int_{u=0}^{r} \frac{-\boldsymbol{L}_{ii}}{\boldsymbol{C}_{\boldsymbol{n}ii}} \mathrm{e}^{\frac{\boldsymbol{L}_{ii}}{\boldsymbol{C}_{\boldsymbol{n}ii}} u} \Bigg(\sum_{\substack{k\in\Phi,\\ k\neq i}} \frac{\boldsymbol{L}_{ik}}{-\boldsymbol{L}_{ii}} \hat{\boldsymbol{k}}_{\boldsymbol{n,1}_{kj}}(r-u)$$

$$+ \sum_{k\in\Phi} \sum_{m=1}^{\infty} \frac{\boldsymbol{F}_{\boldsymbol{m}ik}}{-\boldsymbol{L}_{ii}} \hat{\boldsymbol{k}}_{\boldsymbol{n+m,m+1}_{kj}}(r-u) \Bigg) du.$$

Taking the Laplace transform of the above equation and rearranging the result in a similar manner as before, we obtain

$$s\boldsymbol{C}_{\boldsymbol{n}} \hat{\boldsymbol{k}}^*_{\boldsymbol{n}}(s) = \boldsymbol{B} + \boldsymbol{L}\hat{\boldsymbol{k}}^*_{\boldsymbol{n}}(s) + \sum_{m=1}^{\infty} \boldsymbol{F}_{\boldsymbol{m}} \hat{\boldsymbol{k}}^*_{\boldsymbol{n+m,m+1}}(s), \tag{47}$$

where

$$\hat{\boldsymbol{k}}^*_{\boldsymbol{n+m,m+1}}(s) = \prod_{k=n+m}^{n} \hat{\boldsymbol{k}}^*_{\boldsymbol{k}}(s) = \hat{\boldsymbol{k}}^*_{\boldsymbol{n+m}}(s) \hat{\boldsymbol{k}}^*_{\boldsymbol{n+m-1}}(s) \ldots \hat{\boldsymbol{k}}^*_{\boldsymbol{n+1}}(s) \hat{\boldsymbol{k}}^*_{\boldsymbol{n}}(s).$$

Similar to $\lim_{s\to 0} \boldsymbol{k}^*(s) = \boldsymbol{G}$, we have $\lim_{s\to 0} \hat{\boldsymbol{k}}^*_{\boldsymbol{n}}(s) = \int_{r=0}^{\infty} \hat{\boldsymbol{k}}_{\boldsymbol{n}}(r) dr = Pr(X(\rho_{n-1}) = v_{n-1,j} | X(0) = v_{n,i}) = \hat{\boldsymbol{G}}$. As a consequence, $\lim_{s\to 0} \hat{\boldsymbol{k}}^*_{\boldsymbol{n}}(s)$ is level independent even though $\hat{\boldsymbol{k}}^*_{\boldsymbol{n}}(s)$ is level dependent.

Using the moment generating property of the Laplace transform we define

$$\hat{\boldsymbol{M}}_{\boldsymbol{n}ij} \overset{def}{=} -\frac{d}{ds} \hat{\boldsymbol{k}}^*_{\boldsymbol{n}ij}(s)\big|_{s=0} = \mathrm{E}\left[I_{\{X(\rho_{n-1})=v_{n-1,j}\}} \int_{t=0}^{\rho_{n-1}} \boldsymbol{C}_{X(t),X(t)} dt \middle| X(0) = v_{n,i} \right].$$

The derivative of (47) at $s=0$ gives

$$\boldsymbol{C}_{\boldsymbol{n}} \hat{\boldsymbol{G}} = \boldsymbol{L}\hat{\boldsymbol{M}}_{\boldsymbol{n}} + \sum_{m=1}^{\infty} \boldsymbol{F}_{\boldsymbol{m}} \sum_{\ell=0}^{m} \hat{\boldsymbol{G}}^{\ell} \hat{\boldsymbol{M}}_{\boldsymbol{n+m-\ell}} \hat{\boldsymbol{G}}^{m-\ell}, \tag{48}$$

where we used that $\hat{\boldsymbol{G}} = \lim\limits_{s\to 0} \hat{\boldsymbol{k}}^*_{\boldsymbol{n}}(s)$ and $\hat{\boldsymbol{M}}_{\boldsymbol{n}} = -\frac{d}{ds}\hat{\boldsymbol{k}}^*_{\boldsymbol{n}}(s)\big|_{s=0}$. This equation has to be solved to obtain the required $\hat{\boldsymbol{M}}_{\boldsymbol{n}}$ matrices for $n = 1, 2 \ldots$.

The level dependence of $\boldsymbol{C}_{\boldsymbol{n}}$ makes $\hat{\boldsymbol{M}}_{\boldsymbol{n}}$ level dependent as well, while $\hat{\boldsymbol{A}}$ is level independent. To separate the level dependent and level independent elements of $\hat{\boldsymbol{M}}_{\boldsymbol{n}}$, for $i, j, \ell \in \Phi$, $n \geq 1$, $m \geq 0$ we introduce

$$\hat{\boldsymbol{T}}_{\boldsymbol{m}ij\ell} \overset{def}{=} \mathrm{E}\left[I_{\{X(\rho_{n-1})=v_{n-1,j}\}} \int_{t=0}^{\rho_{n-1}} I_{\{X(t)=v_{n+m,\ell}\}} dt \,\middle|\, X(0) = v_{n,i} \right]. \qquad (49)$$

Due to the spatial homogeneity of the M/G/1 type process $\hat{\boldsymbol{T}}_{\boldsymbol{m}ij\ell}$ is level independent, that is, it does not depend on n. Based on $\hat{\boldsymbol{T}}_{\boldsymbol{m}ij\ell}$, $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{M}}_{\boldsymbol{n}}$ can be obtained as

$$\hat{\boldsymbol{A}}_{ij} = \sum_{m=0}^{\infty} \sum_{\ell\in\Phi} \hat{\boldsymbol{T}}_{\boldsymbol{m}ij\ell} \quad \text{and} \quad \hat{\boldsymbol{M}}_{\boldsymbol{n}ij} = \sum_{m=0}^{\infty} \sum_{\ell\in\Phi} \hat{\boldsymbol{T}}_{\boldsymbol{m}ij\ell} \boldsymbol{C}_{\boldsymbol{n+m}\ell\ell}. \qquad (50)$$

The next task is to compute the global S_D related measures $\boldsymbol{G}$, $\boldsymbol{A}$, and $\boldsymbol{M}$ from the level related measures $\hat{\boldsymbol{G}}$, $\hat{\boldsymbol{A}}$, and $\hat{\boldsymbol{M}}_{\boldsymbol{n}}$. For states $i' = v_{m+1,i}$ and $j' = v_{1,j}$, we have that

$$\boldsymbol{g}^*(s)_{i'j'} = \hat{\boldsymbol{g}}^*_{\boldsymbol{m}}(s)_{ij} = \left[\hat{\boldsymbol{g}}^*(s)^m\right]_{ij},$$

and

$$\boldsymbol{k}^*(s)_{i'j'} = \hat{\boldsymbol{k}}^*_{\boldsymbol{m+1,m}}(s)_{ij} = \left[\prod_{k=m+1}^{2} \hat{\boldsymbol{k}}^*_{\boldsymbol{k}}(s)\right]_{ij}.$$

Using these relations of the transform we obtain

$$\boldsymbol{G}_{ij} = \lim_{s\to 0} \boldsymbol{g}^*(s)_{i'j'} = \lim_{s\to 0} \left[\hat{\boldsymbol{g}}^*(s)^m\right]_{ij} = [\hat{\boldsymbol{G}}^m]_{ij},$$

$$\begin{aligned}\boldsymbol{A}_{ij} = \frac{d}{ds}\boldsymbol{g}^*(s)_{i'j'}\big|_{s=0} = \frac{d}{ds}\hat{\boldsymbol{g}}^*_{\boldsymbol{m}}(s)_{ij}\big|_{s=0} = \left[\frac{d}{ds}\hat{\boldsymbol{g}}^*(s)^m\big|_{s=0}\right]_{ij} = \\ \left[\sum_{\ell=0}^{m-1} \hat{\boldsymbol{g}}^*(s)^\ell \frac{d}{ds}\hat{\boldsymbol{g}}^*(s) \hat{\boldsymbol{g}}^*(s)^{m-1-\ell}\big|_{s=0}\right]_{ij} = \left[\sum_{\ell=0}^{m-1} \hat{\boldsymbol{G}}^\ell \hat{\boldsymbol{A}} \hat{\boldsymbol{G}}^{m-1-\ell}\right]_{ij},\end{aligned}$$

and

$$\boldsymbol{M}_{i'j'} = \frac{d}{ds}\boldsymbol{k}^*(s)_{i'j'}\big|_{s=0} = \left[\sum_{\ell=0}^{m-1} \hat{\boldsymbol{G}}^\ell \hat{\boldsymbol{M}}_{\boldsymbol{m+1-\ell}} \hat{\boldsymbol{G}}^{m-1-\ell}\right]_{ij},$$

where the derivation of $\boldsymbol{M}_{ij}$ follows the same patterns as the one of $\boldsymbol{A}_{ij}$ and we used that $\lim\limits_{s\to 0} \hat{\boldsymbol{k}}^*_{\boldsymbol{k}}(s) = \hat{\boldsymbol{G}}$ is independent of level k.

Substituting $\boldsymbol{G}$, $\boldsymbol{A}$ and $\boldsymbol{M}$ into (12), (13) and (14), for the compressed process we obtain (36), (37) and (38). □

6.2 The G/M/1 type process

An MDP is of G/M/1 type with the considered $\{S_U, S_D\}$ partitioning if its generator matrix has the following block structure

$$\boldsymbol{Q}(\pi) = \left(\begin{array}{c|ccccc} \bar{\boldsymbol{L}}(\pi) & \bar{\boldsymbol{F}}(\pi) & & & & \\ \hline \bar{\boldsymbol{B}}_{\boldsymbol{1}} & \boldsymbol{L} & \boldsymbol{F} & & & \\ \bar{\boldsymbol{B}}_{\boldsymbol{2}} & \boldsymbol{B}_{\boldsymbol{1}} & \boldsymbol{L} & \boldsymbol{F} & & \\ \vdots & & \ddots & \ddots & \ddots & \ddots \end{array}\right). \tag{51}$$

The MDP can be partitioned to levels according to blocks so that each matrix block row in (51) corresponds to a level. In the following we assume that the reward rate matrix also has some level based regularity, thus

$$\boldsymbol{C}(\pi) = \left(\begin{array}{c|ccc} \boldsymbol{C}_{\boldsymbol{1}}(\pi) & & & \\ \hline & \boldsymbol{C}_{\boldsymbol{2}} & & \\ & & \boldsymbol{C}_{\boldsymbol{3}} & \\ & & & \ddots \end{array}\right), \tag{52}$$

where $\boldsymbol{C}_{\boldsymbol{n}} = f(\boldsymbol{C}_{\boldsymbol{0}}, n), \forall n \geq 2$. We discuss some specific cases later in Section 7.1.

Theorem 4. *For an MDP with G/M/1 type structure in S_D, according to* (51) *and* (52) *the parameters of the reduced MDP representation are*

$$\boldsymbol{P}(\pi) = \left(-\textit{\textbf{diagm}}\langle \boldsymbol{Q}_{\boldsymbol{c}}(\pi)\rangle\right)^{-1} \left(\boldsymbol{Q}_{\boldsymbol{c}}(\pi) - \textit{\textbf{diagm}}\langle \boldsymbol{Q}_{\boldsymbol{c}}(\pi)\rangle\right), \tag{53}$$

$$\tau(\pi) = \left(-\textit{\textbf{diagm}}\langle \boldsymbol{Q}_{\boldsymbol{c}}(\pi)\rangle\right)^{-1} \left(\boldsymbol{I} + \sum_{m=1}^{\infty} \hat{\boldsymbol{R}}_{\boldsymbol{1}}(\pi)\boldsymbol{R}^{m-1}\right) \underline{\boldsymbol{1}}, \tag{54}$$

$$c(\pi) = \left(-\textit{\textbf{diagm}}\langle \boldsymbol{Q}_{\boldsymbol{c}}(\pi)\rangle\right)^{-1} \left(\boldsymbol{C}_{\boldsymbol{1}}(\pi) + \sum_{m=1}^{\infty} \hat{\boldsymbol{R}}_{\boldsymbol{1}}(\pi)\boldsymbol{R}^{m-1}\boldsymbol{C}_{\boldsymbol{m+1}}\right) \underline{\boldsymbol{1}}, \tag{55}$$

where $\boldsymbol{Q}_{\boldsymbol{c}}(\pi) = \bar{\boldsymbol{L}}(\pi) + \sum_{m=1}^{\infty} \hat{\boldsymbol{R}}_{\boldsymbol{1}}(\pi)\boldsymbol{R}^{m-1}\bar{\boldsymbol{B}}_{\boldsymbol{m}}$*,*

$$\hat{\boldsymbol{R}}_{\boldsymbol{1}}(\pi) = \bar{\boldsymbol{F}}(\pi)\left(-\boldsymbol{L} - \sum_{m=1}^{\infty} \boldsymbol{R}^m \boldsymbol{B}_{\boldsymbol{m}}\right)^{-1}$$

and $\boldsymbol{R}$ *is the solution of*

$$\boldsymbol{0} = \boldsymbol{F} + \boldsymbol{R}\boldsymbol{L} + \sum_{m=1}^{\infty} \boldsymbol{R}^{m+1}\boldsymbol{B}_{\boldsymbol{m}}.$$

Remark. *Matrix $\boldsymbol{R}$ is the well studied characteristic matrix of the G/M/1 type process [13]. Efficient numerical methods are available for its computation e.g., in [4]. Unlike* (38)*, the $\boldsymbol{C_n}$ matrices appear in* (55) *directly, which makes the computation of the reward term much simpler for the G/M/1 type process, because in this case there is no need to compute $\hat{\boldsymbol{M}}_{\boldsymbol{n}}$ from* (41).

Proof. Let η_n be the time of the first visit to a level equal to or lower than n, that is, $\eta_n = \min_{k\in\{0,1,\dots,n\}}(\rho_k)$, furthermore we define the $\boldsymbol{V}(t)$ matrix function whose ij element is

$$\boldsymbol{V}_{ij}(t) = Pr(X(t) = v_{n,j}, \eta_{n-1} > t | X(0) = v_{n,i}), \forall n > 1.$$

That is, $\boldsymbol{V}_{ij}(t)$ is the probability that the process, assuming that it starts in state i of level n, visits state j of level n at time t such that it does not visit any lower level before t. Furthermore, we define $\boldsymbol{R}(t) = \boldsymbol{F}\boldsymbol{V}(t)$. A stochastic interpretation of its i, j element is

$$\boldsymbol{R}_{ij}(t) = \lim_{\Delta\to 0} \frac{1}{\Delta} Pr(X(t) = v_{n+1,j}, \eta_n > t, X(\Delta) \neq v_{n,i} | X(0) = v_{n,i}).$$

We define the multi level version of $\boldsymbol{R}(t)$ as

$$\boldsymbol{R}_{\boldsymbol{m}ij}(t) = \lim_{\Delta\to 0} \frac{1}{\Delta} Pr(X(t) = v_{n+m,j}, \eta_n > t, X(\Delta) \neq v_{n,i} | X(0) = v_{n,i}).$$

Starting from level n and being at level $n + m$ at time t, let τ be the *last* instance when the process is at level $n + 1$ before time t. Then

$$\boldsymbol{R}_{\boldsymbol{m}ij}(t) = \sum_k \int_{\tau=0}^{t} \boldsymbol{R}_{ik}(\tau) \boldsymbol{R}_{\boldsymbol{m-1}kj}(t-\tau) d\tau$$

That is, for $m > 1$, $\boldsymbol{R}_{\boldsymbol{m}}(t) = \boldsymbol{R}(t) * \boldsymbol{R}_{\boldsymbol{m-1}}(t)$, with $\boldsymbol{R}_{\boldsymbol{1}}(t) = \boldsymbol{R}(t)$ and $*$ denoting the convolution operator.

To evaluate $\boldsymbol{R}_{\boldsymbol{m}}(t)$ we first compute $\boldsymbol{V}(t)$. We express $\boldsymbol{V}_{ij}(t)$ using the law of total probability as the event $X(t) = v_{n,j} | X(0) = v_{n,i}$ can partitioned the following way:

a) The first transition happens after time t and $i = j$. The probability of this is $\delta_{ij} \int_{h=t}^{\infty} -\boldsymbol{L}_{ii} e^{\boldsymbol{L}_{ii} h} dh = \delta_{ij} e^{\boldsymbol{L}_{ii} t}$.

b) The first transition happens before time t and this transition is inside level n. The probability of this is $\int_{h=0}^{t} -\boldsymbol{L}_{ii} e^{\boldsymbol{L}_{ii} h} \sum_{\substack{k\in\Phi\\ k\neq i}} \frac{\boldsymbol{L}_{ik}}{-\boldsymbol{L}_{ii}} \boldsymbol{V}_{kj}(t-h) dh$.

c) The first transition happens before time t and this transition is to level $n+1$.

We can break down c) further according to the last level visited by the process before returning to level n for the first time. The probability that this level is $n+m$ (where $m>0$, and, to avoid special treatment of $m=1$, assuming $(\boldsymbol{V}*\boldsymbol{R_0})(t)=\boldsymbol{V}(t)$) is

$$\int_{h=0}^{t} -\boldsymbol{L}_{ii}\mathrm{e}^{\boldsymbol{L}_{ii}h}\sum_{k\in\Phi}\frac{\boldsymbol{F}_{ik}}{-\boldsymbol{L}_{ii}}\big[(\boldsymbol{V}*\boldsymbol{R_{m-1}}\boldsymbol{B_m}*\boldsymbol{V})(t-h)\big]_{kj}dh$$
$$=\int_{h=0}^{t}\mathrm{e}^{\boldsymbol{L}_{ii}h}\big[(\boldsymbol{R_m}\boldsymbol{B_m}*\boldsymbol{V})(t-h)\big]_{ij}dh.$$

Combining a), b), and c) we have

$$\boldsymbol{V}_{ij}(t)=\underbrace{\delta_{ij}\mathrm{e}^{\boldsymbol{L}_{ii}t}}_{a)}+\underbrace{\int_{h=0}^{t}\mathrm{e}^{\boldsymbol{L}_{ii}h}\sum_{\substack{k\in\Phi\\k\neq i}}\boldsymbol{L}_{ik}\boldsymbol{V}_{kj}(t-h)dh}_{b)}$$
$$\underbrace{+\int_{h=0}^{t}\mathrm{e}^{\boldsymbol{L}_{ii}h}\sum_{m=1}^{\infty}\left[(\boldsymbol{R_m}\boldsymbol{B_m}*\boldsymbol{V})(t-h)\right]_{ij}dh}_{c)}$$

Laplace transforming the above equation and multiplying by $s-\boldsymbol{L}_{ii}$ we get

$$(s-\boldsymbol{L}_{ii})\boldsymbol{V}^*_{ij}(s)=\delta_{ij}+\sum_{\substack{k\in\Phi\\k\neq i}}\boldsymbol{L}_{ik}\boldsymbol{V}^*_{kj}(s)+\left[\boldsymbol{R}^*_{\boldsymbol{m}}(s)\boldsymbol{B_m}\boldsymbol{V}^*(s)\right]_{ij}.$$

After adding $\boldsymbol{L}_{ii}\boldsymbol{V}^*_{ii}(s)$ to both sides we can write the equation in matrix form as

$$s\boldsymbol{V}^*(s)=\boldsymbol{I}+\boldsymbol{L}\boldsymbol{V}^*(s)+\sum_{m=1}^{\infty}\boldsymbol{R}^*_{\boldsymbol{m}}(s)\boldsymbol{B_m}\boldsymbol{V}^*(s),$$

from which

$$\boldsymbol{V}^*(s)=\left(s\boldsymbol{I}-\boldsymbol{L}-\sum_{m=1}^{\infty}\boldsymbol{R}^*(s)^m\boldsymbol{B_m}\right)^{-1}, \qquad (56)$$

using $\boldsymbol{R}^*_{\boldsymbol{m}}(s)=\boldsymbol{R}^*(s)^m$. Multiplying it with $\boldsymbol{F}$ from the left side we get

$$\boldsymbol{F}\boldsymbol{V}^*(s)=\boldsymbol{R}^*(s)=\boldsymbol{F}\left(s\boldsymbol{I}-\boldsymbol{L}-\sum_{m=1}^{\infty}\boldsymbol{R}^*(s)^m\boldsymbol{B_m}\right)^{-1}.$$

By multiplying with the term in the parentheses from the left and rearranging the equation we obtain

$$s\boldsymbol{R}^*(s) = \boldsymbol{F} + \boldsymbol{R}^*(s)\boldsymbol{L} + \sum_{m=1}^{\infty} \boldsymbol{R}^*(s)^{m+1}\boldsymbol{B_m}.$$

At $s \to 0$ this becomes

$$\boldsymbol{0} = \boldsymbol{F} + \boldsymbol{RL} + \sum_{m=1}^{\infty} \boldsymbol{R}^{m+1}\boldsymbol{B_m}, \tag{57}$$

where $\boldsymbol{R} = \int_{\tau=0}^{\infty} \boldsymbol{R}(\tau)d\tau = \lim_{s\to 0} \boldsymbol{R}(s)$. Equation (57) is the well-known matrix equation to obtain matrix $\boldsymbol{R}$, the characteristic matrix of the G/M/1 type process [13]. If the process starts in the irregular first level, then the same methodology can be applied, but

$$\begin{aligned}\hat{\boldsymbol{V}}_{ij}(\pi, t) = \delta_{ij} \int_{h=t}^{\infty} &-\bar{\boldsymbol{L}}_{ii}(\pi)\mathrm{e}^{\bar{\boldsymbol{L}}_{ii}(\pi)h} dh \\ &+ \int_{h=0}^{t} -\bar{\boldsymbol{L}}_{ii}(\pi)\mathrm{e}^{\bar{\boldsymbol{L}}_{ii}(\pi)h} \sum_{\substack{k\in\Phi \\ k\neq i}} \frac{\bar{\boldsymbol{L}}_{ik}(\pi)}{-\bar{\boldsymbol{L}}_{ii}(\pi)} \hat{\boldsymbol{V}}_{kj}\pi, t-h)dh \\ &\qquad + \int_{h=0}^{t} \mathrm{e}^{\bar{\boldsymbol{L}}_{ii}(\pi)h} \sum_{m=1}^{\infty} \left[(\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi)\bar{\boldsymbol{B}}_{\boldsymbol{m}} * \hat{\boldsymbol{V}}(\pi))(t-h) \right]_{ij},\end{aligned}$$

where $\hat{\boldsymbol{R}}(\pi, t) = \bar{\boldsymbol{F}}(\pi)\boldsymbol{V}(t)$, $\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi, t) = (\hat{\boldsymbol{R}}(\pi) * \boldsymbol{R_{m-1}})(t)$, $\forall m > 1$. Using the same steps as before we obtain

$$s\hat{\boldsymbol{R}}^*(\pi, s) = \bar{\boldsymbol{F}}(\pi) + \hat{\boldsymbol{R}}^*(\pi, s)\bar{\boldsymbol{L}} + \sum_{m=1}^{\infty} \hat{\boldsymbol{R}}_{\boldsymbol{m+1}}{}^*(\pi, s)\boldsymbol{B_m}. \tag{58}$$

where $\hat{\boldsymbol{R}}_{\boldsymbol{m+1}}{}^*(\pi, s) = \hat{\boldsymbol{R}}^*(\pi, s)\boldsymbol{R}^*_{\boldsymbol{m}}(s) = \hat{\boldsymbol{R}}^*(\pi, s)\boldsymbol{R}^*(s)^m$, from which at $s \to 0$ (58) becomes

$$\boldsymbol{0} = \bar{\boldsymbol{F}}(\pi) + \hat{\boldsymbol{R}}(\pi)\boldsymbol{L} + \sum_{m=1}^{\infty} \hat{\boldsymbol{R}}(\pi)\boldsymbol{R}^m\boldsymbol{B_m}, \tag{59}$$

where $\hat{\boldsymbol{R}}(\pi) = \lim_{s\to 0} \hat{\boldsymbol{R}}^*(\pi, s)$ and $\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi) = \lim_{s\to 0} \hat{\boldsymbol{R}}_{\boldsymbol{m}}{}^*(\pi, s)$. By rearranging (59) we get that

$$\hat{\boldsymbol{R}}(\pi) = \bar{\boldsymbol{F}}(\pi) \left(-\boldsymbol{L} - \sum_{m=1}^{\infty} \boldsymbol{R}^m\boldsymbol{B_m} \right)^{-1}, \quad \hat{\boldsymbol{R}}_{\boldsymbol{m+1}}(\pi) = \hat{\boldsymbol{R}}(\pi)\boldsymbol{R}^m. \tag{60}$$

For G/M/1 type processes we cannot adopt the approach used for the M/G/1 type process. In this case, we directly express the $\boldsymbol{P}(\pi), \tau(\pi)$, and $c(\pi)$ parameters.

From the definition of $\boldsymbol{R_m}$, $\frac{1}{-\boldsymbol{L}_{ii}}\boldsymbol{R_m}_{ij}$ is the mean time spent in $v_{n+m,j}$ until the process visits some level below n, starting from $v_{n,i}$. For $\boldsymbol{P}(\pi)$ we use the same decomposition as for the general case in Section 4, where we had the following cases:

- Case 1: the process moves to state j ($j \in S_U \setminus i$) directly.
- Case 2: the process first moves to $k \in S_D$, spends some time in S_D, then enters S_U in state j.
- Case 3: the process first moves to $k \in S_D$, spends some time in S_D, then enters S_U in state i. This case adds a recursive term to the formulas.

Using this decomposition we can write, for $i \neq j$:

$$\boldsymbol{P}_{ij}(\pi) = \underbrace{\frac{\bar{\boldsymbol{L}}_{ij}(\pi)}{-\bar{\boldsymbol{L}}_{ii}(\pi)}}_{Case\ 1} + \underbrace{\sum_{m=1}^{\infty}\sum_{k\in\Phi}\frac{1}{-\bar{\boldsymbol{L}}_{ii}(\pi)}\hat{\boldsymbol{R}}_{\boldsymbol{m}ik}(\pi)\bar{\boldsymbol{B}}_{\boldsymbol{m}kj}}_{Case\ 2}$$
$$+ \underbrace{\sum_{m=1}^{\infty}\sum_{k\in\Phi}\frac{1}{-\bar{\boldsymbol{L}}_{ii}(\pi)}\hat{\boldsymbol{R}}_{\boldsymbol{m}ik}(\pi)\bar{\boldsymbol{B}}_{\boldsymbol{m}ki}\boldsymbol{P}_{ij}(\pi)}_{Case\ 3},$$

and $\boldsymbol{P}_{ii}(\pi) = 0$. That is, for $i \neq j$

$$-\left(\bar{\boldsymbol{L}}_{ii}(\pi) + \sum_{m=1}^{\infty}[\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi)\bar{\boldsymbol{B}}_{\boldsymbol{m}}]_{ii}\right)\boldsymbol{P}_{ij}(\pi) \;=\; \bar{\boldsymbol{L}}_{ij}(\pi) + \sum_{m=1}^{\infty}[\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi)\bar{\boldsymbol{B}}_{\boldsymbol{m}}]_{ij},$$

which can be written in matrix from as

$$\boldsymbol{P}(\pi) = \left(-\mathbf{diagm}\langle\boldsymbol{Q_c}(\pi)\rangle\right)^{-1}\left(\boldsymbol{Q_c}(\pi) - \mathbf{diagm}\langle\boldsymbol{Q_c}(\pi)\rangle\right),$$

where $\boldsymbol{Q_c}(\pi) = \bar{\boldsymbol{L}}(\pi) + \sum_{m=1}^{\infty}\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi)\bar{\boldsymbol{B}}_{\boldsymbol{m}}$. The subtraction of the diagonal matrix in the second term ensures that the diagonal of $\boldsymbol{P}(\pi)$ is zero. Using $\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi) = \hat{\boldsymbol{R}}_{\boldsymbol{1}}(\pi)\boldsymbol{R}^{m-1}$, we can also write $\boldsymbol{Q_c}(\pi) = \bar{\boldsymbol{L}}(\pi) + \sum_{m=1}^{\infty}\hat{\boldsymbol{R}}_{\boldsymbol{1}}(\pi)\boldsymbol{R}^{m-1}\bar{\boldsymbol{B}}_{\boldsymbol{m}}$.

To compute $\tau(\pi)$, let $\gamma_{S_U\setminus i}$ be the first time to reach $S_U \setminus i$, furthermore let $\tau_i(\pi,t) = Pr(\gamma_{S_U\setminus i} > t|X(0) = i)$ and $\tau_i^*(\pi,s) = \int_t \tau_i(\pi,t)e^{-st}dt$. For $\tau_i^*(\pi,s)$

we have

$$\begin{aligned}
\tau_i^*(\pi,s) = \quad & \underbrace{\frac{1}{s-\bar{\boldsymbol{L}}_{ii}(\pi)}}_{\text{no transition till } t} \\
& + \sum_{m=1}^{\infty}\sum_{k\in\Phi}\sum_{\ell\in\Phi} \quad \underbrace{\frac{-\bar{\boldsymbol{L}}_{ii}(\pi)}{s-\bar{\boldsymbol{L}}_{ii}(\pi)}}_{\text{transition at } x(<t)} \quad \frac{\bar{\boldsymbol{F}}_{ik}(\pi)}{-\bar{\boldsymbol{L}}_{ii}(\pi)} \underbrace{\left(\boldsymbol{V}^*(s)\boldsymbol{R}^*_{\boldsymbol{m-1}}(s)\right)_{k\ell}}_{\text{time in } S_D > t-x} \\
& + \sum_{m=1}^{\infty}\sum_{k\in\Phi} \underbrace{\frac{-\bar{\boldsymbol{L}}_{ii}(\pi)}{s-\bar{\boldsymbol{L}}_{ii}(\pi)}}_{\text{tr. at } x} \frac{\bar{\boldsymbol{F}}_{ik}(\pi)}{-\bar{\boldsymbol{L}}_{ii}(\pi)} \quad \underbrace{\left(\boldsymbol{V}^*(s)\boldsymbol{R}^*_{\boldsymbol{m-1}}(s)\bar{\boldsymbol{B}}_{\boldsymbol{m}}\right)_{ki}}_{\text{return to } i \text{ at } y \ (x<y<t)} \quad \underbrace{\tau_i^*(\pi,s)}_{\text{time to } S_U \setminus i > t-y} \quad .
\end{aligned}$$

Multiplying with $s-\bar{\boldsymbol{L}}_{ii}(\pi)$ gives

$$\begin{aligned}
(s-\bar{\boldsymbol{L}}_{ii}(\pi))\tau_i^*(\pi,s) = 1 + & \sum_{m=1}^{\infty}\sum_{k\in\Phi}\sum_{\ell\in\Phi}\bar{\boldsymbol{F}}_{ik}(\pi)\left(\boldsymbol{V}^*(s)\boldsymbol{R}^*_{\boldsymbol{m-1}}(s)\right)_{k\ell} \\
& + \sum_{m=1}^{\infty}\sum_{k\in\Phi}\bar{\boldsymbol{F}}_{ik}(\pi)\left(\boldsymbol{V}^*(s)\boldsymbol{R}^*_{\boldsymbol{m-1}}(s)\bar{\boldsymbol{B}}_{\boldsymbol{m}}\right)_{ki}\tau_i^*(\pi,s) = \\
& \qquad 1 + \sum_{m=1}^{\infty}\sum_{\ell\in\Phi}\hat{\boldsymbol{R}}^*_{\boldsymbol{m}}(\pi,s)_{i\ell} + \sum_{m=1}^{\infty}\left(\hat{\boldsymbol{R}}^*_{\boldsymbol{m}}(\pi,s)\bar{\boldsymbol{B}}_{\boldsymbol{m}}\right)_{ii}\tau_i^*(\pi,s).
\end{aligned}$$

We are interested in the mean time to get to $S_U \setminus i$ which is $\int_t \tau_i(\pi,t)dt = \tau_i^*(\pi,s)|_{s=0} \triangleq \tau_i(\pi)$. Substituting $s=0$ we have

$$-\ \bar{\boldsymbol{L}}_{ii}(\pi)\tau_i(\pi) \ = \ 1 \ + \ \sum_{m=1}^{\infty}\sum_{\ell\in\Phi}\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi)_{i\ell} \ + \ \sum_{m=1}^{\infty}\left(\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi)\bar{\boldsymbol{B}}_{\boldsymbol{m}}\right)_{ii}\tau_i(\pi),$$

where we used that $\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi) = \hat{\boldsymbol{R}}^*_{\boldsymbol{m}}(\pi,s)|_{s=0}$. From this

$$-\left(\bar{\boldsymbol{L}}_{ii}(\pi) + \sum_{m=1}^{\infty}[\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi)\bar{\boldsymbol{B}}_{\boldsymbol{m}}]_{ii}\right)\tau_i(\pi) = 1 + \sum_{m=1}^{\infty}\sum_{\ell\in\Phi}\hat{\boldsymbol{R}}_{\boldsymbol{m}i\ell}(\pi).$$

Using $\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi) = \hat{\boldsymbol{R}}_{\boldsymbol{1}}(\pi)\boldsymbol{R}^{m-1}$ its matrix form is (54).

Based on a similar argument, for $c_i(\pi)$ we have

$$\begin{aligned}
c_i(\pi) = & \frac{\boldsymbol{C}_{\boldsymbol{1}ii}}{-\bar{\boldsymbol{L}}_{ii}(\pi)} + \sum_{m=1}^{\infty}\sum_{k\in\Phi}\frac{1}{-\bar{\boldsymbol{L}}_{ii}(\pi)}\hat{\boldsymbol{R}}_{\boldsymbol{m}ik}(\pi)\boldsymbol{C}_{\boldsymbol{m+1}kk} \\
& + \sum_{m=1}^{\infty}\sum_{k\in\Phi}\frac{1}{-\bar{\boldsymbol{L}}_{ii}(\pi)}\hat{\boldsymbol{R}}_{\boldsymbol{m}ik}(\pi)\bar{\boldsymbol{B}}_{\boldsymbol{m}ki}c_i(\pi),
\end{aligned}$$

from which

$$-\left(\bar{\boldsymbol{L}}_{ii}(\pi)+\sum_{m=1}^{\infty}[\hat{\boldsymbol{R}}_{\boldsymbol{m}}(\pi)\bar{\boldsymbol{B}}_{\boldsymbol{m}}]_{ii}\right)c_i(\pi)=\boldsymbol{C_1}_{ii}(\pi)+\sum_{m=1}^{\infty}\sum_{k\in\Phi}\hat{\boldsymbol{R}}_{\boldsymbol{m}ik}(\pi)\boldsymbol{C_{m+1}}_{kk},$$

whose matrix form is (55) □

7 Calculation of $c(\pi)$ with different reward structures

In this section, we provide methods to calculate $c(\pi)$ when the infinite reward matrix $\boldsymbol{C}(\pi)$ follows different regular structures. Specifically, we consider the following cases:

- geometric series: $\boldsymbol{C_n}=\kappa^n\boldsymbol{C_0}, \forall n>1$,
- matrix geometric series: $\boldsymbol{C_n}=\boldsymbol{C_0}^n, \forall n>1$,
- polynomial series: $\boldsymbol{C_n}=p(n)\boldsymbol{C_0}, \forall n>1$,

where $\boldsymbol{C_0}$ is arbitrary diagonal matrix, κ is a positive real number and $p(n)$ is a finite order arbitrary non-negative polynomial of n.

7.1 Calculating the $c(\pi)$ reward function for G/M/1 type processes

For G/M/1 type processes, (55) defines the relation of $c(\pi)$ with $\boldsymbol{C_1}(\pi)$ and $\boldsymbol{C_m}$ ($m\geq 2$). The only non-trivial part of this formula is the computation of the infinite sum $\sum_{m=1}^{\infty}\boldsymbol{R}^{m-1}\boldsymbol{C_{m+1}}$.

7.1.1 Reward function for $\boldsymbol{C_n}=\kappa^n\boldsymbol{C_0}$

If $\boldsymbol{C_n}=\kappa^n\boldsymbol{C_0}$, then the infinite sum converges when $\kappa\lambda_R<1$, where λ_R is the spectral radius of $\boldsymbol{R}$. In this case,

$$\sum_{m=1}^{\infty}\boldsymbol{R}^{m-1}\boldsymbol{C_{m+1}}=\sum_{m=1}^{\infty}\boldsymbol{R}^{m-1}\kappa^{m+1}\boldsymbol{C_0}=\kappa^2(\boldsymbol{I}-\kappa\boldsymbol{R})^{-1}\boldsymbol{C_0}.$$

7.1.2 Reward function for $\boldsymbol{C_n}=\boldsymbol{C_0}^n$

If $\boldsymbol{C_n}=\boldsymbol{C_0}^n$, then $\sum_{m=1}^{\infty}\boldsymbol{R}^{m-1}\boldsymbol{C_0}^{m-1}\boldsymbol{C_0}^2=\boldsymbol{X}\boldsymbol{C_0}^2$, where $\boldsymbol{X}=\sum_{m=1}^{\infty}\boldsymbol{R}^{m-1}\boldsymbol{C_0}^{m-1}$ is the solution of the Sylvester equation $\boldsymbol{X}=\boldsymbol{I}+\boldsymbol{R}\boldsymbol{X}\boldsymbol{C_0}$.

7.1.3 Reward function for $\boldsymbol{C_n} = p(n)\boldsymbol{C_0}$

If $\boldsymbol{C_n} = p(n)\boldsymbol{C_0}$, without loss of generality, we assume $p(n) = \sum_{i=0}^{k} a_i(n-2)^i$. In this case,

$$\begin{aligned}\sum_{m=1}^{\infty} \boldsymbol{R}^{m-1}\boldsymbol{C_{m+1}} &= \sum_{m=1}^{\infty} \boldsymbol{R}^{m-1} \sum_{i=0}^{k} a_i(m-1)^i \boldsymbol{C_0} \\ &= a_0(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{C_0} + \sum_{i=1}^{k} a_i \sum_{m=2}^{\infty} \frac{\boldsymbol{R}^{m-1}}{(m-1)^{-i}} \boldsymbol{C_0} \\ &= a_0(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{C_0} + \sum_{i=1}^{k} a_i Li_{-i}(\boldsymbol{R})\boldsymbol{C_0},\end{aligned}$$

where $Li_\ell(\boldsymbol{Y})$ is the polylogarithm function generalised for matrices, i.e.,

$$Li_\ell(\boldsymbol{R}) = \sum_{m=1}^{\infty} \frac{\boldsymbol{R}^m}{m^\ell}. \tag{61}$$

If $\ell \in \mathbb{Z}^+$ and the spectral radius of $\boldsymbol{R}$ is less than one (which holds if the generator of the MDP is positive recurrent) $Li_{-\ell}(\boldsymbol{R})$ is finite and can be computed as

$$Li_{-k}(\boldsymbol{R}) = (-1)^{k+1} \sum_{i=0}^{k-1} i!S(k+1, i+1)(\boldsymbol{R}-\boldsymbol{I})^{i+1},$$

(see e.g. [18]), where $S(k,i) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n$ denotes the Stirling number of second kind.

7.2 Calculating the $c(\pi)$ reward function for M/G/1 type processes

For M/G/1 type processes, (41) needs to be solved for $\boldsymbol{\hat{M}_n}$, which is hard in general. To simplify the discussion, in this section, we utilize the fact that the M/G/1 type processes is positive recurrent and consequently $\boldsymbol{\hat{G}}\underline{\mathbf{1}} = \underline{\mathbf{1}}$. Multiplying both sides of (41) with $\underline{\mathbf{1}}$ and using $\boldsymbol{\hat{G}}\underline{\mathbf{1}} = \underline{\mathbf{1}}$ we have

$$\begin{aligned}\boldsymbol{C_n}\underline{\mathbf{1}} &= \boldsymbol{L}\boldsymbol{\mu_n} + \sum_{m=1}^{\infty} \boldsymbol{F_m} \sum_{\ell=0}^{m} \boldsymbol{\hat{G}}^\ell \boldsymbol{\mu_{n+m-\ell}}, \\ &= \boldsymbol{L}\boldsymbol{\mu_n} + \sum_{m=1}^{\infty} \boldsymbol{F_m}\boldsymbol{\mu_{n+m}} + \sum_{\ell=1}^{\infty}\sum_{m=\ell}^{\infty} \boldsymbol{F_m}\boldsymbol{\hat{G}}^\ell \boldsymbol{\mu_{n+m-\ell}},\end{aligned} \tag{62}$$

where $\boldsymbol{\mu_n} = \boldsymbol{\hat{M}_n}\underline{\mathbf{1}}, \forall n > 0$, which is sufficient to compute $c(\pi)$ according to (38).

7.2.1 Reward function for $\boldsymbol{C_n} = \kappa^n \boldsymbol{C_0}$

If $\boldsymbol{C_n} = \kappa^n \boldsymbol{C_0}$, then from (50) we have the following relation for $n > 0$

$$\hat{\boldsymbol{M}}_{\boldsymbol{n}ij} = \sum_{m=0}^{\infty} \sum_{\ell \in \Phi} \hat{\boldsymbol{T}}_{\boldsymbol{m}ij\ell} \boldsymbol{C}_{\boldsymbol{n+m}\ell\ell} = \sum_{m=0}^{\infty} \sum_{\ell \in \Phi} \hat{\boldsymbol{T}}_{\boldsymbol{m}ij\ell} \kappa^n \boldsymbol{C}_{\boldsymbol{m}\ell\ell} = \kappa^n \hat{\boldsymbol{M}}_{\boldsymbol{0}ij},$$

and consequently, $\boldsymbol{\mu_n} = \kappa^n \boldsymbol{\mu_0}$. Substituting this into (62), $\boldsymbol{\mu_0}$ can be computed from

$$\boldsymbol{C_0}\underline{\boldsymbol{1}} = \left(\boldsymbol{L} + \sum_{m=1}^{\infty} \kappa^m \boldsymbol{F_m} + \sum_{\ell=1}^{\infty} \sum_{m=\ell}^{\infty} \kappa^{m-\ell} \boldsymbol{F_m} \hat{\boldsymbol{G}}^{\ell} \right) \boldsymbol{\mu_0}, \tag{63}$$

where the existence and the singularity of the matrix in bracket depends on $\boldsymbol{L}$, $\boldsymbol{F_m}$ and $\hat{\boldsymbol{G}}$.

7.2.2 Reward function for $\boldsymbol{C_n} = \boldsymbol{C_0}^n$

This case does not provide a simple relation for the $\boldsymbol{\mu_n}$ series, which makes the solution of (41) possible in general. A practically important case, when $\boldsymbol{F_m} = \boldsymbol{F}^m$, allows analytically compact description and is discussed below.

7.2.3 Reward function for $\boldsymbol{C_n} = p(n)\boldsymbol{C_0}$

For $\boldsymbol{C_n} = p(n)\boldsymbol{C_0} = \sum_{u=0}^{k} a_u n^u \boldsymbol{C_0}$ we utilize the linear structure of (62) and separate the solution into the following sub-problems

$$\boldsymbol{C_{n,u}}\underline{\boldsymbol{1}} = \boldsymbol{L}\boldsymbol{\mu_{n,u}} + \sum_{m=1}^{\infty} \boldsymbol{F_m}\boldsymbol{\mu_{n+m,u}} + \sum_{\ell=1}^{\infty} \sum_{m=\ell}^{\infty} \boldsymbol{F_m} \hat{\boldsymbol{G}}^{\ell} \boldsymbol{\mu_{n+m-\ell,u}}, \tag{64}$$

where $\boldsymbol{C_{n,u}} = n^u \boldsymbol{C_0}$, for $u = \{0, 1, \ldots, k\}$. From the solutions for the sub-problems, $\boldsymbol{\mu_{n,u}}$, the solution for $\boldsymbol{C_n} = \sum_{i=0}^{k} a_u \boldsymbol{C_{n,u}}$ is obtained as

$$\boldsymbol{\mu_n} = \sum_{u=0}^{k} a_u \boldsymbol{\mu_{n,u}}.$$

For $\hat{\boldsymbol{M}}_{\boldsymbol{n,0}}$ we have

$$\begin{aligned} \hat{\boldsymbol{M}}_{\boldsymbol{n,0}ij} &= \sum_{m=0}^{\infty} \sum_{\ell \in \Phi} \hat{\boldsymbol{T}}_{\boldsymbol{m}ij\ell} \boldsymbol{C}_{\boldsymbol{n+m,0}\ell\ell} \\ &= \sum_{m=0}^{\infty} \sum_{\ell \in \Phi} \hat{\boldsymbol{T}}_{\boldsymbol{m}ij\ell} (n+m)^0 \boldsymbol{C}_{\boldsymbol{0}\ell\ell} = \sum_{m=0}^{\infty} \sum_{\ell \in \Phi} \hat{\boldsymbol{T}}_{\boldsymbol{m}ij\ell} \boldsymbol{C}_{\boldsymbol{0}\ell\ell} \overset{def}{=} \hat{\boldsymbol{M}}, \end{aligned}$$

from which $\boldsymbol{\mu_{n,0}} = \boldsymbol{\hat{M}\underline{1}} \overset{def}{=} \boldsymbol{\mu}$ is independent of n and can be computed from

$$\boldsymbol{C_0\underline{1}} = \left(\boldsymbol{L} + \sum_{m=1}^{\infty} \boldsymbol{F_m} + \sum_{\ell=1}^{\infty}\sum_{m=\ell}^{\infty} \boldsymbol{F_m}\boldsymbol{\hat{G}}^{\ell}\right)\boldsymbol{\mu}. \tag{65}$$

For $u > 0$ we have

$$\begin{aligned}\boldsymbol{\hat{M}_{n+v,u}}_{ij} &= \sum_{m=0}^{\infty}\sum_{\ell\in\Phi} \boldsymbol{\hat{T}_{m}}_{ij\ell}\boldsymbol{C_{n+v+m,u}}_{\ell\ell} = \sum_{m=0}^{\infty}\sum_{\ell\in\Phi} \boldsymbol{\hat{T}_{m}}_{ij\ell}(n+v+m)^u\boldsymbol{C_0}_{\ell\ell} \\ &= \sum_{m=0}^{\infty}\sum_{\ell\in\Phi} \boldsymbol{\hat{T}_{m}}_{ij\ell}\sum_{r=0}^{u}\binom{u}{r}n^{u-r}(v+m)^r\boldsymbol{C_0}_{\ell\ell} \\ &= \sum_{r=0}^{u}\binom{u}{r}n^{u-r}\sum_{m=0}^{\infty}\sum_{\ell\in\Phi} \boldsymbol{\hat{T}_{m}}_{ij\ell}\boldsymbol{C_{v+m,r}}_{\ell\ell} = \sum_{r=0}^{u}\binom{u}{r}n^{u-r}\boldsymbol{\hat{M}_{v,r}}_{ij},\end{aligned}$$

that is $\boldsymbol{\mu_{n+v,u}} = \sum_{r=0}^{u}\binom{u}{r}n^{u-r}\boldsymbol{\mu_{v,r}}$.

For $v = 1$ this gives, $\boldsymbol{\mu_{n+1,u}} = \sum_{r=0}^{u}\binom{u}{r}n^{u-r}\boldsymbol{\mu_{1,r}}$. Substituting it into (64) for $n = 1$ and $u > 0$ gives an equation in which the unknowns are $\boldsymbol{\mu_{1,r}}$ for $r = 0, 1, \ldots, u$. E.g., for $n = 1, u = 1$ we have

$$\begin{aligned}\boldsymbol{C_0\underline{1}} - \left(\sum_{m=1}^{\infty} m\boldsymbol{F_m} + \sum_{\ell=1}^{\infty}\sum_{m=\ell}^{\infty}(m-\ell)\boldsymbol{F_m}\boldsymbol{\hat{G}}^{\ell}\right)\boldsymbol{\mu_{1,0}} = \\ \left(\boldsymbol{L} + \sum_{m=1}^{\infty}\boldsymbol{F_m} + \sum_{\ell=1}^{\infty}\sum_{m=\ell}^{\infty}\boldsymbol{F_m}\boldsymbol{\hat{G}}^{\ell}\right)\boldsymbol{\mu_{1,1}},\end{aligned}$$

from which $\boldsymbol{\mu_{1,1}}$ can be computed, since $\boldsymbol{\mu_{1,0}} = \boldsymbol{\mu}$ is known from (65). Recursively, applying the same procedure for $u = 1, 2, \ldots, k$ provides the required $\boldsymbol{\mu_{1,u}}$ matrices, from which all $\boldsymbol{\mu_{n,u}}$ matrices ($n \geq 1, u \geq 0$) can be calculated using $\boldsymbol{\mu_{n,u}} = \sum_{r=0}^{u}\binom{u}{r}(n-1)^{u-r}\boldsymbol{\mu_{1,r}}$.

7.2.4 Special case of matrix geometric $\boldsymbol{F_m}$ series

When $\boldsymbol{F_m} = \boldsymbol{F}^m$, the infinite summations of the previous subsections containing $\boldsymbol{F_m}$ simplifies significantly. E.g., in (63)

$$\begin{aligned}\sum_{\ell=1}^{\infty}\sum_{m=\ell}^{\infty}\kappa^{m-\ell}\boldsymbol{F_m}\boldsymbol{\hat{G}}^{\ell} = \sum_{\ell=1}^{\infty}\sum_{m=\ell}^{\infty}\kappa^{m-\ell}\boldsymbol{F}^{m-\ell}\boldsymbol{F}^{\ell}\boldsymbol{\hat{G}}^{\ell} = (\boldsymbol{I}-\kappa\boldsymbol{F})^{-1}\underbrace{\sum_{\ell=1}^{\infty}\boldsymbol{F}^{\ell}\boldsymbol{\hat{G}}^{\ell}}_{\boldsymbol{X}-\boldsymbol{I}} \\ = (\boldsymbol{I}-\kappa\boldsymbol{F})^{-1}(\boldsymbol{X}-\boldsymbol{I}),\end{aligned}$$

where $\boldsymbol{X}$ is the solution of the Sylvester equation $\boldsymbol{X} = \boldsymbol{I} + \boldsymbol{F}\boldsymbol{X}\hat{\boldsymbol{G}}$.

In addition to this analytical simplicity, $\boldsymbol{F_m} = \boldsymbol{F}^m$ makes it possible to compute the solution for the matrix geometric reward function.

Theorem 5. *If $\boldsymbol{C_n} = \boldsymbol{C_0}^n$ and $\boldsymbol{F_m} = \boldsymbol{F}^m$ then the solution of* (62) *is $\boldsymbol{\mu_n} = \boldsymbol{H}\boldsymbol{C_0}^n\underline{\boldsymbol{1}}$, where $\boldsymbol{H}$ is the solution of the Sylvester equation*

$$\boldsymbol{H} = (\boldsymbol{L} - \boldsymbol{I} + \boldsymbol{X})^{-1}(\boldsymbol{I} - \boldsymbol{F}\boldsymbol{C_0}) + (\boldsymbol{L} - \boldsymbol{I} + \boldsymbol{X})^{-1}\boldsymbol{F}(\boldsymbol{L} - \boldsymbol{I})\ \boldsymbol{H}\,\boldsymbol{C_0} \tag{66}$$

and $\boldsymbol{X}$ is the solution of the Sylvester equation $\boldsymbol{X} = \boldsymbol{I} + \boldsymbol{F}\boldsymbol{X}\hat{\boldsymbol{G}}$.

Proof. If $\boldsymbol{C_n} = \boldsymbol{C_0}^n$ and $\boldsymbol{F_m} = \boldsymbol{F}^m$ then (62) takes the form

$$\begin{aligned}
\boldsymbol{C_0}^n\underline{\boldsymbol{1}} &= \boldsymbol{L}\boldsymbol{\mu_n} + \sum_{m=1}^{\infty}\sum_{\ell=0}^{m}\boldsymbol{F}^m\hat{\boldsymbol{G}}^{\ell}\boldsymbol{\mu_{n+m-\ell}} \\
&= (\boldsymbol{L} - \boldsymbol{I})\boldsymbol{\mu_n} + \sum_{m=0}^{\infty}\sum_{\ell=0}^{m}\boldsymbol{F}^m\hat{\boldsymbol{G}}^{\ell}\boldsymbol{\mu_{n+m-\ell}} \\
&= (\boldsymbol{L} - \boldsymbol{I})\boldsymbol{\mu_n} + \sum_{k=0}^{\infty}\sum_{m=k}^{\infty}\boldsymbol{F}^m\hat{\boldsymbol{G}}^{m-k}\boldsymbol{\mu_{n+k}} \\
&= (\boldsymbol{L} - \boldsymbol{I})\boldsymbol{\mu_n} + \sum_{k=0}^{\infty}\boldsymbol{F}^k\boldsymbol{X}\boldsymbol{\mu_{n+k}},
\end{aligned}$$

which suggests a matrix geometric solution, $\boldsymbol{\mu_n} = \boldsymbol{W}^n\boldsymbol{\mu}$. Substituting this solution, for $n > 0$ we get

$$\boldsymbol{C_0}^n\underline{\boldsymbol{1}} = (\boldsymbol{L} - \boldsymbol{I})\boldsymbol{\mu_n} + \sum_{k=0}^{\infty}\boldsymbol{F}^k\boldsymbol{X}\boldsymbol{W}^{n+k}\boldsymbol{\mu} = \left(\boldsymbol{L} - \boldsymbol{I} + \sum_{k=0}^{\infty}\boldsymbol{F}^k\boldsymbol{X}\boldsymbol{W}^k\right)\boldsymbol{W}^n\boldsymbol{\mu},$$

Since $\boldsymbol{C_0}$ is a diagonal matrix the spectral decomposition of $\boldsymbol{W}$ should be $\boldsymbol{W} = \boldsymbol{H}\boldsymbol{C_0}\boldsymbol{H}^{-1}$, where the unknowns (matrix $\boldsymbol{H}$ and vector $\boldsymbol{\mu}$) are defined by $\boldsymbol{H}^{-1}\mu = \underline{\boldsymbol{1}}$ and

$$\boldsymbol{I} = \left(\boldsymbol{L} - \boldsymbol{I} + \sum_{k=0}^{\infty}\boldsymbol{F}^k\boldsymbol{X}\boldsymbol{H}\boldsymbol{C_0}^k\boldsymbol{H}^{-1}\right)\boldsymbol{H}, \tag{67}$$

since

$$\boldsymbol{C_0}^n\underline{\boldsymbol{1}} = \underbrace{\left(\boldsymbol{L} - \boldsymbol{I} + \sum_{k=0}^{\infty}\boldsymbol{F}^k\boldsymbol{X}\boldsymbol{H}\boldsymbol{C_0}^k\boldsymbol{H}^{-1}\right)\boldsymbol{H}}_{\boldsymbol{I}}\,\boldsymbol{C_0}^n\,\underbrace{\boldsymbol{H}^{-1}\boldsymbol{\mu}}_{\underline{\boldsymbol{1}}}.$$

Let $\boldsymbol{Z} = \sum_{k=0}^{\infty} \boldsymbol{F}^k \boldsymbol{X}\boldsymbol{H}\boldsymbol{C_0}^k$. On the one hand $\boldsymbol{Z}$ is the solution of the Sylvester equation $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{H} + \boldsymbol{F}\boldsymbol{Z}\boldsymbol{C_0}$. On the other hand, from (67), we have

$$\boldsymbol{I} - (\boldsymbol{L} - \boldsymbol{I})\,\boldsymbol{H} = \boldsymbol{Z},$$

which gives the following linear equation for $\boldsymbol{H}$

$$\boldsymbol{I} - (\boldsymbol{L} - \boldsymbol{I})\,\boldsymbol{H} = \boldsymbol{X}\boldsymbol{H} + \boldsymbol{F}\,(\boldsymbol{I} - (\boldsymbol{L} - \boldsymbol{I})\,\boldsymbol{H})\,\boldsymbol{C_0},$$

whose standard Sylvester form is (66), and from $\boldsymbol{\mu} = \boldsymbol{H}\underline{\mathbf{1}}$ and $\mu_n = \boldsymbol{W}^n \mu$ we have $\mu_n = \boldsymbol{H_0}^n \underline{\mathbf{1}}$ which was to be proven. □

8 Conclusions

We presented a methodology for computing a reduced representation of MDPs when there is a finite subset of states with decisions and a potentially infinite subset of decision independent states. This methodology requires the computation of some state dependent reward measures, which we preformed for two practically important cases when the infinite subset of decision independent states has M/G/1-type and G/M/1-type structure. The special case, when the decision independent part has a QBD structure, can be computed by either of the two general cases. Some required measures are already provided in the literature of M/G/1-type and G/M/1-type processes, but the reward related measures, e.g., the ones discussed in the previous section has not been considered before.

References

[1] Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

[2] Eitan Altman. *Applications of Markov decision processes in communication networks: A survey.* PhD thesis, INRIA, 2000.

[3] Nicole Bäuerle and Ulrich Rieder. *Markov decision processes with applications to finance.* Springer Science & Business Media, 2011.

[4] Dario A. Bini, Guy Latouche, and Beatrice Meini. *Numerical Methods for Structured Markov Chains (Numerical Mathematics and Scientific Computation).* Oxford University Press, Inc., New York, NY, USA, 2005.

[5] L. Bodrog, M. Gribaudo, G. Horváth, A. Mészáros, and M. Telek. Control of queues with MAP servers: experimental results. In *The Eighth International Conference on Matrix-Analytic Methods in Stochastic Models (MAM8)*, pages 9–11, 2014.

[6] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial Intelligence*, 121(1):49–107, 2000.

[7] Michael George Bulmer. *Principles of statistics*. Courier Corporation, 1979.

[8] Dmitry Efrosinin. *Controlled Queueing Systems with Heterogeneous Servers*. PhD thesis, University of Trier, 2004.

[9] Ronald A Howard. *Dynamic programming and Markov processes.* John Wiley, 1960.

[10] Krishna Jagannathan, Shie Mannor, Ishai Menache, and Eytan Modiano. A state action frequency approach to throughput maximization over uncertain wireless channels. *Internet Mathematics*, 9(2-3):136–160, 2013.

[11] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

[12] Yasar Levent Kocaga and Amy R. Ward. Admission control for a multi-server queue with abandonment. *Queueing Systems*, 65(3):275–323, 2010.

[13] G. Latouche and V. Ramaswami. *Introduction to Matrix-Analytic Methods in Stochastic Modeling*. Series on statistics and applied probability. ASA-SIAM, 1999.

[14] A. Mészáros and M. Telek. Markov decision process and linear programming based control of MAP/MAP/N queues. In *Computer Performance Engineering, EPEW*, volume 8721 of *LNCS*, pages 179–193, Sept. 2014.

[15] Marcel Neuts. *Structured Stochastic Matrices of M/G/1 Type and Their Applications*. Probability: Pure and Applied. Taylor & Francis, 1989.

[16] Andrew Y Ng and Michael Jordan. Pegasus: A policy search method for large mdps and pomdps. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 406–415. Morgan Kaufmann Publishers Inc., 2000.

[17] Joris Slegers, Isi Mitrani, and Nigel Thomas. Optimal dynamic server allocation in systems with on/off sources. In *Formal Methods and Stochastic Models for Performance Evaluation*, pages 186–199. Springer, 2007.

[18] David Wood. The computation of polylogarithms. Technical Report 15-92*, University of Kent, Computing Laboratory, University of Kent, Canterbury, UK, June 1992.