A multi-cluster time aggregation approach for Markov chains [☆]

E. F. Arruda[☆☆a], **M. D. Fragoso**[b], **F. O. Ourique**[c]

[a]*Industrial Engineering Program, Alberto Luiz Coimbra Institute - Graduate School and Research in Engineering, Federal University of Rio de Janeiro. Caixa Postal 68507, Rio de Janeiro RJ 21941-972, Brazil.*
[b]*Center for Systems and Control, National Laboratory for Scientific Computation. Av. Getúlio Vargas, 333. Petrópolis, RJ 25651-075, Brazil.*
[c]*Department of Computing, Federal University of Santa Catarina, Araranguá SC, Brazil.*

Abstract

This work focuses on the computation of the steady state distribution of a Markov chain, making use of an embedding algorithm. In this regard, a well-known approach dubbed *time aggregation* has been proposed in [7]. Roughly, the idea hinges on the partition of the state space into two subsets. The linchpin in this partitioning process is a small subset of states, selected to be the state space of the aggregated process, which will account for the state space of the embedded semi-Markov process. Although this approach has provided an interesting body of theoretical results and advanced in the study of the so-called curse of dimensionality, one is still left with a high-dimensional problem to be solved. In this paper we investigate the possibility to remedy this problem by proposing a time aggregation approach with multiple subsets. This is achieved by devising a decomposition algorithm which makes use of a partition scheme to evaluate the steady state probabilities of the chain. Besides the convergence proof of the algorithm, we prove also a result for the cardinality of the partition, vis-à-vis the computational effort of the algorithm, for the case in which the state space is partitioned in a collection of subsets of the same cardinality.

Keywords: Markov Processes, Embedding, Time Aggregation, Policy Evaluation.

1. INTRODUCTION

Large scale Markov chains can benefit from embedding algorithms [e.g., 18] by finding an equivalent process in a reduced dimension. This approach, also known in the literature as time aggregation [e.g., 7], gives rise to an embedded Markov chain with modified costs and time scale, in such a way that the performance of the process in reduced dimension equals that of the original process under study. Arruda and Fragoso [2] argue that this can be seen as a transformation of a Markov chain into an embedded semi-Markov process with costs and transition times determined by the trajectories leaving and returning to the embedded domain.

Another popular approach for large scale Markov processes is state aggregation [e.g., 1, 10]. Similarly to time aggregation, state aggregation seeks a problem in a reduced dimension. However, as stated by Cao et al. [7], the latter method fails to maintain the Markov property and works instead with an approximated version of the original problem. Dimensionality is also an issue when control is involved, in the context of Markov decision processes. In such a context, approximate dynamic programming [21] and reinforcement learning [24] tools can be effectively used to find approximate sub-optimal solutions.

[☆] A preliminary version of this paper was presented at the 2017 IEEE Conference on Decision and Control (CDC) - [3]

[☆☆] Corresponding author. Tel.: +55 21 3938 8255; fax +55 21 2270 9702.

Email addresses: efarruda@po.coppe.ufrj.br (E. F. Arruda,[☆☆]), frag@lncc.br (M. D. Fragoso), fabricio.ourique@ufsc.br (F. O. Ourique)

Also concerned with dimensionality reduction are multi-grid aggregation/disaggregation methods [e.g. 12, 19, 5]. These methods typically work on two levels, *a coarse level* or aggregated domain with reduced cardinality and a *fine level* in the original domain [17], in order to iteratively evaluate the stationary probability of the Markov chain. Aggregation/disaggregation methods consist in an aggregated step where they evaluate the steady state probabilities in the coarse grid. A disaggregation step follows in which the results in the coarse grid are propagated to the larger domain, to recover the steady state probabilities in the whole state space. This approach allows parallelisation [8, 15] and may be efficiently applied to large scale problems. However, it is not guaranteed to converge faster than the direct algorithm for problems with moderate dimension [13].

As proposed by Cao et al. [7], time aggregation partitions the state space into two components: a small subset of states which is selected to be the state space of the aggregated process, and a much larger subset with all the remaining states. The embedded semi-Markov process which is solved in reduced dimension, however, requires the evaluation of trajectories leaving and returning to the embedded domain. Hence, although the embedded domain is typically reduced, one is still left with a high-dimensional Markov chain problem to solve. To circumvent that, this paper proposes a generalisation of the partition scheme in time aggregation in the sense that, instead of only two subsets, we allow an arbitrary number of subsets, an approach that resembles aggregation/disaggregation routines. The inchoate idea here lies on evaluating trajectories leaving the subsets in the partition, which tend to be much shorter than the sojourns outside of each subset, because the cardinality of the subsets tend to be much smaller than that of the whole state space. Hence, one can expect that the system spends a fairly larger portion of the time outside of a given subset. By doing so, we keep the computational effort as a function of the cardinality of the subsets in the partition scheme.

The main focus of the paper is on the design of the partition scheme. The transitions between subsets are modelled as a semi-Markov master model whose state space is comprised of the subsets in the partition scheme; hence, the cardinality of the state space is the number of subsets in the scheme. An auxiliary problem is solved within each subset to determine the average length and the number of visits to each state in an outbound trajectory, which are incorporated into the master model to produce transition probabilities and times between successive transitions. The steady state probabilities of the Markov chain are then found by an iterative algorithm which produces solutions to the semi-Markov process and then disaggregates the solution to find the probabilities in the original domain. The proposed approach also resembles the rationale in the context of large deviations [25], in that the semi-Markov process can be seen as connecting subsets in different time scales, whereas the transitions among the subsets can be seen as perturbations that occur with reduced probability in a model that is driven by the dynamics within the subsets of the partition.

Although it may seem at first that by increasing the number of subsets we improve the classical time aggregation approach, notice that by increasing too much the number of subsets one increases, as a by-product, the state space of the semi-Markov master model. In fact we have two equivalent extremes: a single subset and a partition scheme where each state is a subset of the semi-Markov master model. To select the proper scheme between these extremes, a key question in the partition scheme proposed here is regarding the suitable number of subsets. In this paper, the problem of properly selecting the subsets in the partition, with a view to optimise the overall computational effort, is tackled and we find the optimal number of subsets for a partition scheme with subsets of the same cardinality. Numerical results are presented to illustrate this choice.

While this paper deals with the design of a partition scheme, the focus is on the properties of the partition scheme and on the advantages of generalising time aggregation for multiple subsets. Since the main purpose here is to improve upon standard time aggregation approach, we performed a direct comparison with the latter in the numerical experiments, as it seemed natural. For a thorough evaluation of other numerical procedures to find steady state probabilities of Markov chains, we refer to [23].

It is perhaps worth mentioning that a very preliminary version of this paper appeared in [3]. The present manuscript incorporates many developments and new results, which add up to a completely reformulated text. For instance, the results regarding the optimal choice of the partition scheme in Section 5 are completely new; the literature review was completely reformulated; the numerical experiments were expanded and the proofs are modified and expanded.

This paper is organized as follows. Section 2 introduces the partition scheme and the section that follows makes use of it to derive the stationary probabilities of the original process. Then, an algorithm is brought forth in Section 4 which iteratively finds the stationary distribution, making use of the partition scheme. Next, Section 5 addresses the problem of finding an optimal number of subsets in the partition, and Section 6 features numerical experiments that highlight the results. The paper is concluded in Section 7.

2. Notations and Preliminaries

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space carrying a homogenous discrete-time Markov chain X_k, $k \geq 0$. To motivate the partition scheme, suppose that the state space S is finite, but *very large*. We assume that the one-step transitions of the Markov chain are governed by a transition probability matrix $P := [p_{ij}]$, with $p_{ij} \in [0,1]$, $i,j \in S$ denoting the probability that the process will visit state j in the next period, given that it arrived to state i at the current period. In addition, we assume that the chain is ergodic, i.e. it is irreducible and aperiodic. It then follows from classical Markov chain results that the chain possesses a unique stationary distribution, which is independent of the initial states [6]. Furthermore, we denote by I the identity matrix (with suitable dimension); by $\mathcal{E}\{.\}$ the mathematical expected value and by $\mathcal{E}_{\mu_l}\{.\}$ the expected value with relation to the measure μ_l. Finally, $|S|$ stands for the cardinality of a set S.

Since the proposed approach is rooted in partitioning the state space, we consider n disjoint subsets $F_1, \ldots, F_n$ such that $\cup_{l=1}^{n} F_l = S$ (see Figure 1 for an example of a partition scheme with two subsets, with the arrows representing positive transition probabilities in P). In addition, for the decision maker to map the transitions between subsets one can think of a subset F_l, $i = 1, \ldots, n$ as a transient class. Henceforth, we use the terms class and subsets interchangeably. Since the chain is ergodic, it follows that subset F_l will be left in finite time. Naturally, after evolving within subset F_l for a finite time, the process will finally leave and reach a distinct subset $F_m, m \neq l$. We first stop our analysis precisely at the time the process reaches a different subset, taking into account the probability of reaching each of the other subsets.

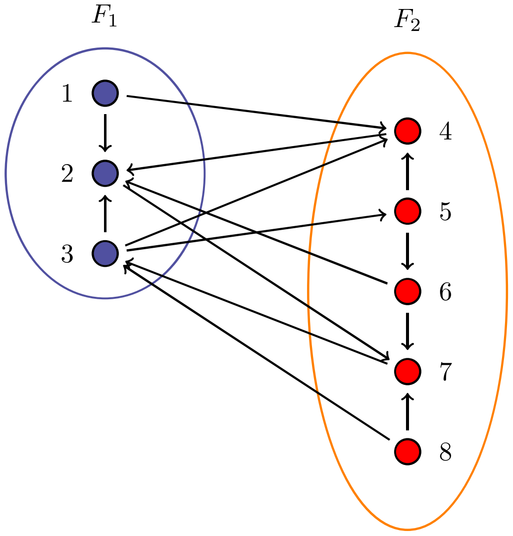

Figure 1: An example of a two-subset partition scheme

Roughly, a key technical idea behind the proposed approach here is to decompose the problem of finding the stationary distribution of process X_k, $k \geq 0$ in two steps. In the first, a partition scheme is devised and we define a master semi-Markov process that models the transitions between subsets in the partition. That allows us to evaluate the long-term probabilities of each subset. Then, to evaluate individually the long-term probability of each state in a given subset, we make use of absorption analysis. This step is applied individually for each subset in the partition, which is treated as a transient class, and the process is detailed in Section 2.2.

In what follows we fleetingly refer to some technical results which are germane for the development of our approach, including results on absorption analysis with a single transient class.

2.1. Absorption analysis

Absorption analysis is centred on the division of the state space into recurrent and transient classes; the objective is to find the time, and possibly the costs, until the chain is *absorbed* by a recurrent class [see, e.g., 6]. Standard absorption analysis is often performed by splitting the transition probability matrix of the Markov chain into blocks by writing:

$$M = \left[\begin{array}{c|c} \mathcal{O} & R \\ \hline 0 & I \end{array}\right], \quad R = [R_1\, R_2\, \dots R_n]. \tag{1}$$

In the decomposition above one can make R_l, $l = 1, \dots, n$ represent the aggregate probabilities that the process will be immediately absorbed by recurrent class l. In addition, $\mathcal{O}$ is the sub-matrix that represents the transitions between transient states.

To understand the behaviour of the system up to absorption, we can make use of the *fundamental matrix*:

$$M_F = (I - \mathcal{O})^{-1},$$

which gives the number of visits to each transient state, j, before absorption, for each possible initial state i from the transient subset, see for example [6, Section 4.6]. For any finite absorbing Markov chain, the inverse in the right hand side of the preceding expression exists [e.g., 16, Theorem 3.21, Page 46]. To find the probability of absorption by each recurrent class, from each initial transient state, one simply evaluates:

$$M_A = M_F R.$$

2.2. State space decomposition

In order to carry out our analysis, we consider $\tau_0 = 0$ and $\tau_k = \min\{t > \tau_{k-1} : X_t \in F_m, \text{and}\, X_{t-1} \notin F_m \,\text{for some}\, m \in \{1, \dots n\} \,\text{and}\, t \in \mathbb{N}\}$. One can see that τ_k, $k \geq 0$ is the sequence of times on which the process X_k, $k \geq 0$ registers changes of subsets within the partition. Between any successive times in this sequence, the process evolves in a single subset of the partition.

To account for the hitting times of each subset F_l, $l = 1, \dots, n$ we denote by $\bar{X}_k \triangleq X_{\tau_k}$, $k \geq 0$ the process at the moments the system changes subsets, also known in the literature as the jump process [20]. Note that process $\bar{X}_k$, $k \geq 0$ also evolves in S. However, as stated in Remark 2, some states in S may be transient with respect to $\bar{X}_k$, $k \geq 0$.

The transitions among subsets are monitored by means of a semi-Markov process Y_k, $k \geq 0$, defined as

$$Y_k = \sum_{l=1}^{n} l \cdot \mathbb{1}_{\{\bar{X}_k \in F_l\}}, \tag{2}$$

where $\mathbb{1}_{\{C\}}$ is the indicator function of statement C, which is equal to one whenever C holds true and is nil otherwise. One can easily see from (2) that process Y_k, $k \geq 0$ gives, at each time, the label of the subset the system is evolving in. Hence, the state space of this process is given by $S_Y = \{1, 2, \dots, n\}$, and the transition matrix for its embedded Markov chain, denoted as $Q := [q_{ij}]$, with $q_{ij} \in [0, 1]$, $i, j \in S_Y$, conveys the dynamics of the transitions among subsets of the partition [e.g., 14].

Now, to quantify the transitions between subsets we make use of the block structure in (1) in order to define, for each subset F_l, an auxiliary absorption problem with:

$$\begin{gathered} \mathcal{O} = P_l \;\text{ and} \\ R_l = \left[\begin{array}{c|c|c|c|c|c|c} R_{l1} & R_{l2} & \dots & R_{l\,(l-1)} & R_{l\,(l+1)} & \dots & R_{ln} \end{array}\right], \end{gathered} \tag{3}$$

where

$$P_l = [p_{ij}], \quad i, j \in F_l, \tag{4}$$

$$R_{lm} = [\sum_{j \in F_m} p_{ij}], \quad i \in F_l,\, m \neq l, \tag{5}$$

where P_l is a square matrix with dimension equal to the cardinality of the subset F_l and R_{lm} is a column vector with number of elements equal to the cardinality of the subset F_l.

In what follows we shall need the additional notations:

- Matrix $E_l = [e_l(i,j)]$ concerning the *outbound trajectories from subset* F_l. The element $e_l(i,j)$ gives the expected number of visits that an outbound trajectory starting at state $i \in F_l$ will pay to state $j \in F_l$ before leaving subset F_l. From Section 2.1, it follows that:

$$E_l = (I - P_l)^{-1}; \tag{6}$$

- Matrix $A_l = [a_l(i,m)]$ concerning the *jumps between subsets*. Element $a_l(i,m)$ is the probability that a trajectory starting at state $i \in F_l$ reaches subset F_m, $m \in S_Y$, $m \neq l$, upon leaving F_l. Bearing in mind the definition of M_A in Section 2.1, it follows that:

$$A_l = E_l \cdot R_l. \tag{7}$$

Remark 1. *Notice that, since matrix Q monitors transitions between subsets, it evolves in the same time scale as process $\bar{X}_k$, $k \geq 0$, which we will call the* jump time scale. *Such a time scale does not account for the actual time spent between transitions. Therefore, to completely specify the semi-Markov process Y_k, $k \geq 0$ we need to keep an account of the inter-jump times ($\tau_{k+1} - \tau_k$, $k \geq 0$) in order to evaluate the process in the original time scale (the time scale of X_k, $k \geq 0$). Let π_Y be the steady state probabilities of process Y_k, $k \geq 0$ and $\tau(l)$ denote the total duration of a visit to subset F_l, i.e. the inter-jump time when τ_k and τ_{k+1} are the entrance and departure times of subset F_l, respectively. Standard semi-Markov results yield that [e.g 14]:*

$$\pi_Y(l) = \tilde{\pi}_Y(l) \frac{\mathcal{E}(\tau(l))}{\bar{\tau}},\ l \in S_Y, \tag{8}$$

where $\tilde{\pi}$ is the steady state probability vector of the embedded process induced by Q, $\mathcal{E}(\tau(l)) = \mu_l\, E_l\, e$ (where μ_l is a row vector denoting a probability distribution over F_l and e is a column vector of ones of appropriate dimension) is the expected duration of a visit to subset F_l with respect to the measure generated by process X_k, $k \geq 0$ (in the time scale of X_k, $k \geq 0$) and $\bar{\tau} = \sum_{l=1}^{n} \tilde{\pi}(l)\mathcal{E}(\tau(l))$ is the expected time between subset jumps.

Theorem 1. *Let $\mu_l = [\mu_l(j)]$, $j \in F_l$, be a row vector of dimension $|F_l|$ representing the initial distribution of X_k, $k \geq 0$ upon reaching set F_l, with $\mu_l(j) \in [0,1)\, \forall j \in F_l$. Then*

$$q_{lm} = P(Y_{k+1} = m | Y_k = l) = \sum_{i \in F_l} \mu_l(i) a_l(i,m). \tag{9}$$

Proof. We first make sure that the block structure in (1), with the elements given in (3) is indeed a absorbing Markov chain, i.e. one that eventually leaves the transient subset F_l. To see that, we recall that by hypothesis the process X_k, $k \geq 0$ is irreducible. Hence, the states in F_l cannot form by themselves a *closed* communication class, otherwise the chain could be reduced to such a class [e.g., 6]. As a consequence, there must be some states not in F_l that can be directly accessed from F_l, which implies that the block structure chosen for F_l is an absorbing Markov chain.

It is clear that, if the initial state in the trajectory is drawn according to μ_l, then Eq. (9) gives the probability of reaching F_m from F_l, and that concludes the proof. □

We shall also need additional notations below:

- A matrix for *transitions between states outside of F_l*

$$\bar{P}_l = [p_{ij}], \quad i, j \notin F_l; \tag{10}$$

- A matrix for *transitions to subset F_l from any other subset* in the partition

$$\bar{R}_l = [p_{ij}], \quad i \notin F_l, \, j \in F_l; \tag{11}$$

- A matrix for *inbound trajectories to F_l*, starting from any other subset F_m, $m \neq l$, in the partition

$$\bar{E}_l = (I - \bar{P}_l)^{-1}. \tag{12}$$

  Similarly to matrix E_l in (6), $\bar{E}_l$ is concerned with the average length of the trajectories started elsewhere until they reach subset F_l. Element $\bar{e}_l(i, j)$ gives the expected number of visits that an outbound trajectory starting at state $i \notin F_l$ will pay to state $j \notin F_l$ before reaching subset F_l;

- A matrix to evaluate the *destination of outbound trajectories started in F_l*

$$B_l = E_l \cdot R_p^l, \quad \textit{with} \quad R_p^l = [p_{ij}], \, i \in F_l, \, j \notin F_l. \tag{13}$$

  Element $b_l(i, j)$ gives the probability that an outbound trajectory started at $i \in F_l$ reaches state $j \notin F_l$ immediately after leaving F_l;

- A transition matrix to evaluate *the destination of inbound trajectories to F_l* starting at any other subset in the partition

$$\bar{A}_l = \bar{E}_l \cdot \bar{R}_l. \tag{14}$$

  Element $\bar{a}_l(i, j)$, $i \notin F_l$, $j \in F_l$, yields the probability that an inbound trajectory started at $i \notin F_l$ reaches F_l through state j.

It is important at this point to differentiate matrices A_l, defined in (7) and B_l, defined in (13). Matrix A_l is concerned with the subset that will be reached following a trajectory started in $i \in F_l$, with $a_l(i, m)$ being the probability of reaching any state in F_m from state $i \in F_l$. Matrix B_l, on the other hand, is concerned with the specific state outside of F_l that will be immediately reached following a trajectory started in $i \in F_l$, with $b_l(i, j)$ being the probability of reaching a given state $j \notin F_l$ from $i \in F_l$.

Lemma 1. *Let $\bar{X}_k$, $k \geq 0$ and Y_k, $k \geq 0$ be defined according to the partition scheme previously introduced. In addition, let μ_l be a normalized subset of the steady state distribution of process $\bar{X}_k$, $k \geq 0$ over F_l. Then,*

$$\lim_{k \to \infty} P(\bar{X}_k = i | Y_k = l) = \mu_l(i), \, \forall i \in F_l, \tag{15}$$

where $\mu_l = [\mu_l(\cdot)]$ is a row vector of dimension $|F_l|$ such that

$$\mu_l B_l \bar{A}_l = \mu_l. \tag{16}$$

Proof. Let us first have a look at matrix $B_l \bar{A}_l$. From equations (13)-(14), one can see that this product refers to a regenerative cycle starting and finishing in subset F_l, where the first matrix maps the cycle until it leaves F_l, whereas the second maps the remaining part of it until the process returns to F_l. In other words, element $b_l \bar{a}_l(i, j)$, $i, j \in F_l$, yields the probability that a trajectory started upon reaching F_l through state i leaves this subset and then returns to this same subset through state j. Observe that this embedded process maps the transition between *first visited* states in F_l in two subsequent stays at this subset. Therefore, $B_l \bar{A}_l$ is the transition matrix of process $\bar{X}_k$, $k \geq 0$ embedded in subset F_l. With some abuse of notation, we call this embedded process $\{\bar{X}_k : Y_k = l\}$.

Now, since process X_k, $k \geq 0$ is ergodic by hypothesis, $\bar{X}_k$, $k \geq 0$ must also have a single recurrent class, and all subsets in the partition must be visited infinitely often. Consequently, the embedded process $\{\bar{X}_k : Y_k = l\}$ must have at least one positive recurrent state. This implies that there is an invariant probability vector $\mu_l = [\mu_l(j)]$, $j \in F_l$ of dimension $|F_l|$ which solves (16). As a result, the solution to (15) is the invariant probability that solves (16).

□

Remark 2. *It is worth pointing out that the state space of the chain $\{\bar{X}_k : Y_k = l\}$ induced by $B_l \cdot \bar{A}_l$ is F_l, which may contain states that are not directly accessible from other subsets F_m, $m \neq l$. These states will be transient with respect to process $\bar{X}_k$, $k \geq 0$, for being not accessible from other subsets, they cannot be the first visited states in F_l. They are also transient with respect to process $\{\bar{X}_k : Y_k = l\}$, for they will be visited at most once. That happens because, if for some $j \in F_l$*

$$p_{ij} = 0, \forall i \notin F_l, \tag{17}$$

it must follow that $\bar{a}_l(k, j) = 0$, $\forall k \notin F_l$. That, in turn, implies that $b_l\bar{a}_l(i, j)$ must be nil for all $i \in F_l$, which signifies that, as far as process $\{\bar{X}_k : Y_k = l\}$ is concerned, j is not accessible from any other state in F_l. Therefore, it must be a transient state and $\mu_l(j)$ must be zero.

3. Stationary distributions

In this section we establish the relationship between the steady state probabilities of the original process X_k, $k \geq 0$ and that of the embedded process $\bar{X}_k$, $k \geq 0$. To accomplish that, we make use of a suitably defined regenerative cycle and make use of classical results to derive the steady state probability as a function of the number of visits to each state during the regenerative cycle.

To define the regenerative cycle, assume that the process starts at subset F_l, with distribution μ_l according to (16). Then, from Lemma 1, it follows that $P(X_0 = i) = P(X_{\tau_k} = i) = \mu_l(i)$, $i \in F_l$, where $k = \min\{t > 0 : Y_t = l \,\text{and}\, Y_{t-1} \neq l\}$. Consequently, the trajectories leaving and returning to subset F_l form regenerative cycles. Theorem 3 at the end of this section derives the steady state distribution of process $\bar{X}_k$, $k \geq 0$, for all states $i \in F_l$ by counting the number of visits to these states during such a regenerative cycle. It builds upon an auxiliary result in Lemma 2 that follows.

Lemma 2. *Let μ_l solve (16) and let τ_k be a stopping time of process X_k, $k \geq 0$ such that $k = \min\{t > 0 : Y_t = l\, and Y_{t-1} \neq l\}$, and assume $P(X_0 = j) = \mu_l(j)$, $\forall j \in F_l$. Then*

$$\mathcal{E}_{\mu_l}\left\{\sum_{t=0}^{\tau_k - 1} \mathbb{1}_{\{X_t = i\}}\right\} = [\mu_l B_l \bar{E}_l](i),\ \forall i \notin F_l. \tag{18}$$

Proof. The trajectories leaving and returning to set F_l can be divided in two parts, the first comprising the portion of them within this subset and the second accounting for the sojourn outside of F_l prior to the return to this subset. Let $D \subset S$ be the subset of states that can be directly accessed from F_l and that lie outside of F_l and define τ_D as the time to get to subset D for the first time. Then, we can write:

$$\mathcal{E}_{\mu_l}\left\{\sum_{k=0}^{\tau_k - 1} \mathbb{1}_{\{X_k = i\}}\right\} = \mathcal{E}_{\mu_l}\left\{\sum_{k=0}^{\tau_D - 1} \mathbb{1}_{\{X_k = i\}}\right\} + \sum_{j \notin F_l} [\mu_l B_l](j) \mathcal{E}\left\{\sum_{t=\tau_D}^{\tau_k - 1} \mathbb{1}_{\{X_t = i\}} | X_{\tau_D = j}\right\},$$

for $i \in S$. Assuming that $i \notin F_l$ one can easily see that the first part of the summation above must be zero, given no state outside of F_l can be visited before the process leaves this subset. Consequently, we obtain:

$$\mathcal{E}_{\mu_l}\left\{\sum_{k=0}^{\tau_k - 1} \mathbb{1}_{\{X_k = i\}}\right\} = \sum_{j \notin F_l} [\mu_l B_l](j) \mathcal{E}\left\{\sum_{t=\tau_D}^{\tau_k - 1} \mathbb{1}_{\{X_t = i\}} | X_{\tau_D} = j\right\},$$

for all $i \notin F_l$. To conclude the proof it suffices to note that $\bar{E}_l(j, i)$ equals the last term in the preceding equation. □

The next results shows how one can obtain the solution of (15) from the stationary distribution of the original process X_k, $k \geq 0$.

Theorem 2. *Suppose that μ_l is the solution of* (16) *and let τ_k be a stopping time of process X_k, $k \geq 0$ such that $k = \min\{t > 0 : Y_t = l \, and Y_{t-1} \neq l\}$, then for all $j \in F_l$ we have:*

$$\mu_l(j) = \frac{\sum\limits_{i \notin F_l} \pi(i) p_{ij}}{\sum\limits_{i \notin F_l} \left(\pi(i) \sum\limits_{\bar{j} \in F_l} p_{i\bar{j}} \right)}, \; j \in F_l, \tag{19}$$

where $\pi : S \to (0,1)$ is the stationary distribution of process X_k, $k \geq 0$.

Proof. Bearing in mind equation (14), the equation (16), which holds by hypothesis, can also be expressed as:

$$\mu_l = \mu_l B_l \bar{E}_l \bar{R}_l.$$

Hence, the $j-th$ component of vector μ_l can be written as:

$$\mu_l(j) = \sum_{i \notin F_l} [\, \mu_l B_l \bar{E}_l \,](i) p_{ij}.$$

Now, from Lemma 2, we have that $[\, \mu_l B_l \bar{E}_l \,](i) = \mathcal{E}_{\mu_l} \left(\sum\limits_{t=0}^{\tau_k - 1} \mathbb{1}_{\{X_t = i\}} \right)$, for $i \notin F_l$. Hence, it follows that

$$\mu_l(j) = \sum_{i \notin F_l} \mathcal{E}_{\mu_l} \left(\sum_{t=0}^{\tau_k - 1} \mathbb{1}_{\{X_t = i\}} \right) p_{ij}$$

Recall that, by hypothesis, Eq. (16) holds and μ_l is the stationary distribution of process $\{\bar{X}_k : Y_k = l\}$. Then, [9, Eq. (19)] implies:

$$\mathcal{E}_{\mu_l} \left(\sum_{t=0}^{\tau_k - 1} \mathbb{1}_{\{X_t = i\}} \right) = \mathcal{E}_{\mu_l}[\tau_k] \pi(i), \; i \in S.$$

To see that, it suffices to define $k_i^X = 1$, $k_j^X = 0$, $\forall j \neq i$, and $Y = F_l$ in the notation of Forestier and Varaiya [9]. By doing so, the expression above is a direct application of Eq. (19) in that reference.

Hence, we must have

$$\mu_l(j) = \mathcal{E}_{\mu_l}[\tau_k] \sum_{i \notin F_l} \pi(i) p_{ij}.$$

By noting that $\sum\limits_{j \in F_l} \mu_l(j) = 1$ we are spared the need to directly evaluate $\mathcal{E}_{\mu_l}[\tau_k]$ in the preceding expression, and that leads to Eq. (19), which concludes the proof. □

We just found the distribution of process $\bar{X}_k$, $k \geq 0$ from that of the original process X_k, $k \geq 0$. The next result goes in the opposite direction, to derive the distribution of the former from that of the latter, making use of the stationary distribution of the master process Y_k, $k \geq 0$.

Theorem 3. *Let $\pi : S \to (0,1)$ be the stationary distribution of process X_k, $k \geq 0$, and $\pi_Y : S_Y \to (0,1)$ be the stationary distribution of process Y_k, $k \geq 0$. The following expression holds,*

$$\pi(i) = \frac{[\mu_l E_l](i)}{\mu_l E_l e} \cdot \pi_Y(l), \; i \in F_l, \tag{20}$$

where e is a vector of ones with appropriate dimension and μ_l is the solution of (15) *in F_l.*

Proof. Let us set up a regenerative cycle of length τ_k, such that $k = \min\{t > 0 : Y_t = l \,\text{and}\, Y_{t-1} \neq l\}$, with $X_0 \in F_l$ and distributed according to μ_l, which solves (16). The discussion in Section 2.1 yields that the numerator in (20) is the expected number of visits to state i before leaving F_l when μ_l is the initial distribution in F_l. Accordingly, the denominator gives the expected sojourn time in F_l under initial distribution μ_l.

Since X_k, $k \geq 0$ does not return to F_l before period τ_k, it holds that:

$$P(X_{\tau_k} = i | Y_k = l) = \frac{[\mu_l E_l](i)}{\mu_l E_l e},\ i \in F_l,$$

Employing the total probability theorem and taking into account the fact that $\{(X_{\tau_k} \in F_l) \cap (Y_k \notin l)\} = \emptyset$, we find

$$P(X_{\tau_k} = i) = \frac{[\mu_l E_l](i)}{\mu_l E_l e} P(Y_k = l),\ i \in F_l.$$

Eq. (20) follows by noting that τ_k is a regenerative cycle and $\lim\limits_{k \to \infty} P(Y_k = l) = \pi_Y(l)$. □

4. Iterative evaluation of the stationary distribution

This section builds upon the technical results in the last couple of sections to propose an iterative algorithm to find the stationary distribution of the original Markov chain by means of a decomposition algorithm. Such a proposal will exploit the correspondence between the stationary distributions of the original process and that of the master process Y_k, $k \geq 0$, established in the last section.

By hypothesis, process X_k, $k \geq 0$ is ergodic, which means it possesses a unique stationary distribution. A well known algorithm to iteratively find this distribution is:

$$\pi_{k+1} = \pi_k P, \tag{21}$$

where $\pi_0 : S \to (0, 1)$ is an arbitrary probability distribution disposed in a row vector [e.g., 4, 11]. The calculations are a function of the cardinality of the state space S, which renders the procedure inefficient as the cardinality increases. The subsequent algorithm addresses this issue to find an equivalent set of equations in reduced dimension.

Algorithm 1 (Decomposed stationary distribution).

1. *Select a partition $\{F_1, \ldots, F_n\}$ for S. For each $l = 1, \ldots, n$, define arbitrary probabilities $\mu_l^0(j)\, \forall j \in F_l$, adding up to one*[1]*. Set $k = 0$ and an arbitrary tolerance ϵ.*
2. *For each $l \in \{1, 2, \ldots, n\}$ make $h_1^k(l) = \mu_l^k E_l e$, where E_l is evaluated using Eq.* (6) *and e is a column vector of ones with appropriate dimension.*
3. *Solve $\tilde{\pi}_Y Q = \tilde{\pi}_Y$. Make*
$$\bar{h} = \sum_{l=1}^{n} \tilde{\pi}_Y(l) h_1^k(l); \quad \pi_Y(l) = \frac{\tilde{\pi}_Y(l) \cdot h_1^k(l)}{\bar{h}}$$
4. *For each $i \in S$, evaluate $\pi^k(i)$ according to* (20)*.*
5. *For each $l \in \{1, 2, \ldots n\}$ and $j \in F_l$, apply* (19) *to assess $\mu_l^{k+1}(j)$.*
6. *If*
$$\|\mu_l^{k+1} - \mu_l^k\| < \epsilon, \ \ \forall l \in \{1, 2, \ldots, n\},$$
stop. Otherwise, return to Step 2.
7. *For all $i \in S$, $\pi(i) \leftarrow \pi^k(i)$.*

[1] Preferably set $\mu_l^0(j) = 0$, for all $j \in F_l$ for which (17) holds, for these states will be transient (see Remark 2).

As defined in Section 2.2, process Y_k, $k \geq 0$ is a semi-Markov process with an embedded Markov chain with transition matrix Q. Since matrix Q features no self transitions, the departure times of each partition are the transition times of the semi-Markov process. Step 3 of Algorithm 1 makes use of these times to find the steady state probabilities of process Y_k, $k \geq 0$ [e.g., 22, Chapter 11], see also Remark 1. These probabilities yield the fraction of time the process stays at each subset F_l, and are combined with the number of steps the process spends at each state $i \in S$ to obtain a new estimate of the steady state probability of the original process X_k, $k \geq 0$, in Step 4. Such an estimate is then employed in Step 5 to derive a new estimate of the initial distributions of process $\bar{X}_k$, $k \geq 0$. Convergence is then tested in Step 6.

We must now prove that Algorithm 1 converges and that its output is the stationary distribution of X_k, $k \geq 0$. This is accomplished in the next theorem.

Theorem 4. *Let $\pi : S \to (0, 1)$ be the output of Algorithm 1. Then,*

$$\pi P = \pi, \tag{22}$$

where P is the transition matrix of process X_k, $k \geq 0$.

Proof. Eq. (19) provides an explicit way to evaluate the $B_l \bar{A}_l$ as defined in Eq. (16), without an explicit calculation of the $B_l \bar{A}_l$. This is demonstrated in Theorem 2. Hence, it follows that Step 5 of Algorithm 1 evaluates:

$$\mu_l^{k+1} = \mu_l^k B_l \bar{A}_l.$$

One can easily see that this is the iteration in (21) with $\mu_l = \pi$ and $\bar{P} = P$. Since we are dealing with a finite chain, it converges to the solution of (16) [e.g., 4, 11]. Consequently, Theorem 3 implies that Step 4 yields, upon convergence, the steady state distribution of X_k, $k \geq 0$. □

5. Choice of the partition scheme

This section investigates the computational effort in Algorithm 1, with a view at finding an efficient configuration of the partition scheme.

Theorem 5. *Assume the set F is partitioned into n subsets of the same cardinality, and suppose that for the set:*

$$V_j = \{i \notin F_l : p_{ij} > 0\},\ j \in F_l \tag{23}$$

we have $|V_j| \leq \frac{|S|}{n}$. Then, the optimal number of subsets within the partition in terms of computation effort should obey:

$$n \propto \sqrt{|S|},$$

where S is the state space of X_k, $k \geq 0$.

Proof. Let c be the computation effort in a single iteration of Algorithm 1. Then,

$$c \propto |S| + \frac{|S|^2}{n} + n^3. \tag{24}$$

The first term corresponds to the effort in Steps 2 and 4 of Algorithm 1. Step 2 makes n matrix calculations, each with $\dfrac{|S|}{n}$ multiplications, resulting in $|S|$ operations; the same applies to Step 4. Hypothesis (23) implies that each update in Step 5 requires $\left(\dfrac{|S|}{n}\right)^2$ operations; as we update n states, the total effort is proportional do the second term in the preceding equation. Finally, Step 3 performs a $n-$dimensional matrix inversion, which demands n^3 operations.

To conclude the proof, it is straightforward that a division into $|S|$ groups is equivalent to not decomposing the state space. If that is the case, one has to directly solve $\pi P = \pi$, which requires computations of the order of $|S|^3$. That is the upper bound at the extreme points. Now, assume $n > \sqrt{|S|}$, then the order of the computations in the third term of the equation above will surpass $|S|^{\frac{3}{2}}$. On the other hand, if $n < \sqrt{|S|}$, the order of the computations in the second term will exceed $|S|^{\frac{3}{2}}$. To conclude the proof, it suffices to see that $n = \sqrt{|S|}$ results in computations of the order of $|S|^{\frac{3}{2}}$ in both terms. □

Remark 3. *Observe, in Theorem 5, that the computational effort c should be interpreted as an order of magnitude rather than an exact value. With that in mind, the result of the theorem gives an order of magnitude of the ideal number of clusters in the partition.*

It is worth pointing out that Theorem 5 is devised under the sparsity hypothesis in (23). The problem of identifying the best partition scheme under general conditions remains open and will be a subject of future research.

6. Numerical simulations

We performed some experiments to illustrate the computational effort in Algorithm 1. We generated random transition matrices for the Markov chain and limited the number of neighbour states to 200 in each simulation, i.e., for each state $i \in S$ there are 200 states $j \in S$ such that $p_{ij} > 0$ and these states, as well as the transition probabilities, are randomly selected in each experiment; each probability is drawn from a uniform distribution and then the rows of the transition matrix are normalised to add up to one. For each experiment, we ran Algorithm 1 30 times, each time employed a new randomly generated chain of the same dimension. The computational times of Steps 2-5 were measured and averaged for a large number of possible partition schemes. For each partition scheme, the states were grouped in sets of equal cardinality, ordered according to the proximity of the state labels. For an illustrative toy example we refer to [3, Section V].

The numerical experiments were performed in a personal computer with a 3.0 GHz Intel Processor Core 2 Duo processor and 4GB of RAM memory, running Ubuntu Linux 16.04.2. For all numerical experiments the tolerance ϵ in Step 6 of Algorithm 1 was set to 10^{-4}.

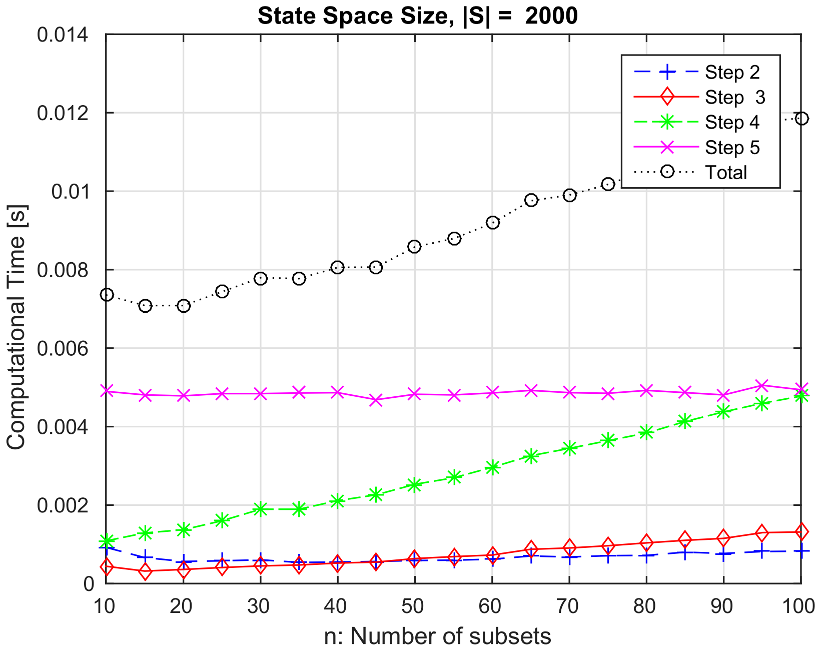

Figure 2: Number of subsets versus computational time

Figures 2 and 3 depict the computational time for Markov chains with state space cardinality of 2.000 and 5.000, respectively. The computational times for Steps 2-5 are individually measured and featured in each figure. One can see in Figure 2 that the total computational time varies slightly for schemes with up to 45 states, then it starts to rapidly increase. Note also that Step 4 - the disaggregation step - increases with the number of subsets. This is mainly due to the memory management process, i.e. the process of loading and unloading the matrices of each subset that are used in the calculations. On the other hand, the aggregation step - Step 5 - is more robust with regards to the number of subsets. Observe also that the results are consistent with the results in Theorem 5, whereby the optimal number of subsets would be around $\sqrt{2000} \approx 44$.

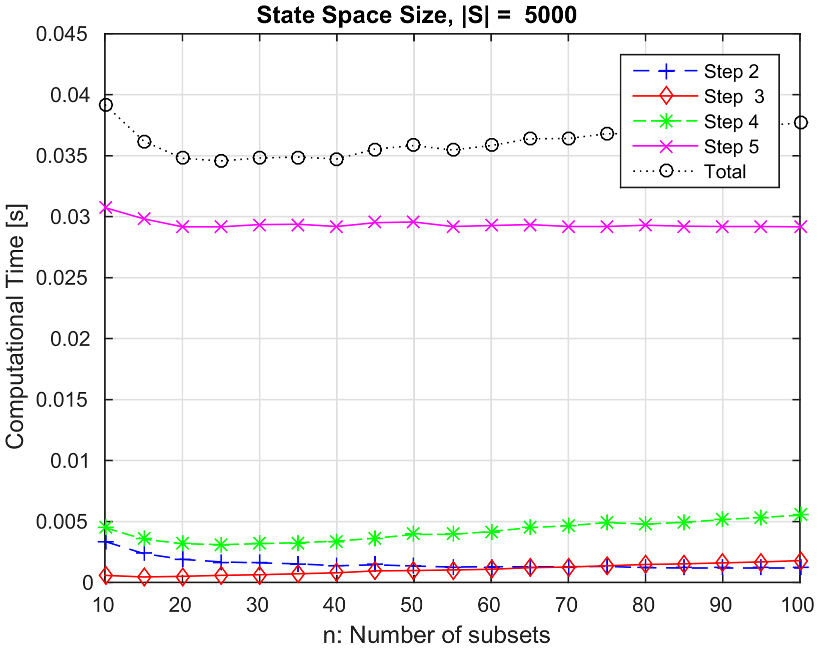

Figure 3: Number of subsets versus computational time

Figure 3 shows that, for a cardinality of 5000 the fastest results comes for a partition scheme with 20 subsets. Increasing the number of subsets, the results present only slight variations up to a cardinality of 60, which is slightly smaller than the optimal cardinality of approximately 70 according to Theorem 5. Observe also that Step 4 presents a more stable behaviour in this example. That is because the memory handling time becomes less significant once the overall execution time increases about one order of magnitude with respect to the previous example.

The results illustrate the theoretical results in the paper and suggest that the proposed algorithm tends to work better with a fairly small number of subsets. That is an interesting result that may be applied in further studies to extend the results to Markov decision processes, which also feature an optimisation step and which are particularly sensitive to the cardinality of the state space.

6.1. Comparison to standard time aggregation

To further analyse the proposed approach, we compare the performance of the proposed algorithm with standard time aggregation, making use of the same set of experiments introduced in the previous section. This is performed by noting that standard time aggregation can be seen as a partition scheme with two subsets. For the sake of simplicity, we divide the state space into two subsets of the same cardinality.

Table 1 shows the numerical results of the comparison. There, the times for $n = 2$ refer to standard time aggregation, whereas the times for n^* refer to the optimal partition scheme in the proposed approach. To

Table 1: Computational Time (s)

Steps	$\lvert S\rvert = 2000$		$\lvert S\rvert = 5000$	
	$n = 2$	$n^\star = 20$	$n = 2$	$n^\star = 25$
Step 2	0.0041	0.0006	0.0220	0.0017
Step 3	0.0004	0.0004	0.0005	0.0006
Step 4	0.0074	0.0014	0.0424	0.0031
Step 5	0.0067	0.0048	0.0390	0.0292
Total	0.0186	0.0071	0.1040	0.0346

be able to perform a meaningful comparison, the aggregation is performed over both subsets with a view to attain the steady state probabilities of all states in the Markov chain. Observe that the proposed method is significantly faster than standard time aggregation in both examples.

7. Concluding remarks

This paper introduces a new algorithm for the evaluation of the steady state probability of finite state space ergodic Markov chains. The algorithm makes use of a partition scheme which results in two distinct processes: one master semi-Markov process that models the transitions between the subsets of the partition and a set of auxiliary embedded Markov processes that register the first state visited at each sojourn within a given subset of the partition.

The normalised steady distributions within each subset in the partition can be obtained by adequately combining the steady state distribution of the auxiliary process with the expected number of visits to each state in the partition in an outbound trajectory within the subset. Then, by combining the normalised distributions with the stationary distribution of the master semi-Markov process, one is able to recover the stationary distribution of the original Markov chain.

The algorithm, which is shown to converge to the stationary distribution of the original process, enables the decision maker to find it by performing decentralised calculations within subsets of reduced dimension. Moreover, a single calculation of the outbound trajectories is needed, reducing the computational burden. Further studies should investigate the application of the results for policy evaluation in the policy iteration algorithm.

There are many issues to be considered in future research. One important issue is the choice of the partition scheme, which could be further investigated both in the absence of sparsity conditions and for particular structures of the Markov chain. Another research direction includes establishing bounds on the convergence rate of the proposed approach, possibly as a function of the number of subsets in the partition scheme.

Acknowledgements

The authors would like to thank the editor and the reviewers for their many constructive comments and suggestions, which certainly helped improve the paper. We are particularly grateful to one of the reviewers, which pointed out some interesting improvements for Sections 4 and 5.

This work was partially supported by the Carlos Chagas Filho Foundation for Research Support of the State of Rio de Janeiro, FAPERJ, under grant No. E-26/202.789/2015, and by the Brazilian national research council - CNPq, under grants 303543/2015-9 and 304801/2015-1.

References

[1] Aldhaheri, R.W., Khalil, H.K., 1991. Aggregation of the policy iteration method for nearly completely decomposable Markov chains. IEEE Transactions on Automatic Control 36, 178–187. doi:10.1109/9.67293.

[2] Arruda, E.F., Fragoso, M.D., 2011. Time aggregated Markov decision processes via standard dynamic programming. Operations Research Letters 39, 193 – 197. doi:DOI: 10.1016/j.orl.2011.03.006.

[3] Arruda, E.F., Fragoso, M.D., Ourique, F.O., 2017. Multi-partition time aggregation for Markov chains, in: 2017 IEEE 56th Annual Conference on Decision and Control (CDC), pp. 4922–4927. doi:10.1109/CDC.2017.8264387.

[4] Barker, G.P., Plemmons, R.J., 1986. Convergent iterations for computing stationary distributions of Markov chains. SIAM Journal of Algebraic and Discrete Methods 7, 390–398. doi:10.1137/0607044.

[5] Biswas, K., 2015. An iterative aggregation and disaggregation method for the steady state solution of large scale continuous systems. Computer Physics Communications 191, 25 – 32. doi:10.1016/j.cpc.2015.01.012.

[6] Brémaud, P., 1999. Gibbs fields, monte carlo simulation, and queues. Springer-Verlag, New York.

[7] Cao, X., Ren, Z., Bhatnagar, S., Fu, M., Marcus, S., 2002. A time aggregation approach to Markov decision processes. Automatica 38, 929–943.

[8] De Sterck, H., Manteuffel, T.A., McCormick, S.F., Nguyen, Q., Ruge, J., 2008. Multilevel adaptive aggregation for Markov chains, with application to web ranking. SIAM Journal on Scientific Computing 30, 2235–2262. doi:10.1137/070685142.

[9] Forestier, J., Varaiya, P., 1978. Multilayer control of large Markov chains. IEEE Transactions on Automatic Control 23, 298–305. doi:10.1109/TAC.1978.1101707.

[10] Geiger, B.C., Petrov, T., Kubin, G., Koeppl, H., 2015. Optimal Kullback-Leibler Aggregation via Information Bottleneck. IEEE Transactions on Automatic Control 60, 1010–1022.

[11] Haggstrom, O., 2002. Finite Markov Chains and Algorithmic Applications. Cambridge University Press, Cambridge.

[12] Haviv, M., 1987. Aggregation/disaggregation methods for computing the stationary distribution of a Markov chain. SIAM Journal on Numerical Analysis 24, 952–966.

[13] Heyman, D.P., Goldsmith, M.J., 1995. Comparisons between aggregation/disaggregation and a direct algorithm for computing the stationary probabilities of a Markov chain. ORSA Journal on Computing 7, 101–108.

[14] Howard, R., 1971. Dynamic Probabilistic Systems. volume II. John Wiley & Sons, New York.

[15] Huang, N., Ma, C., 2015. Parallel multisplitting iteration methods based on m-splitting for the pagerank problem. Applied Mathematics and Computation 271, 337 – 343. doi:10.1016/j.amc.2015.09.016.

[16] Kemeny, G.J., Snell, J.L., 1976. Finite Markov Chains. Springer-Verlag, New York.

[17] Krieger, U.R., 1995. On a two-level multigrid solution method for finite Markov chains. Linear Algebra and its Applications 223, 415 – 438. doi:10.1016/0024-3795(95)00166-O. honoring Miroslav Fiedler and Vlastimil Ptak.

[18] Leizarowitz, A., Shwartz, A., 2008. Exact finite approximations of average-cost countable Markov decision processes. Automatica 44, 1480–1487.

[19] Marek, I., Mayer, P., 1996. Iterative aggregation/disaggregation method for computing stationary probability vectors of Markov type operators. Computers & Mathematics with Applications 31, 27 – 40. doi:10.1016/0898-1221(95)00213-8. selected Topics in Numerical Methods.

[20] Norris, J.R., 1997. Control Techniques for Complex Networks. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, New York.

[21] Powell, W.B., 2007. Approximate dynamic programming: Solving the curses of dimensionality. John Wiley & Sons, Hoboken.

[22] Puterman, M.L., 1994. Markov decision processes: Discrete stochastic dynamic programming. John Wiley & Sons, New York.

[23] Stewart, W.J., 1994. Introduction to the numerical solution of Markov chains. Princeton University Press, Princeton NJ.

[24] Sutton, R.S., Barto, A.G., 1998. Reinforcement learning: An introduction. MIT Press, Cambridge.

[25] Wentzel, A., Freidlin, M., 1984. Random perturbations of dynamical systems. Springer, Berlin.