

PARALLEL COMPUTING FOR MARKOV CHAINS WITH ISLANDS AND PORTS

by

Amod Basnet

A dissertation submitted to the faculty of  
The University of North Carolina at Charlotte  
in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy in  
Applied Mathematics

Charlotte

2017

Approved by:

---

Dr. Isaac M. Sonin

---

Dr. Michael Grabchak

---

Dr. Jaya Bishwal

---

Dr. Arun Ravindran



## ABSTRACT

AMOD BASNET. Parallel Computing for Markov Chains with Islands and Ports. (Under the direction of DR. ISAAC M. SONIN)

We present recursive algorithms to calculate invariant distributions and fundamental matrices of Markov Chains specified by the “Islands & Ports” (IP) model. The state space of the IP model can be partitioned into “islands” and “ports”. An island is a group of states with potentially many connections inside of the island but a relatively small number of connections between islands. The states connecting different islands are called “ports”. Our algorithm is developed in the framework of “state reduction approach”, but the special structure of the state space allows computations to be performed in *parallel*.

## ACKNOWLEDGMENTS

First and foremost, I am grateful to Dr. Isaac M. Sonin without whose guidance this thesis would not be possible. The algorithms presented in this thesis are only possible because of his expertise in the field, his willingness to let me use his published works, his insightful approach to problem solving, and his joy for discovery. I am incredibly humbled to collaborate with him and contribute in every way possible. This thesis is an embodiment of his passion for academia and his dedication to mentorship, even during difficult times.

I would also like to thank my committee members Dr. Jaya Bishwal, Dr. Michael Grabchak and Dr. Arun Ravindran for their patience, encouragement, and willingness to guide me in the right direction. I am also thankful to my committee members for their invaluable suggestions during our presentation at the probability seminar. I also wish to thank Dr. Wei Cai, and his research group, for warmly receiving us at their weekly seminar.

## TABLE OF CONTENTS

LIST OF FIGURES	vii
LIST OF TABLES	viii
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: THE STATE REDUCTION APPROACH	7
1. State Reduction (SR)	8
2. Order Count	13
3. Grassman, Taskar and Heyman/Sheskin (GTH/S) Algorithm	14
CHAPTER 3: THE IP MODEL: THE INVARIANT DISTRIBUTION	15
4. IP model	15
5. SR approach for the IP model.	17
6. Main results: The IP algorithm	19
7. Performance Evaluation	28
8. Numerical Example	29
9. Application: approximations of nearly uncoupled MCs; Perturbation estimates	33
CHAPTER 4: THE IP MODEL: THE FUNDAMENTAL MATRIX	40
10. Fundamental matrix N	40
11. The FUNDQ algorithm	43
12. Order Count for the FUNDQ algorithm.	47
13. Numerical Example	48
14. Application: The fundamental Matrix for the IP Model	49
15. Numerical Example	55

CHAPTER 5: <a href="#">Appendix</a>	60
16. <a href="#">Proofs</a> . . . . .	60
17. <a href="#">Algorithms</a> . . . . .	65
<a href="#">REFERENCES</a>	68

## LIST OF FIGURES

FIGURE 1: An example of an IP model with 3 islands	16
FIGURE 2: Stages 1, 2 and 3 (step 1) for the case $k = 2$	20
FIGURE 3: State reduction and calculation of $P^*$ for the case $k = 2$	23
FIGURE 4: Transition matrices $P_1$ and $P_2$ for the case $k = 2$	25
FIGURE 5: Parallel features of the IP algorithm	26
FIGURE 6: An example of redistribution of small probabilities (1 state case)	35
FIGURE 7: An example of redistribution of small probabilities (2 states case)	36
FIGURE 8: Censored MCs	42
FIGURE 9: Example of stages 1 and 2 for the case $k = 2$	52
FIGURE 10: Decompositions of the matrix $N$ for the case $k = 2$	53

## LIST OF TABLES

TABLE 1: Transition matrix $P$	30
TABLE 2: Transition matrix $P^*$	31
TABLE 3: Transition matrices for models $M_1, M_2$ and $M_3$	32
TABLE 4: Matrices $P_1$ and $N_1$	48
TABLE 5: Fundamental matrix $N_4$ for $Q_4$	48
TABLE 6: Insertion of states 3, 1 and 6	49
TABLE 7: Matrices $P$ and $Q$ (dotted in red)	56
TABLE 8: Fundamental matrix $N(F_1(\text{bronze}), F_2(\text{sky blue}), F_3(\text{light green}))$	57
TABLE 9: Substochastic matrix $Q^*$	57
TABLE 10: Fundamental matrix $N^*$	57
TABLE 11: Insertion of state 4	57
TABLE 12: Insertion of state 7	58
TABLE 13: Insertion of state 8	58
TABLE 14: Insertion of state 11	58
TABLE 15: Insertion of state 12	59



## CHAPTER 1: INTRODUCTION

Discrete Time Markov Chains (DTMCs), or simply Markov Chains (MCs), are used to model various systems of interest. Recently there has been a growing interest in the study of MCs with “large” and “very large” state spaces. These MCs appear in applications such as Web search, genetic modeling and queueing theory. The state spaces of such MCs may have thousands, and even millions of states. For example, in population genetics, the so-called Wright-Fisher model involves transitions matrices that can be of size  $(2N + 1) \times (2N + 1)$ ,  $N \in [20, 1000]$  [22]. In Web search, Google’s PageRank algorithm uses a  $n \times n$  transition matrix  $P$  associated with an ergodic MC to rank Web pages. The transition matrix is given by  $P = cM + (1 - c)U$ , where  $U = \frac{1}{n}ee^T$  is a uniform matrix,  $e$  is a vector of all ones,  $c \in (0, 1)$  and  $M$  is the hyperlink matrix [17]. According to Google  $c = 0.85$  which is the probability that a surfer jumps to a random page [2]. The matrix  $P$  is an example of a very large ergodic MC with billions of Web pages as nodes. The PageRank vector is the invariant distribution,  $\pi$ , of  $P$  which approximates the long run probability that a random surfer visits a particular Web page. In other words, it ranks the Web pages based on the surfer’s random behavior. A survey of various other modifications like updating and accelerating the calculation of the PageRank vector can be found in [17]. Google uses the power method to compute  $\pi$ , which it claims converges after about 50 iterations. Because the MC is ergodic, power method ensures that  $P$  converges a limiting matrix  $A = e\pi^T$ , almost surely.

Although we can use classical methods, like the power method, to analyze large MCs, such

methods may be inefficient for large state spaces. As an alternative, we can develop methods that exploit specific structures that may be present in the state spaces. For example, we can partition the state spaces of some MCs into “clusters”. These are groups of states such that states inside each group have strong connections, but each group is only weakly connected to other groups. In such cases, we can use cluster-based methods to calculate various characteristics of these MCs. In the cluster-based approach, we calculate “local” characteristics of each cluster first, and then use them to approximate “global” characteristics for the whole chain [19]. In the literature, MCs with such structure are said to be “nearly uncoupled”<sup>1</sup> and the cluster-based approach is also called the aggregation/disaggregation (AD) approach.

Following this approach, numerous algorithms have been proposed to approximate the invariant distribution,  $\pi$  [6, 7]. A similar approach is also taken in the so-called the BlockRank algorithm — a modified version of the PageRank algorithm to compute the PageRank vector — developed after experiments indicated clustering of Web pages in the form of “hosts” [14]. A host page is a Web page inside of which a large number of other Web pages sit. Such pages include university domain like `www.uncc.edu`, the IBM domain or various news sites. Variants of the AD approach to compute the PageRank vector, for example, the partial aggregation method (PAM), fast two-stage algorithm (FTSA) and others, are summarized in [2]. In addition, the effect of analytic perturbation on the PageRank vector can also be found in [2, 1, 3].

To apply cluster-based methods, clusters must first be identified in the state spaces. There are many methods to identify clusters for MCs [18, 20]. However, they are not always accurate and only give an approximate number of clusters, yet these cluster-based

---

<sup>1</sup>These MCs are also known as Nearly Completely Decomposable (NCD) MCs

methods provide a convenient way to analyze a large MC. Moreover, calculations can be done even faster if we can take advantage of parallel computing. By “parallel computing” we mean using multiple processors to perform calculations involving, in our case, a large transition matrix.

In this thesis, we study finite, ergodic Markov models with large “state spaces”. The “state space” of our model is partitioned into islands where each island is further divided into “interior states” and “ports”. Interactions between any two “islands” are only allowed to occur through the “ports” (comprised of states that interact with other “islands”), whereas the interior states only interact with states inside each island. Here, we assume that the number of ports cannot exceed the size of the biggest island. The configuration of states in our model, as “islands” and “ports”, is a special case of clustering that can be found in the “state spaces” of many Markov models. In order to emphasize this unique configuration of states in the “state space” of our model, we refer to such model as the “Islands & Ports” (IP) model.

Our main objective is to develop algorithms to calculate essential characteristics for MCs specified by the IP model. The characteristics we are interested in are given by the invariant distribution,  $\pi$ , and the fundamental matrix,  $N$ . Both of these characteristics are important in applications. We already know that the distribution  $\pi$  gives the long run probability distribution of a MC. The entries of the matrix  $N$  gives the expected number of visits to states inside a certain (nonabsorbing) subset of the state space before exiting it at some finite time. By taking advantage of the assumptions we made about the state space of our model, our algorithms allow most of the calculations involved in computing these characteristics to be performed in *parallel*.

The outline of the thesis is as follows: In Chapter 2 we present some preliminary results.

In particular, we discuss the so-called State Reduction (SR) approach. The SR approach is the building block in all our algorithms. In Chapter 3 we discuss the IP model and develop a three-stage recursive algorithm to calculate  $\pi$  for this model. We refer to our algorithm as the *IP algorithm*.

**(Algorithm 1: The IP algorithm)** This is a recursive algorithm that calculates  $\pi$  in three stages. Even though it is a sequential algorithm, calculations for the most expensive stages — stages 1 and 3 of the algorithm can be done in parallel. This algorithm has time complexity of order  $n^3/k^2$ , where  $n$  is the total number of states and  $k$  is the total number of islands in our model. In comparison, solving for  $\pi$  using the GTH/S algorithm, which is the most accurate algorithm, has time complexity of order  $n^3$ .

Under certain conditions, the IP model reduces to a nearly uncoupled Markov model<sup>2</sup>. Despite such similarities, our model is quite general and, in many cases, very different from the nearly uncoupled Markov model. Even though there exists a rich literature on relatively newer algorithms to compute  $\pi$  for nearly uncoupled MCs [16, 33, 6]; to our knowledge, none of these algorithms allow parallel computations to the extent our algorithm does. In addition, most of these methods are iterative whereas our algorithm gives exact results.

Because transition probabilities in our model are more restrictive than that in the nearly uncoupled models, we also discuss the possibility of extending our results to approximate invariant distributions for the nearly uncoupled MCs. For this, we perform some preliminary simulations with our model. We also discuss the effect of our approximations using previously known bounds. These bounds appear in the perturbation theory of stochastic matrices. To elaborate this further, the invariant distributions of MCs under perturbations have been rigorously studied since as early as 1960. These perturbations include linear

---

<sup>2</sup>We refer to a Markov model that specifies nearly uncoupled MCs as the “nearly uncoupled model”

perturbations [24, 11, 5] as well as analytical perturbations [3]. Under linear perturbations, most bounds for  $\pi$  include the so-called fundamental matrix  $V$  for ergodic MCs. A comparison of the bounds for linear perturbations can also be found in [5]. Our approximations of the nearly uncoupled model only includes a simple case of linear perturbation.

There are many algorithms to calculate the matrix  $V$  [13, 30, 10]. However, the fundamental matrices  $N$  for transient MCs are not studied as extensively. One of the reason why this is so, may be due to the fact that calculation of  $N$  is simpler. The calculation of  $N$  involves a matrix inversion. If the state space is large, inversion is computationally expensive. There are few recursive algorithms proposed to calculate such matrices [12]. In Chapter 4, we develop another algorithm to compute  $N$  for any substochastic matrix  $Q$ .

**(Algorithm 2: The FUNDQ algorithm)** This recursive algorithm consists of two stages: In the forward stage, we apply the SR approach to eliminate a subset of states from the matrix  $Q$ . The backward stage involves computing a sequence of fundamental matrices using the recursive formula given in Proposition 15. The time complexity of this algorithm is approximately of order  $k(n^2) + (n - k)^3$ ,  $1 \leq k < n$ , where  $n$  is the number of states in the subset being considered.

We apply the FUNDQ algorithm to develop another algorithm to calculate the fundamental matrix  $N$  for the IP model.

**(Algorithm 3: The IP FUND algorithm)** This is a recursive algorithm which has a similar three stage structure as that in the IP algorithm. The calculations in stages 1 and 3 of the algorithm can also be done in parallel.

Additionally, we also discuss another set of matrices  $N^{(1)}, N^{(2)}, \dots, N^{(k)}$ , where  $k$  is the number of islands in the model, that result, without any extra calculations, from the IP algorithm in Chapter 3.

In the Appendix, we give some proofs to the (previously published) lemmas and propositions we use in this thesis. We also provide pseudo-codes to the three algorithms we present in Chapters 3 and 4, in the Appendix.

## CHAPTER 2: THE STATE REDUCTION APPROACH

Let  $M = (S, P)$  be a homogeneous, ergodic Markov model, where  $S = \{0, 1, 2, \dots, n\}$ ,  $n \leq \infty$ , is the discrete state space and  $P$  is the transition matrix. In this chapter, we discuss the state reduction (SR) approach for MCs. This approach was first used in the GTH/S algorithm (see section 3) to compute  $\pi$  for model  $M$ . Since then it has been used to recursively calculate many important characteristics of MCs. It served as the building block in many algorithms such as those used to calculate fundamental matrices of transient MCs [12] and for ergodic MCs [30], in matrix-analytic methods for block-structured matrices [34, 8], to solve optimal stopping problems [26, 31, 32]; [28, 29], and to calculate mean passage times [13]. The SR approach also serves as the building block in our algorithms in Chapters 3 and 4. Our discussion of the SR approach in this chapter is similar to that given in [27] and [29]. The SR approach has a two-stage structure:

1. (**Forward stage**) Let MC  $(Z_n)$  be specified by model  $M$ . And let  $D \subset S$ . Then we can obtain another MC  $(Y_n)$  (see Lemma 1) from the model  $M$  by eliminating set  $D$  from  $S$ . The path of the MC  $(Y_n)$  coincides with that of the original MC  $(Z_n)$  in set  $S \setminus D$ . The MC  $(Y_n)$  is called a *censored* MC specified by the  $D$ -reduced model  $M_D = (S \setminus D, P_D)$ , where the matrix  $P_D$  is the new transition matrix for model  $M_D$ . The concept of censored MCs is an insightful idea of Kolmogorov and Doëblin. According to this idea, the censored chain observed in the subset of the original state space is still a MC. The states that do not belong to the observation set of the censored MC are said to be *eliminated*. One can calculate various characteristics for MC  $(Y_n)$ .

2. (**Backward stage**) States that were previously eliminated from set  $D$  in the forward stage are “*inserted*” or “*restored*”. Using results calculated for the censored MC  $(Y_n)$ , one can also calculate the characteristics of the original MC  $(Z_n)$ .

### 1 State Reduction (SR)

Let MC  $(Z_n)$  be a  $S$ -valued MC specified by model  $M$  with some initial distribution  $\mu_0$ .

If we let  $D \subset S$ ,  $C = S \setminus D$ , we can decompose matrix  $P$  as the first matrix below

$$P = \begin{array}{c} \begin{array}{cc} (D) & (C) \\ \begin{bmatrix} Q & T \\ R & K \end{bmatrix} \end{array} \end{array}, \quad P^D = \begin{array}{c} \begin{array}{cc} (D) & (C) \\ \begin{bmatrix} 0 & NT \\ 0 & P_D \end{bmatrix} \end{array} \end{array}, \quad W^D = \begin{array}{c} \begin{array}{cc} (D) & (C) \\ \begin{bmatrix} QN & NT \\ RN & P_D \end{bmatrix} \end{array} \end{array}. \quad (1)$$

where the substochastic matrices  $Q := \{p(x, y) : x, y \in D\}$ ,  $T := \{p(x, y) : x \in D, y \in C\}$ ,  $R := \{p(x, y) : x \in C, y \in D\}$  and  $K := \{p(x, y) : x, y \in C\}$  describe transitions inside of set  $D$ , from sets  $D$  to  $C$ , from sets  $C$  to  $D$ , and inside of set  $C$ , respectively. Let  $\tau_0 = 0$  and  $\tau_{n+1} = \min\{n > \tau_n : Z_n \in C\}$ . Then  $\tau_0, \tau_1, \tau_2, \dots$  are the times of the zero, the first, the second and so on, *visits* to set  $C$ . Let  $Z_0 \in C$  and consider the random sequence  $(Y_n = Z_{\tau_n})$ . According to Lemma 1,  $(Y_n)$  is a censored MC specified by model  $M_D = (C, P_D)$ . The transition matrix  $P_D$  is calculated using formula (2).

Lemma 1 follows from the strong Markov property and standard probabilistic reasoning.

We credit this Lemma to the works of Kolmogorov and Doëblin [27].

**Lemma 1.** (*SR Lemma*)

*The random sequence  $(Y_n)$  is a Markov chain in model  $M_D = (C, P_D)$ , where the transition matrix  $P_D = \{p(x, y) : x, y \in C\}$  is given by the formula*

$$P_D = K + RU = K + RNT. \quad (2)$$



**Remark 1.** The matrix  $U = N_D T$  (see Lemma 46 Chapter 4) is a matrix of the distribution of the MC  $(Z_n)$  at the time of the first exit to  $C$ , and  $N_D \equiv N$  is the fundamental matrix of the sub-stochastic matrix  $Q$ . Matrix  $P_D$  is also known as the stochastic complement of  $K$  in  $P$  [19].

A proof based on probability theory is given in [15]. An algebraic proof of Lemma 1 can be found in [19].

We can represent the matrix  $P_D$  in formula (2) as

$$p_D(x, y) = p(x, y) + \sum_{j \in D} p(x, j) P_j(Z_{\tau_1} = y), \quad (3)$$

where  $\tau_1 = \min\{n > 0 : Z_n \in C\}$ . In formula (2), the matrix  $N$  can be calculated as  $N = \sum_{n=0}^{\infty} Q^n = (I - Q)^{-1}$ , where  $I$  is an  $|Q| \times |Q|$  identity matrix. We discuss the calculation of  $N$  in Chapter 4 in greater detail. Instead of calculating the matrix  $P_D$  by eliminating  $D$  in one step, thus requiring a matrix inversion to calculate  $N$ , we can, instead, eliminate it in  $|D|$  steps by eliminating one state  $z \in D$  at a time. To elaborate this further, assume set  $D$  only consists of one non-absorbing point  $z$ . In this case, each column of the matrix  $P_D = P_{\{z\}}$  in (2) can be written as

$$p_{\{z\}}(x, \cdot) = p(x, \cdot) + p(x, z)n(z)p(z, \cdot), \quad x \in C, \quad (4)$$

where  $n(z) = 1/(1 - p(z, z))$  is the one-dimensional fundamental matrix. We say that the matrix  $P_{\{z\}}$  is obtained from the matrix  $P$  in one *iteration*. If  $|D| = d$ , then we can eliminate one state from  $D$  in each iteration using (4) and obtain  $P_D$  in  $d$  iterations. Note that  $n(z)$  in (4) gives expected number of visits to  $D = \{z\}$  by MC  $(Z_n)$  before exiting  $D$  at time  $\tau_1$ . The matrix  $P_D$  describes the behavior of the censored MC  $(Y_n)$  with values only in set  $C$ .

Sometimes, for calculations, it is more convenient to have the initial and the reduced

stochastic matrices of equal full size. For this, the matrix  $P_D$  is extended in [28, 29] to the full size  $|S| \times |S|$ , stochastic matrix  $P^D$  (second matrix in (1)) by assuming that the MC ( $Y_n$ ) can have initial points in  $D$ , but after one step it will be in  $C$ , and will remain in  $C$ . Equivalently, if  $Z_0 \in D$ , then the MC ( $Z_n$ ) will exit at time  $\tau_1$ , and for all  $n > \tau_1$ , it will remain in  $C$  as  $D$  is eliminated. Thus  $p^D(z, z) = 0$ , for all  $z \in D$ . For  $z \in D, y \in C$ , using the first equality in formula (2), we obtain

$$p^D(z, y) = p(z, y) + \sum_{z' \in D} p(z, z')u(z', y). \quad (5)$$

By noting that  $U = NT$ , in matrix form, the right hand side of (5) equals  $T + QU = T + QNT = (I + QN)T = NT$ , where the last equality follows from equation (48) Chapter 4. For  $x, y \in C$ , the corresponding distribution is given by the submatrix  $P_D$ .

To obtain the matrix  $P^D$  in  $d$  iterations, we apply formula (4) to the full size matrices. Thus, if we denote the initial matrix  $P$  as  $P_1$  and the matrix obtained after eliminating one state  $\{z\}$  as  $P_2$ , then the columns of matrix  $P_2$  can be written as

$$p_2(\cdot, z) = 0, \quad p_2(\cdot, y) = p_1(\cdot, y) + p_1(\cdot, z) \frac{p_1(z, y)}{1 - p_1(z, z)}, \quad y \neq z. \quad (6)$$

To distinguish between the matrices in models  $M_D$  and  $M^D$ , we introduce the following notation: *matrices* in reduced models are denoted by subscripts, i.e.,  $M_D, P_D, N_D$  and so on; whereas *matrices* in full size model are denoted with superscripts, i.e.  $M^D, P^D, N^D$  and so on.

For computations, we prefer to calculate another full (nonstochastic) matrix  $W^D$  (third matrix in 1). Before we explain why this is so, we first explain how the matrix  $W^D$  is calculated and compare it with previously discussed matrices  $P_D$  and  $P^D$ . The matrix  $W^D$  is calculated by using the *elimination formula* given in (7), by applying formula (6) to all

states, including all the states that were previously eliminated. If we let  $W_1 = P$ , then we can obtain a matrix  $W_2$  after eliminating one state  $z$  using the elimination formula

$$w_2(\cdot, y) = p_1(\cdot, y) + p_1(\cdot, z) \frac{p_1(z, y)}{1 - p_1(z, z)}, \quad y \neq z. \quad (7)$$

Now let  $D = \{1, 2, \dots, k, \dots, d\} \subset S$ . The matrix  $W^D$  is obtained applying formula (7) by eliminating set  $D$  in  $d$  iterations. Let matrix  $W_k$  be the full matrix obtained after eliminating first  $k$  states from set  $D$ , then using formula (7) the columns of matrix  $W_{k+1}$  can be calculated as

$$w_{(k+1)}(\cdot, y) = w_{(k)}(\cdot, y) + w_{(k)}(\cdot, z) \frac{w_{(k)}(z, y)}{1 - w_{(k)}(z, z)}, \quad y \neq z. \quad (8)$$

Although the form of the matrix  $W^D$  given in (1) is quite transparent, proving it is quite difficult. Formula (7) and the interpretation of the nonzero columns in  $W^D$  is given in [28, 29]. By the definition of the matrices  $W^D$  and  $P^D$ , their columns for  $y \notin D$  coincide but the matrix  $P^D$  has zero columns for  $y \in D$ .

Now let  $P_1 = P'_1 = P$ . From our discussion so far, we can eliminate  $D$  by computing three matrices  $P_D$ ,  $P^D$  and  $W^D$ . Note that  $P_D \subset P^D \subset W^D$ .

$$P_1 \implies P_2 \implies \dots P_d \implies P_{d+1} = P_D \text{ (stochastic)}, \quad (9)$$

$$P'_1 \implies P'_2 \implies \dots P'_d \implies P'_{d+1} = P^D \text{ (stochastic)}, \quad (10)$$

$$W_1 = P_1 \implies W_2 \implies \dots W_d \implies W_{d+1} = W^D \text{ (nonstochastic)}, \quad (11)$$

where the  $(d+1)^{th}$  matrix is the matrix obtained after  $D$  is completely eliminated, in  $d$  iterations, from the state space  $S$ . The matrices  $P'$  and  $W$  are full matrices calculated using (6) and (7), respectively.

**Remark 2.** *Unless otherwise stated, all of our calculations involve full matrices  $W$  as given*

in (11). We obtain the matrix  $P_D$  from  $W^D$  which is the submatrix in  $W^D$  corresponding to states in  $S \setminus D$ .

We now explain why we prefer to use full matrices  $W$  in our calculations. In the two-stage structure of the SR approach, the forward stage involves the elimination of a set of states, say  $D$ , from  $S$ . As explained above, the matrix  $P_D$  is a stochastic matrix describing transitions in the reduced model  $M_D$ . The idea of the forward stage of the SR approach is straightforward — it allows for dimension reduction, that is, we reduce the matrix  $P$  into a smaller stochastic matrix  $P_D$ . We do calculations using this matrix  $P_D$ . Sometimes, it is the characteristics of the original MC ( $Z_n$ ) that we are interested in. This requires *inserting* states that were eliminated in the forward stage. In particular, we want to *restore* any of the eliminated states, *in any order*, without taking into account the *order* in which that state was eliminated during the forward stage. This can be achieved, starting with the matrix  $W^D$  and using the “insertion” formula in Lemma 2; See Appendix for proof.

**Lemma 2.** [28, 29] *If matrix  $W_{k+1}$  is obtained from matrix  $W_k$  by the elimination formula (8), then the matrix  $W_k$  can be obtained from matrix  $W_{k+1}$  by*

$$w_k(\cdot, y) = w_{k+1}(\cdot, y) - w_{k+1}(\cdot, z) \frac{w_{k+1}(z, y)}{1 + w_{k+1}(z, z)}, \quad y \in S. \quad (12)$$

To distinguish between the matrices obtained in the forward stage by applying formula (8) from the matrices calculated in the backward stage by applying formula (12), we denote the latter as matrix  $\hat{W}$ . As a consequence of the insertion formula (12), we obtain

$$\hat{W}_1 = W_1 = P_1 \Leftarrow \hat{W}_2 \Leftarrow \cdots \Leftarrow \hat{W}_{d+1} = W^D, \quad (13)$$

where the full matrix  $\hat{W}$ , other than  $W^D$  and  $W_1$ , in (11) and (13) need not be the same because the order of elimination and insertion can vary.

The following Lemma follows from Lemma 1. It was also proved in [15]; See Appendix for a proof.

**Lemma 3.** *Let  $\pi = (\pi_1, \pi_2, \dots, \pi_s, \dots, \pi_n)^T$  be the invariant distribution of the original MC  $(Z_n)$ . Let  $|C| = s$  and let  $\bar{\pi}_C = (\bar{\pi}_1, \bar{\pi}_2, \dots, \bar{\pi}_s)^T$  be the invariant distribution for MC  $(Y_n)$ . Then*

$$\bar{\pi}_i = \frac{\pi_i}{\sum_{i \in C} \pi_i} \quad \text{and} \quad \bar{\pi}_C P_D = \bar{\pi}_C, \quad i = 1, 2, \dots, s. \quad (14)$$

## 2 Order Count

We do a straightforward analysis of the time complexity in the calculation of the matrices  $P_D$  in (9),  $W_D$  in (11) and  $\hat{W}$  in (13). Suppose  $P$  be a  $n \times n$  matrix.

**(Calculation of matrix  $P_D$ )** Each step of the state elimination from the sequence given in (9) requires  $(n-1)^2$  additions, 1 subtraction,  $(n^2-1)$  multiplications and 1 division, so, eliminating  $n$  states requires  $n^3 - 2n^2 + 2n$  additions and subtractions,  $n^3$  divisions and multiplications. If we count only multiplication and divisions, it has approximately the time complexity of order  $n^3$ .

**(Calculation of matrix  $W^D$ )** Each step of the state reduction from the sequence given in (11) requires  $n^2$  additions, 1 subtraction,  $n^2$  multiplications and 1 division. Thus, for  $n$  steps, the time complexity is also of order  $n^3$ .

**(Calculation of matrix  $\hat{W}_1$ )** Each step of the state insertion for the sequence in (13) requires 1 addition,  $n^2$  subtractions,  $n^2$  multiplications and 1 division. The insertion of  $n$  states requires  $n^3 + n$  additions and subtractions,  $n^3 + n$  multiplications and divisions. If we only count the multiplications and divisions, then the complexity is also of order  $n^3$ .

### 3 Grassman, Taskar and Heyman/Sheskin (GTH/S) Algorithm

We now discuss the GTH/S proposed, independently, by Grassman, Taskar and Heyman [9], and by Sheskin in 1985 [25], to calculate  $\pi$  for a  $n \times n$  transition matrix  $P$  of an ergodic model. The distribution  $\pi^T = [\pi_1 \ \pi_2 \ \dots \ \pi_n]$  is the solution of the linear system

$$\pi^T = \pi^T P, \quad \text{and} \quad \sum_{i=1}^n \pi_i = 1. \quad (15)$$

This algorithm is based on the general idea of the SR approach. In the forward stage of the GTH/S algorithm, we calculate a sequence of stochastic matrices  $P_1, P_2, \dots$  and  $P_n$  by eliminating one state in each iteration using Lemma 1. The elimination of a state at the  $j^{\text{th}}$  step,  $1 < j \leq n$ , corresponds to a transformation of the initial MC to a censored MC with transition matrix  $P_j$ . The censored MC is the original MC observed only in the set that is not eliminated. As a result, every matrix  $P_j$  in this sequence has dimension one less than the previous matrix  $P_{j-1}$ . After  $n - 1$  states are eliminated, the last matrix  $P_n$  has a single state with a trivial invariant distribution  $\pi_n = \{1\}$ . In the backward stage, a sequence of invariant distributions  $\pi_i$  for  $i = n - 1, \dots, 2, 1$ , is calculated in the reverse order.

$$\begin{aligned} P = P_1 &\implies P_2 \implies \dots P_s \dots \implies P_{n-1} \implies P_n = \{1\} \\ \pi = \pi_1 &\longleftarrow \pi_2 \longleftarrow \dots \pi_s \dots \longleftarrow \pi_{n-1} \longleftarrow \pi_n = \{1\} \end{aligned} \quad (16)$$

We apply the GTH/S algorithm for modified transition matrices in different stages of our algorithm in Chapter 3. This algorithm involves approximately  $\frac{2}{3}n^3 + 2n^2 - \frac{8}{3}$  operations. It has been shown to be more stable and accurate [21]. A pseudo-code presented by Sheskin [25] is given in the Appendix.

## CHAPTER 3: THE IP MODEL: THE INVARIANT DISTRIBUTION

### 4 IP model

We consider a special class of ergodic Markov models  $M = (S, P)$ , where  $S$  is the state space and  $P$  is the transition matrix. The state space is finite and discrete,  $S = \{1, 2, \dots, n\}$ ,  $n < \infty$ , and it can be partitioned into  $k$  disjoint “islands”  $L_i$ ,  $S = \bigcup_{i=1}^k L_i$ . Here, we use  $+$  to denote a union of disjoint sets. Each island  $L_i$  is also a disjoint union of “interior” states  $R_i$  and “ports”  $T_i$ , i.e.  $L_i = R_i + T_i$ . We allow transitions between two states to occur only if these states belong to the same island or to ports, i.e. if  $p(i, j) > 0$ , then either  $i, j \in L_r$  or  $i \in T_s, j \in T_r$ , for some  $1 \leq r, s \leq k$ . Let  $T$  be the union of all ports,  $T = \sum_{j=1}^k T_j$  and  $|T| = t$ . We assume that  $t$  is of the same order as the size of the biggest island and, for simplicity, we assume that  $t = |L_i| = m$ ,  $1 \leq i \leq k$ ; See Fig.1 for an example.

It is easy to see that our model is different from the clusters (or nearly decoupled MCs) discussed in Chapter 1. In our model, transitions between two islands can occur only through the ports. Unlike the transitions in nearly decoupled MCs, transitions between any two islands in our models are certainly more restrictive, however, these transitions do not have to be weak. This means that the islands  $L_i$  and  $L_j$ ,  $i \neq j$ , in our model can be strongly or weakly linked to each other. In this sense, our model is quite general. If all the transitions between islands are close to zero, then the IP model reduces to a nearly uncoupled model.

Our model can describe many real scenarios. For example, if we consider trading zones

as islands and large ports — limited in number— as ports, then we can use our model to analyze trade routes of modern day tankers or large container ships. We can also apply our model to study interactions between international and multinational agencies. Such agencies have many local branches (in countries, regions, etc.) but interactions across agencies only occur through the central governing bodies. In fact, our model can be used to describe any multi-layered, diversified organization (e.g. banks) because communication channels in such organizations tend to be structured in a similar way.

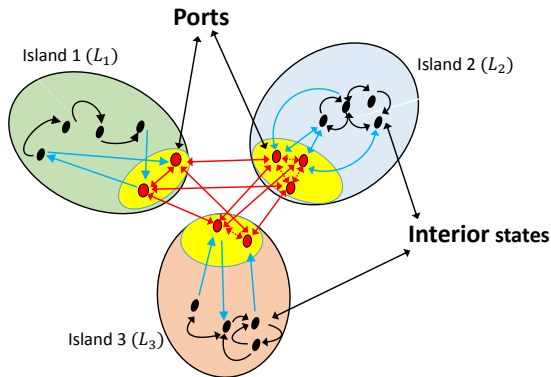


Figure 1: An example of an IP model with 3 islands

Our main goal is to present the IP algorithm which is used to compute  $\pi$  for the IP model. The main advantage of this algorithm, due to the special features of the state space, is that most of the calculations solving for  $\pi$  are performed in *parallel*. Note that the possibility of parallel calculations in our model is not obvious. The invariant distribution on each island depends on transition probabilities on all of the other islands, even when transitions between islands are weak.

The parallel feature of this algorithm also allows making external changes inside an island without significantly affecting the calculations in the other islands. Such changes include altering the number of interior states or varying their transition probabilities. And parallel



features of the algorithm are also preserved even if we introduce new islands into our model.

In this sense, updating our algorithm to new information is relatively easy.

The invariant distribution  $\pi$  for any ergodic MC, with a  $n \times n$  transition matrix  $P$ , is calculated by solving the linear system

$$\pi^T = \pi^T P \iff \pi^T (I - P) = 0, \quad (17)$$

where  $\pi^T = [\pi_1 \ \pi_2 \ \dots \ \pi_n]$  satisfying  $\sum_{i=1}^n \pi_i = 1$ ,  $\pi_i > 0$ , and  $I$  is a  $n \times n$  identity matrix.

There are many direct and iterative methods to solve (17). Some direct methods include the GTH/S algorithm, the LU factorization (with some adjustment), and iterative methods include the Gauss-Seidel, the Jacobi method, the power method and other hybrid schemes [33, 23]. Algorithms to calculate  $\pi$  for nearly uncoupled MCs include iterative schemes such the KMS algorithm [16] and the KMS-GTH/S hybrid algorithm [6].

## 5 SR approach for the IP model

We now present the recursive formula used in the backward stage of the GTH/S algorithm. We state this formula in the form given as Lemma 1 in [27]; See Appendix for proof.

**Lemma 4.** *Let  $M_1 = (S_1, P_1)$  be an ergodic Markov model,  $S_2 = S_1 \setminus \{z\}$ , and  $M_2 = (S_2, P_2)$  be  $z$ -reduced Markov model. Let  $S_2$  and state  $z$  communicate in model  $M_1$ , i.e., there are states  $i, j \in S_2$  such that  $p_1(i, z) > 0$  and  $p_1(z, j) > 0$ . If the invariant distribution  $\pi_2(\cdot)$  exists in model  $M_2$ , the invariant distribution  $\pi_1(\cdot)$  also exists in  $M_1$  and can be calculated by formulas*

$$(i) \ \pi_1(y) = \alpha_1 \pi_2(y), \text{ for all } y \in S_2 \text{ ,and } \alpha_1 = 1 - \pi_1(z), \quad (18)$$

$$(ii) \ \pi_1(z) = \beta_1 \sum_{y \in S_2} \pi_2(y) p_1(y, z) = \beta_1 R_1, \quad (19)$$

where  $\alpha_1 = \frac{1}{1+n_1 R_1}$ ,  $n_1 \equiv n_1(z) = \frac{1}{1-p_1(z,z)}$  and  $R_1 \equiv R_1(z) = \sum_{y \in S_2} \pi_2(y) p_1(y, z)$ .

We make some useful remarks based on our results thus far.

**Remark 3.** Both Lemma 3 and Lemma 4 imply that invariant distributions of states in the reduced model are proportional to their distributions in the original model. However, Lemma 4 gives an explicit recursive formula that relates any two consecutive distributions in (16). This is particularly useful for calculations.

**Remark 4.** Instead of a single state  $\{z\}$ , now suppose we eliminate a set  $D \subset S_1$  from model  $M_1 = (S_1, P_1)$  to obtain model  $M_2 = (S_2, P_2)$ , where  $S_2 = S_1 \setminus D$  and  $P_2$  is given by formula (2). Lemma 4 implies that  $\pi_1(y) = e\pi_2(y)$  for all  $y \in S_2$  and for some scalar  $e > 0$ . Similarly, if  $L \subseteq S_2$ , then  $\pi_1(L) = e\pi_2(L)$ , where  $\pi_i(L) = \sum_{x \in L} \pi_i(x)$ ,  $i = 1, 2$ . We can use Lemma 3 to show that  $e = \sum_{x \in S_2} \pi_1(x)$ .

We will apply Remark 4 in the extension step of Stage 3 of our algorithm.

**Remark 5.** Suppose that we know the invariant distribution  $\pi_s$  in (16) for some stochastic matrix  $P_s$ ,  $1 \leq s \leq k$ , calculated after  $s - 1$  iterations of the initial matrix  $P$ . Using Lemma 4, we can also recursively calculate the invariant distributions  $\pi_j$ ,  $1 \leq j < s$ , in the backward stage of the GTH/S algorithm starting with distribution  $\pi_s$ .

The following lemma follows directly from Lemma 1.

**Lemma 5.** Let Markov model  $M_1 = (S_1, P_1)$  be given. Suppose  $S_2 = S_1 \setminus \{z\}$ , and let  $M_2 = (S_2, P_2)$  be the  $z$ -reduced model obtained from model  $M_1$  where  $P_2$  is obtained by formula (4). Now suppose  $p_1(i, z) = 0$  or  $p_1(z, j) = 0$  for  $i, j \in S_2$ , then  $p_2(i, j) = p_1(i, j)$  in model  $M_2$ , i.e. elimination of state  $z$  does not change the probabilities between states  $i$  and  $j$  in  $M_2$ .

*Proof.* The proof follows from formula (4) applied to  $x = i, y = j$  and  $z = z$ .  $\square$

**Remark 6.** Suppose  $D \subset S_1$ , and one of  $p_1(i, z) = 0$  or  $p_1(z, j) = 0$  for all  $z \in D$ , then repeated application of Lemma 5 for states  $z \in D$  results in  $p_D(i, j) = p_1(i, j)$  for such  $j \in S_1 \setminus D$  in model  $M_D$ . That is, set  $D$  can also be eliminated without affecting the probabilities for state  $i \in S_2$  and some states  $j \in S_2$ .

Lemma 5 and Remark 6 will play an important role in stages 1 and 3 of the IP algorithm in the next section.

## 6 Main results: The IP algorithm

The IP algorithm consists of *three main stages*. The *third stage* involves two steps. Without going into detail, we first outline *the three stages of the IP algorithm*:

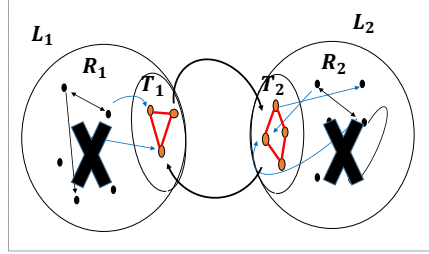
(*Stage 1*) In the first stage, we apply the *SR* approach to eliminate the interior states  $R_i$  from island  $L_i$ . By Lemma 5 and Remark 6, this can be done in parallel for each island  $L_i$ ,  $1 \leq i \leq k$ . The elimination of interior states  $R_i, 1 \leq i \leq k$ , results in the model  $M^* = (T, P^*)$  for ports  $T$ ; see Fig.2a.

(*Stage 2*) In the second stage, we calculate the invariant distribution  $\pi^*$  for model  $M^*$ . Any direct method can be used. We use the GTH/S algorithm to calculate  $\pi^*$  in our numerical example; see Fig.2b.

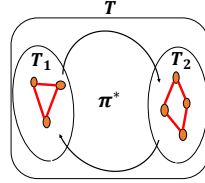
(*Stage 3. step 1.*) In the third stage, we consider models  $M_i = (L_i^*, P_i^*)$  for *augmented islands*  $L_i^* = R_i \cup T, 1 \leq i \leq k$ . Using the distribution  $\pi^*$  obtained in the second stage, we calculate the invariant distribution  $\pi_{(i)}$  for each model  $M_i$  separately, and in parallel. Note that the notation  $\pi_{(j)}$  only refers to the invariant distribution of model  $M_j$  and not that of individual states in  $M_j$ . For example,  $\pi_{(j)}(x)$  refers to the invariant distribution of state  $x \in L_j^*$ ; see Fig.2c.

For each model  $M_i$ , we calculate the coefficients  $w_i$  and  $\lambda_i$ , which are given by

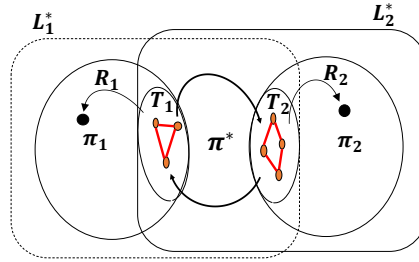
$$w_i = \pi_{(i)}(R_i) \quad \text{and} \quad \lambda_i = \frac{w_i}{1 - w_i}, \quad i = 1, 2, \dots, k. \quad (20)$$



(a) *Stage 1.* Elimination of  $R_1$  and  $R_2$



(b) *Stage 2.* Model  $M^*$  with invariant distribution  $\pi^*$



(c) *Stage 3. step 1.* Calculation of  $\pi_1$  and  $\pi_2$  for models  $M_1$  and  $M_2$ , respectively

Figure 2: Stages 1, 2 and 3 (step 1) for the case  $k = 2$

(*Stage 3. step 2. (Extension step)*) Finally, we use the distributions  $\pi_{(i)}, 1 \leq i \leq k$ , to calculate the distribution  $\pi$  for model  $M$ . According to Remark 4, the invariant distribution of  $L_i$  inside model  $M_i, 1 \leq i \leq k$ , is proportional to its distribution w.r.t. model  $M$ , that is,

$$\pi(L_i) = e_i \pi_{(i)}(L_i), \quad i = 1, 2, \dots, k, \quad (21)$$

where  $e_i = \pi(L_i^*)$ . If the coefficients  $e_i$  were known, we could directly use formula (21) to calculate distribution  $\pi(L_i)$  in each model  $M_i, 1 \leq i \leq k$ , but they are not known. Our main result, Theorem 6, gives an explicit formula (23) to calculate these unknown coefficients  $e_i, 1 \leq i \leq k$ .

**Theorem 6.** *The invariant distribution of state  $y \in L_i^*$  is given by the formula*

$$\pi(y) = e_i \pi_{(i)}(y), \quad i = 1, 2, \dots, k, \quad (22)$$

where coefficients  $e_i$  are given by

$$e_i = \frac{1 + \lambda_i}{1 + \sum_j \lambda_j}, \quad i = 1, 2, \dots, k. \quad (23)$$

It is easy to check that  $\pi$  is a probability distribution. We give a proof of Theorem 1 at the end of this section.

**Remark 7.** *Since  $T = \bigcup_{i=1}^k T_i$ , formula (22) Theorem 6 implies that any one of the islands could be used to find the distribution on  $T$  because  $e_1 \pi_{(1)}(x) = e_2 \pi_{(2)}(x) = \dots = e_k \pi_{(k)}(x)$  for all  $x \in T$ . These equalities follow from (21) written for  $L_i^*$ .*

We now discuss each stages of the IP algorithm in greater detail.

#### Stage 1

Consider an IP model  $M = (S, P)$  with the state space  $S = \sum_{i=1}^k L_i$  and transition matrix  $P$ . Let us first represent the stochastic matrix,  $P$ , as a union of blocks by *ignoring the blocks that are always zero*. Here, by a union of blocks we mean a collection of submatrices of an actual matrix. We will use such a representation for our matrices to make our discussion easy to follow; see Fig.3a,

$$P = \left( \bigcup_{i=1}^k P_i \right) \cup \left( \bigcup_{\substack{i,j=1 \\ i \neq j}}^k T_{ij} \right), \quad P_i = \begin{bmatrix} P_{i_0 i_0} & P_{i_0 i} \\ P_{i i_0} & P_{i i} \end{bmatrix}, \quad P^* = \left( \bigcup_{i=1}^k P_{ii}^* \right) \cup \left( \bigcup_{\substack{i,j=1 \\ i \neq j}}^k T_{ij} \right), \quad (24)$$

where block  $P_i := \{p(x, y), x, y \in L_i\}$  and block  $T_{ij} := \{p(x, y), x \in T_i, y \in T_j, i \neq j\}$ .

As in (1), we decompose each block  $P_i$  as the second matrix in (24), where submatrices

$P_{i_0 i_0} := \{p(x, y), x, y \in R_i\}$ ,  $P_{i_0 i} := \{p(x, y), x \in R_i, y \in T_i\}$ ,  $P_{i i} := \{p(x, y), x, y \in T_i\}$  and

$P_{i i_0} := \{p(x, y), x \in T_i, y \in R_i\}$ .

Now let  $R = \sum_{i=1}^k R_i$  be the collection of all interior states,  $T = \sum_{i=1}^k T_i$  be the collection of all ports of model  $M$ , and  $S = R + T$ . In stage 1 we eliminate  $R$  from  $S$  to obtain a reduced model  $M^* = (T, P^*)$  for the ports  $T$ . The matrix  $P^*$  can also be represented as the third matrix in (24) where  $P_{ii}^* := \{p^*(x, y), x, y \in T_i\}$  is the block with new probabilities for the ports  $T_i$  calculated using formula (2) from block  $P_i$  after set  $R_i$  is eliminated, and is given by

$$P_{ii}^* = P_{ii} + P_{i i_0} N_{i_0} P_{i_0 i}, \quad i = 1, 2, \dots, k. \quad (25)$$

We have used the notation ‘\*’ in block  $P_{ii}$  of matrix  $P^*$  in (24) to indicate that the elimination of  $R = \bigcup_{i=1}^k R_i$  affects only blocks  $P_{ii}$  but blocks  $T_{ij}$  are not affected for  $i \neq j, 1 \leq i, j \leq k$ . According to Lemma 7, blocks  $P_{ii}^*, 1 \leq i \leq k$ , in matrix  $P^*$  can be calculated in parallel.

**Lemma 7.** *The blocks  $P_{ii}^*$  for each island  $L_i, 1 \leq i \leq k$ , can be calculated in parallel.*

*Proof.* First, recall that we mentioned earlier that the IP model is ergodic. This implies that there are states  $x \in R_i$  and  $y \in T_i$  such that  $p(x, y) > 0$  in each island  $L_i, 1 \leq i \leq k$ . Then according to Lemma 1 from Chapter 2, eliminating  $R$  from  $S$  affects probabilities of the remaining states in the ports  $T, T_i \subset L_i, 1 \leq i \leq k$ . Since  $L_i \cap L_j = \emptyset$  and  $p(x, y) = 0$

for all  $x \in L_i$  and  $y \in R_j$ ,  $i \neq j$ ,  $1 \leq j \leq k$ , Lemma 5 and Remark 6 from section 5 implies that the elimination of interior states  $R_j$  from islands  $L_j$ ,  $j \neq i$ ,  $1 \leq j \leq k$ , has no effect on the probabilities for states in island  $L_i$ . As a result, only the elimination of set  $R_i$  affects the block  $P_{ii}$  in  $P_i$  in (24). Then the elimination of  $R$  can be done by eliminating  $R_i$  from the block  $P_i$  for island  $L_i$  separately and in parallel. This is illustrated in Fig. 3b.  $\square$

However, the blocks  $T_{ij}$  remain unchanged because interior states in each island  $L_i$  do not communicate with ports  $T_j$  from other islands, i.e.  $p(i, j) = 0$  for  $i \in R_i$ ,  $j \in T_j$ ,  $j \neq i$ ,  $1 \leq j \leq k$ . By aggregating  $P_{ii}^*$  and  $T_{ij}$  for  $i \neq j$ ,  $1 \leq i, j \leq k$ , we obtain the matrix  $P^*$ ; see Fig.3c for the case when  $k = 2$  case.

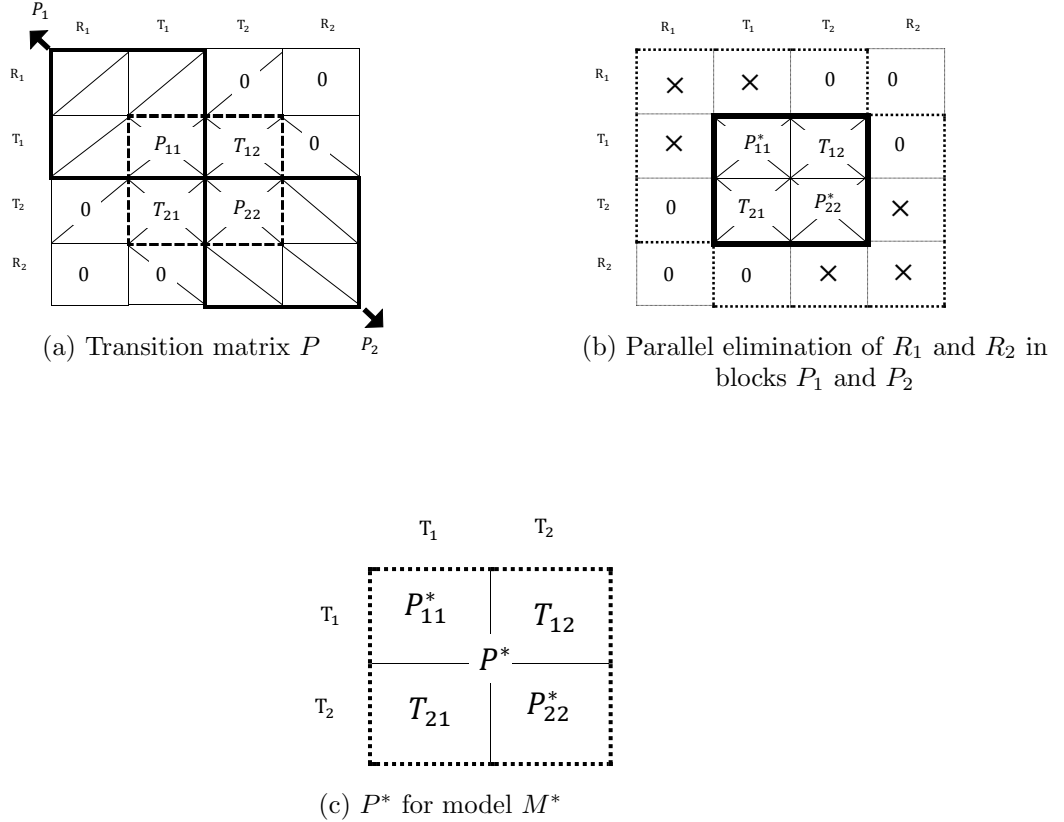


Figure 3: State reduction and calculation of  $P^*$  for the case  $k = 2$

## Stage 2

Let us consider a model  $M^* = (T, P^*)$  for the ports  $T$ . The transition matrix  $P^*$  was obtained at the end of stage 1. In stage 2, we calculate the invariant distribution  $\pi^*$  for model  $M^*$ . For this calculation, we can use the GTH/S algorithm or any other direct method. Applying the GTH/S algorithm (as we do in our numerical example) has the complexity of order  $\frac{2}{3}m^3$ , where  $|T| = |L_i| = m, 1 \leq i \leq k$ .

## Stage 3

In this stage, we first consider “augmented islands”  $L_i^*, 1 \leq i \leq k$ , which are obtained by enlarging each island,  $L_i$ , to include all ports, i.e.  $L_i^* = R_i \cup T$ . Let us define new models  $M_i = (L_i^*, P_i^*), 1 \leq i \leq k$ , with state space  $L_i^*$  and transition matrix  $P_i^*$ . Although the state space  $L_i^*$  of each model  $M_i, 1 \leq i \leq k$ , includes  $T$ , i.e.  $T = \bigcap_{i=1}^k L_i^*$ , each  $M_i$  is a separate model with its own transition matrix  $P_i^*$ . The matrix  $P_i^*$  for model  $M_i^*$  can be calculated from the initial matrix  $P$  by eliminating all states outside of  $L_i^*$ , that is, by eliminating  $\bigcup_{j=1}^k R_j, j \neq i$  from  $S$ . However, let us show that the matrix  $P_i^*$  can be obtained from stage 1 without any additional calculations. Indeed, by ignoring the parts that are always zero, the matrix  $P_i^*$  can also be represented as

$$P_i^* = P_i \cup \left( \bigcup_{\substack{j=1 \\ j \neq i}}^k P_{jj}^* \right) \cup \left( \bigcup_{\substack{i,j=1 \\ i \neq j}}^k T_{ij} \right), \quad i = 1, 2, \dots, k. \quad (26)$$

Blocks  $P_{jj}^*, 1 \leq j \leq k, j \neq i$ , were calculated in stage 1 and blocks  $P_i$  and  $T_{ij}$  can be obtained directly from the original matrix  $P$ . Figures (4a) and (4b) show how matrices  $P_1^*$  and  $P_2^*$  are obtained for case  $k = 2$ .

The block form in (26) of matrix  $P_i^*$  also follows directly from Lemma 5 and Remark 6 which implies that elimination of the interior states  $R_j$  from matrix  $P$  in (24) only affects



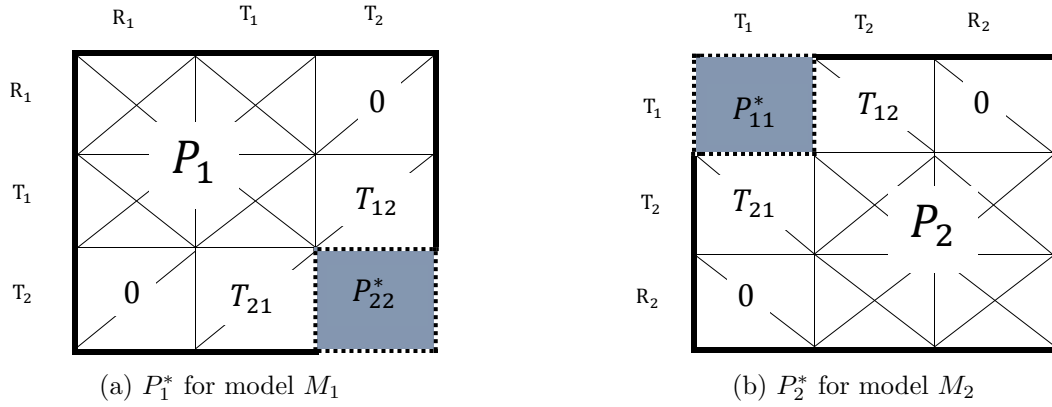


Figure 4: Transition matrices  $P_1$  and  $P_2$  for the case  $k = 2$

the block  $P_{jj}$  inside the block  $P_j$  of each island  $L_j, 1 \leq j \leq k$ . As we discussed in stage 1, we indicate this change by updating block  $P_{jj}$  to block  $P_{jj}^*$ .

According to Remark 5 in Sect.5, the invariant distribution,  $\pi$ , in (16) can be calculated from the distribution  $\pi_s$  of an intermediate stochastic matrix  $P_s, 1 \leq s \leq n$ . From the discussion in Chapter 1, the matrix  $P_s$  is the transition matrix of a reduced model  $M_s$  after eliminating states  $\{1, 2, 3, \dots, s-1\}$  from the initial model  $M_1 = (S_1, P_1)$ . This fact leads to Lemma 8.

**Lemma 8.** *The invariant distribution,  $\pi_{(i)}$ , for each model  $M_i$  can be calculated in parallel,  $1 \leq i \leq k$ .*

*Proof.* For each  $i \in \{1, 2, \dots, k\}$ , let  $M_i = (L_i^*, P_i^*)$  be the initial model. If we eliminate just the interior states  $R_i \subset L_i^* = L_i + T$  from  $M_i$ , we obtain a reduced model for the ports  $T$  only. The model we obtain is the model  $M^*$ , which was introduced in stage 2. In other words, the model  $M^*$  is a  $R_i$ -reduced model of model  $M_i$ . Because  $T = \bigcap_{i=1}^k L_i^*$ , the model  $M^*$  is also a  $R_i$ -reduced model of every model  $M_i, 1 \leq i \leq k$ . Given that the model  $M^*$  is a common model for each separate model  $M_i, 1 \leq i \leq k$ , we can start with the distribution

$\pi^*$  and use Lemma 4 to recursively compute  $\pi_{(i)}$  for each model  $M_i$  in parallel.  $\square$

This concludes the discussion of the three stages of the IP algorithm. In Fig.5a, we summarize stages 1 and 2 of the IP algorithm; in Fig.5b we illustrate stage 3 (step 1) of the algorithm.

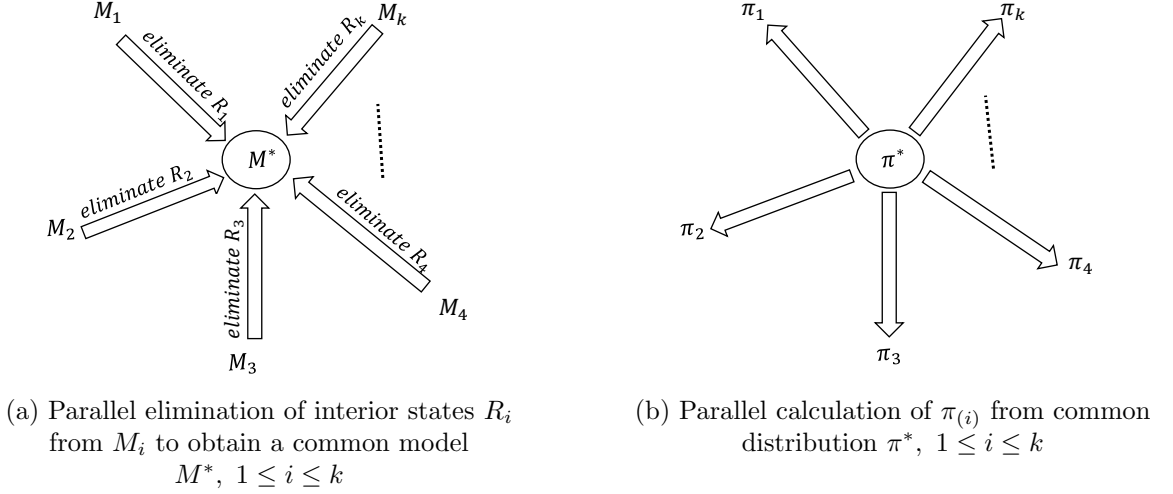


Figure 5: Parallel features of the IP algorithm

In the extension (final) step, we now use distributions  $\pi_{(i)}$ ,  $1 \leq i \leq k$ , calculated in stage 3 to compute the actual distribution  $\pi$  for model  $M$  using formulas (22) and (23) from Theorem 6.

*Proof of Theorem 6:* Let us first introduce the coefficients  $r_i = \pi(R_i)$  and  $t = \pi(T)$ . Notice that proportionality in (21) also holds for the interior states  $R_i \subset L_i^*$ ,

$$\pi(R_i) = e_i \pi_{(i)}(R_i) \quad \text{i.e.} \quad r_i = e_i w_i, \quad i = 1, 2, \dots, k. \quad (27)$$

Since  $L_i^* = R_i \cup T$  and  $S = \cup_{i=1}^k R_i \cup T$ , we have the equalities

$$r_i = e_i - t, \quad (28)$$

$$\sum_{i=1}^k r_i + t = 1. \quad (29)$$

We substitute (28) into (29) to get  $\sum_i(e_i - t) + t = \sum_i e_i - (k - 1)t = 1$ . Hence,

$$t = \frac{\sum_i e_i - 1}{k - 1}. \quad (30)$$

We substitute (30) for  $t$  in (28) to obtain  $r_i = e_i - (\sum_i e_i - 1)/(k - 1)$ . By the second equality in (27)

$$\begin{aligned} e_i - \frac{\sum_i e_i - 1}{k - 1} &= e_i w_i, \\ e_i(1 - w_i)(k - 1) &= \sum_i e_i - 1. \end{aligned} \quad (31)$$

Let  $e = \sum_i e_i$  and  $\lambda_i = w_i/(1 - w_i)$ , or *equivalently*,  $1 - w_i = 1/(1 + \lambda_i)$ . Thus,  $\lambda_i$  gives the ratio of the probability mass of interior states  $R_i$  to that of ports  $T$  in each model  $M_i$ ,  $1 \leq i \leq k$ . Equation (31) can be simplified to

$$e_i = \frac{(e - 1)(1 + \lambda_i)}{k - 1}. \quad (32)$$

If we sum (32) over all  $i$ ,  $1 \leq i \leq k$ , we obtain

$$e = \frac{e - 1}{k - 1} \sum_i (1 + \lambda_i) = (e - 1)\lambda, \quad (33)$$

where we let  $\lambda = \sum_i (1 + \lambda_i)/(k - 1)$ . Equation (33) can be rearranged to give  $e - 1 = 1/(\lambda - 1)$ . Then, we can simplify (32) as

$$e_i = \frac{(1 + \lambda_i)}{(e - 1)(k - 1)}. \quad (34)$$

We use the expression for  $\lambda$  to obtain

$$(k - 1)(\lambda - 1) = (k - 1) \left( \frac{\sum_i (1 + \lambda_i)}{k - 1} - 1 \right) = (k - 1) \left( \frac{k + \sum_i \lambda_i}{k - 1} - 1 \right) = 1 + \sum_i \lambda_i. \quad (35)$$

Substitution of the expression (35) in the equality (34) gives

$$e_i = \frac{1 + \lambda_i}{(k-1)(\lambda-1)} = \frac{1 + \lambda_i}{1 + \sum_i \lambda_i}, \quad (36)$$

with  $\lambda_i = w_i/(1 - w_i)$ . The expression in (22) follows by applying formula (21) to the augmented island  $L_i^*$ .

□

We include a pseudo-code (2) in the Appendix.

## 7 Performance Evaluation

Let us assume that there are  $n$  states,  $k$  islands, and  $n/k$  states in each island. If we let  $m$  be the number of ports, then there are  $(n/k - m)$  interior states in each island.

*(Stage 1)* In stage 1, we eliminate  $(n/k - m)$  states from each island  $L_i, 1 \leq i \leq k$ . For each  $L_i$ , calculation of sequences of matrices  $W_{R_i}^{(i)}$  requires approximately  $(n/k - m)((n/k)^2 + 1)$  additions and subtractions,  $(n/k - m)((n/k)^2 + 1)$  divisions and multiplications; for a total of  $k$  islands, it requires approximately  $2k(n/k - m)((n/k)^2 + 1)$  operations.

*(Stage 2)* For  $T = km$  ports, the GTH/S algorithm requires  $2/3(km)^3 + 2(km)^2 - 8/3$  operations while computing  $\pi$  for model  $M^*$ .

*(Stage 3)* (i) The calculation of matrix  $\hat{W}^{(i)}$  when restoring the interior states  $R_i$  for each model  $M_i, 1 \leq i \leq k$ , requires approximately the same number of operations as in the calculation of matrix  $W_{R_i}^{(i)}$  in stage 1 (ii) Each step in the calculation of  $\pi_{(i)}$  using formulas (18),(19) involves approximately  $(n/k - m)$  additions and  $(n/k - m)$  multiplications, 1 subtraction and 2 divisions for each model. This results in a total count of  $2k(n/k - m)^2 + 3(n/k - m)$  after  $n/k - m$  states are restored. (iii) The

calculation of  $w_i$  in (20) involves performing  $n/k - m$  additions, calculation of  $\lambda_i$  requires 1 division and 1 subtraction for each island. (iv) Finally, the calculation of  $e_i, 1 \leq i \leq k$ , in (23) requires  $3k + 2$  additions and  $k$  divisions; calculation of  $\pi$  involves  $n$  multiplications.

We assume that the total number of ports  $t \simeq n/k$  (size of the biggest island). If we suppose that there are an equal number of ports  $m$  in each  $L_i$ , i.e.  $m = n/k^2$ , then an approximate total count for the IP algorithm is given by

$$TC_{IP} \simeq \left(4k + \frac{2}{3}\right) \left(\frac{n}{k}\right)^3.$$

Thus, this algorithm has a time complexity of order  $n^3/k^2$ .

## 8 Numerical Example

We consider a Markov model with the state space  $S = \{1, 2, \dots, 12\}$  and consisting of three islands:  $L_1 = \{T_1 = (1, 2)\} \cup \{R_1 = (3, 4)\}$ ,  $L_2 = \{T_2 = (5, 6)\} \cup \{R_2 = (7, 8)\}$  and  $L_3 = \{T_3 = (9, 10)\} \cup \{R_3 = (11, 12)\}$ . Further we assume that based on this partition of  $S$ , the transition matrix  $P$  can be rearranged as in Table 1. All calculations were performed in MATLAB with a tolerance of  $10^{-7}$ .

(Stage 1) We eliminate the interior states  $R_1 = \{3, 4\}$  from  $L_1$ , states  $R_2 = \{7, 8\}$  from  $L_2$  and states  $R_3 = \{11, 12\}$  from  $L_3$  in parallel. After this elimination, the states that remain in  $L_1, L_2$ , and  $L_3$  are ports  $T_1 = \{1, 2\}, T_2 = \{5, 6\}$  and  $T_3 = \{9, 10\}$ , respectively. This elimination changes blocks  $P_{11}$  to  $P_{11}^*$ ,  $P_{22}$  to  $P_{22}^*$  and  $P_{33}$  to  $P_{33}^*$  but blocks  $T_{ij}, i \neq j, i, j = 1, 2, 3$ , are unchanged. By aggregating these blocks, we obtain the stochastic matrix  $P^*$ ; See Table 2.

$$\left(\frac{1}{100}\right) \times$$

	$R_1$		$T$						$R_2$		$R_3$	
			$T_1$		$T_2$		$T_3$					
	(3)	(4)	(1)	(2)	(5)	(6)	(9)	(10)	(7)	(8)	(11)	(12)
(3)	10	9	48	33	0	0	0	0	0	0	0	0
(4)	20	16	11	53	0	0	0	0	0	0	0	0
(1)	20	17	7	10	9	5	20	12	0	0	0	0
(2)	11	20	13	2	7	15	11	21	0	0	0	0
(5)	0	0	20	4	7	8	23	10	18	10	0	0
(6)	0	0	3	10	20	14	7	13	16	17	0	0
(9)	0	0	33	10	10	30	2	5	0	0	10	0
(10)	0	0	7	13	25	15	5	13	0	0	10	12
(7)	0	0	0	0	30	20	0	0	3	47	0	0
(8)	0	0	0	0	43	10	0	0	26	21	0	0
(11)	0	0	0	0	0	0	30	10	0	0	40	20
(12)	0	0	0	0	0	0	10	40	0	0	30	20

Table 1: Transition matrix  $P$ 

(Stage 2) The invariant distribution,  $\pi^*$ , for model  $M^* = (T, P^*)$  is given by

$$\pi^* = \begin{bmatrix} (1) & (2) & (5) & (6) & (9) & (10) \\ \frac{438}{2393}, & \frac{183}{1237}, & \frac{317}{1522}, & \frac{544}{3209}, & \frac{359}{2442}, & \frac{440}{3051} \end{bmatrix}.$$

Here, we used the GTH/S algorithm for this calculation.

(Stage 3) We consider three augmented islands:  $L_1^* = \{1, 2, 3, 4, 5, 6, 9, 10\}$ ,  $L_2^* = \{1, 2, 5, 6, 7, 8, 9, 10\}$ , and  $L_3^* = \{1, 2, 5, 6, 9, 10, 11, 12\}$ . For these, we consider three Markov models  $M_1 = (L_1^*, P_1^*)$ ,  $M_2 = (L_2^*, P_2^*)$ , and  $M_3 = (L_3^*, P_3^*)$ . Matrices  $P_1^*$ ,  $P_2^*$  and  $P_3^*$  are given in Table 3 for models  $M_1$ ,  $M_2$  and  $M_3$ , respectively.

Using formulas (18) and (19) in Lemma 4, we compute the invariant distribution  $\pi_{(1)}$  for model  $M_1$

$$\pi_{(1)} = \begin{bmatrix} (1) & (2) & (3) & (4) \\ \frac{320}{2023}, & \frac{213}{1666}, & \frac{387}{5843}, & \frac{119}{1711} \end{bmatrix}.$$

The distributions  $\pi_{(2)}$  and  $\pi_{(3)}$  for models  $M_1$  and  $M_2$  are calculated in a similar way, which

	$R_1$		$T$						$R_2$		$R_3$	
	(3)	(4)	(1)	(2)	(5)	(6)	(9)	(10)	(7)	(8)	(11)	(12)
(3)	0	0	0	0	0	0	0	0	0	0	0	0
(4)	0	0	0	0	0	0	0	0	0	0	0	0
(1)	0	0	640/2821	2578/8233	9/100	5/100	20/100	12/100	0	0	0	0
(2)	0	0	843/3449	725/3363	7/100	15/100	11/100	21/100	0	0	0	0
(5)	0	0	20/100	4/100	747/2771	471/2936	23/100	10/100	0	0	0	0
(6)	0	0	3/100	10/100	281/639	341/1481	7/100	13/100	0	0	0	0
(9)	0	0	33/100	10/100	10/100	30/100	43/525	37/420	0	0	0	0
(10)	0	0	7/100	13/100	25/100	15/100	13/84	103/420	0	0	0	0
(7)	0	0	0	0	0	0	0	0	0	0	0	0
(8)	0	0	0	0	0	0	0	0	0	0	0	0
(11)	0	0	0	0	0	0	0	0	0	0	0	0
(12)	0	0	0	0	0	0	0	0	0	0	0	0

Table 2: Transition matrix  $P^*$ 

gives

$$\pi_{(2)} = \left[ \begin{array}{cccc} (5) & (6) & (7) & (8) \\ \frac{300}{1759}, & \frac{265}{1909}, & \frac{431}{5301}, & \frac{119}{1192} \end{array} \right], \quad \pi_{(3)} = \left[ \begin{array}{cccc} (9) & (10) & (11) & (12) \\ \frac{306}{2303}, & \frac{186}{1427}, & \frac{254}{4143}, & \frac{79}{2265} \end{array} \right].$$

Using formula (20), we calculate the coefficients  $w = (w_1, w_2, w_3)$  and  $\lambda = (\lambda_1, \lambda_2, \lambda_3)$  for models  $M_1, M_2$  and  $M_3$ , which are given by

$$w = \left( \frac{691}{5089}, \frac{121}{668}, \frac{618}{6425} \right), \quad \lambda = \left( \frac{691}{4398}, \frac{121}{547}, \frac{618}{5807} \right).$$

We use formula (23) in Theorem 6 to calculate the coefficients  $e = (e_1, e_2, e_3)$

$$e = \left( \frac{611}{784}, \frac{1177}{1431}, \frac{3372}{4525} \right).$$

We can verify that the above coefficients are correct by calculating  $e_i = \pi(R_i)/\pi_{(i)}(R_i)$ .

Finally, we calculate  $\pi$  using formula (22),

$$\pi = \left[ \begin{array}{cccccccccccc} (1) & (2) & (3) & (4) & (5) & (6) & (7) & (8) & (9) & (10) & (11) & (12) \\ \frac{438}{3553}, & \frac{138}{1385}, & \frac{67}{1298}, & \frac{325}{5996}, & \frac{171}{1219}, & \frac{269}{2356}, & \frac{495}{7402}, & \frac{325}{3958}, & \frac{241}{2434}, & \frac{281}{2893}, & \frac{197}{4312}, & \frac{137}{5271} \end{array} \right].$$

	$R_1$		$T$					
	(3)	(4)	(1)	(2)	(5)	(6)	(9)	(10)
(3)	10/100	9/100	48/100	33/100	0	0	0	0
(4)	20/100	16/100	11/100	53/100	0	0	0	0
(1)	20/100	17/100	7/100	10/100	9/100	5/100	20/100	12/100
(2)	11/100	20/100	13/100	2/100	7/100	15/100	11/100	21/100
(5)	0	0	20/100	4/100	747/2771	471/2936	23/100	10/100
(6)	0	0	3/100	10/100	281/639	341/1481	7/100	13/100
(9)	0	0	33/100	10/100	10/100	30/100	43/525	37/420
(10)	0	0	7/100	13/100	25/100	15/100	13/84	103/420

(a) Transition matrix  $P_1^*$ 

	$T$						$R_2$	
	(1)	(2)	(5)	(6)	(7)	(8)	(9)	(10)
(1)	640/2821	2578/8233	9/100	5/100	0	0	20/100	12/100
(2)	843/3449	725/3363	7/100	15/100	0	0	11/100	21/100
(5)	20/100	4/100	7/100	8/100	18/100	10/100	23/100	10/100
(6)	3/100	10/100	20/100	14/100	16/100	17/100	7/100	13/100
(7)	0	0	30/100	20/100	3/100	47/100	0	0
(8)	0	0	43/100	10/100	26/100	21/100	0	0
(9)	33/100	10/100	10/100	30/100	0	0	43/525	37/420
(10)	7/100	13/100	25/100	15/100	0	0	13/84	103/420

(b) Transition matrix  $P_2^*$ 

	$T$						$R_3$	
	(1)	(2)	(5)	(6)	(9)	(10)	(11)	(12)
(1)	640/2821	2578/8233	9/100	5/100	20/100	12/100	0	0
(2)	843/3449	725/3363	7/100	15/100	11/100	21/100	0	0
(5)	20/100	4/100	747/2771	471/2936	23/100	10/100	0	0
(6)	3/100	10/100	281/639	341/1481	7/100	13/100	0	0
(9)	33/100	10/100	10/100	30/100	2/100	5/100	10/100	0
(10)	7/100	13/100	25/100	15/100	5/100	13/100	10/100	12/100
(11)	0	0	0	0	30/100	10/100	40/100	20/100
(12)	0	0	0	0	10/100	40/100	30/100	20/100

(c) Transition matrix  $P_3^*$ Table 3: Transition matrices for models  $M_1$ ,  $M_2$  and  $M_3$



## 9 Application: approximations of nearly uncoupled MCs; Perturbation estimates

As we mentioned in Chapter 1, the transitions between states in the IP models are more restrictive than in the nearly uncoupled (ergodic) Markov models. Most of the applications discussed in Chapter 1 are examples of the nearly uncoupled MCs. In this section, we will discuss the possibilities of considering the IP model as an approximation of a nearly uncoupled Markov model. We will show that such an approximation can be understood as a *perturbation* of a stochastic matrix.

Assume that  $M_1 = (S_1, P_1)$  is a nearly uncoupled model. Let  $S_1 = \sum_{i=1}^k L_i$ ,  $|S_1| = n$  and  $L_i = R_i + T_i$ . Suppose that most of the transitions between the clusters  $L_i$  and  $L_j$ ,  $i \neq j$ , occur only through some states in  $T_i$ , but assume that there are also small transitions between some states  $i \in R_i$  and  $j \in R_j$ ,  $i \neq j$ ,  $1 \leq i, j \leq k$ , i.e.  $0 < p_1(i, j) \leq \epsilon$  for some  $\epsilon$  close to zero. In this simple case, we can obtain another model  $M_2 = (S_2, P_2)$  from model  $M_1$  by redistributing these small probabilities  $p_1(i, j)$ ,  $i \neq j$ ,  $i \in R_i$ ,  $j \in R_j$ , back to each island  $L_i$ ,  $1 \leq i \leq k$ . Then,  $M_2$  can be thought of as a new model resulting after some *perturbation* is applied to the original model  $M_1$ . Most importantly,  $M_2$  is now an IP model. Before we discuss the perturbation theory of stochastic matrices and present some known results, we investigate the following three ways to redistribute small probabilities in  $M_1$ :

(a) Let  $\epsilon_i = \sum_j p_1(i, j)$ . Then we can divert probabilities  $p_1(i, j)$ ,  $i \in R_i$ ,  $j \in \{R_1, R_2, \dots, R_j\}$ ,  $i \neq j$ , back to the state of origin, i.e.  $p_2(i, i) = p_1(i, i) + \epsilon_i$ . All other probabilities remain the same but now there are no small transitions between the islands.

(b) Let  $|T_i| = t_i$  be the number of ports in the island  $L_i$ . We can also distribute  $\epsilon_i$  evenly to the ports  $T_i$ , i.e.  $p_2(i, j) = p_1(i, j) + \epsilon_i/t_i$ ,  $j \in T_i$ .

(c) Similar to (b) but we can also redistribute  $\epsilon_i$  evenly to all states in  $L_i, 1 \leq i \leq k$ .

Let  $\pi_1$  and  $\pi_2$  be the invariant distributions for models  $M_1$  and  $M_2$  respectively. Let us define the total relative error as

$$\text{Total relative error} = \sum_{i=1}^n \left| \frac{\pi_1(i) - \pi_2(i)}{\pi_1(i)} \right| \times 100 \quad (37)$$

We performed a simple simulation for an example with 2 clusters to examine if (a), (b) or (c) is the best method for the redistribution of small probabilities. Note that there many other ways to do so, in addition to the three ways we propose. Our example consists of clusters  $L_1 = \{1, 2, 3, 4\}$  and  $L_2 = \{5, 6, 7, 8, 9\}$  with most interactions occurring through states  $T_1 = \{3, 4\}$  and  $T_2 = \{5, 6\}$  respectively. All calculations were performed using the GTH/S algorithm.

**Simulation 1.** Consider states  $1 \rightarrow 7$  in the model  $M_1$  in a randomly generated stochastic matrix  $P_1$ <sup>3</sup>. Let  $p_1(1, 7) = \epsilon$  where  $\epsilon \in \text{uniform}[0, \epsilon_{max}]$ . All other transitions between  $L_1$  and  $L_2$  occur through  $T_1$  and  $T_2$ . For the model  $M_2$ , we tested method (a) where we divert  $\epsilon$  back to  $p_1(1, 1)$ , i.e.  $p_2(1, 1) = p_1(1, 1) + \epsilon$ . We calculate  $\pi_2$  and compute the total relative error; then we tested method (b) where we let  $p_2(1, j) = p_1(1, j) + \epsilon/2, j \in T_1$ ; finally in method (c),  $p_2(i, j) = p_1(i, j) + \epsilon/4$ , if  $i, j \in L_1$ . By varying  $\epsilon_{max}$ , we plotted the total relative errors for 20 simulations<sup>4</sup>.

For the case of one small transition, as we increase  $\epsilon_{max}$  in Fig.6a–6d, we see that the total relative error increases, which is expected, but method (b) gives the lowest total relative error.

---

<sup>3</sup>For each simulation we generate a new matrix  $P_1$

<sup>4</sup>Pattern is similar for  $N = 100$  or larger as well. We chose  $N = 20$  so that it is easy to see the pattern.

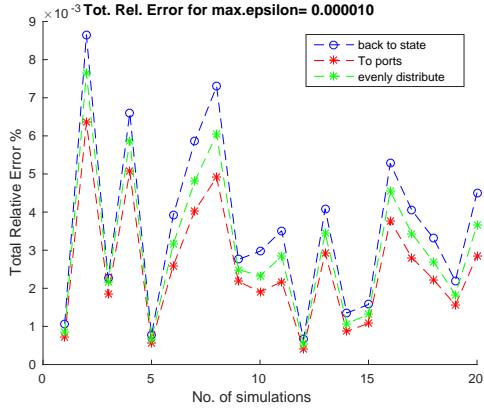
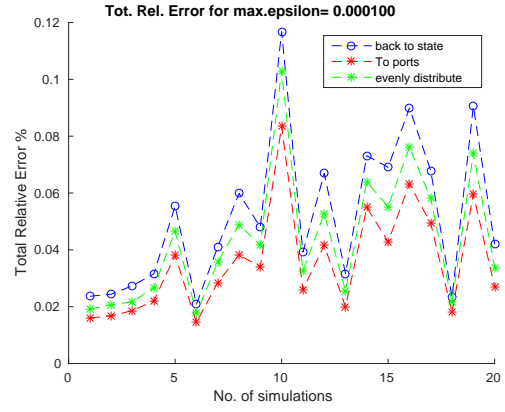
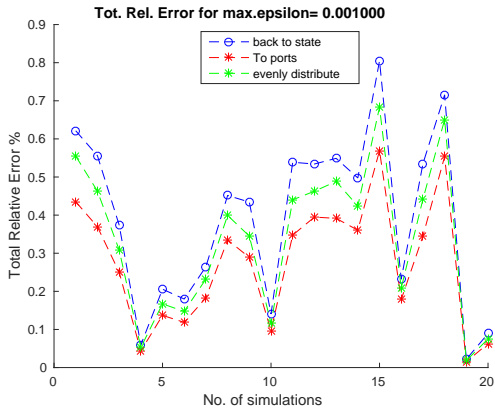
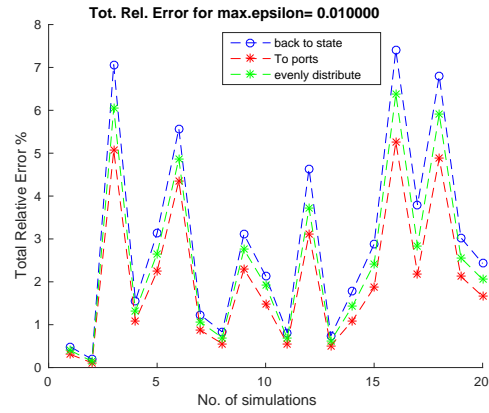
(a)  $\epsilon = \text{unif}[0, 0.00001]$ (b)  $\epsilon = \text{unif}[0, 0.0001]$ (c)  $\epsilon = \text{unif}[0, 0.001]$ (d)  $\epsilon = \text{unif}[0, 0.01]$ 

Figure 6: An example of redistribution of small probabilities (1 state case)

**Simulation 2.** We now consider the case when  $p_1(1, 7) = \epsilon_1, p_1(8, 2) = \epsilon_2$ , where  $\epsilon_1, \epsilon_2 \in \text{unif}[0, \epsilon_{max}]$ . We calculate the total relative error using method (a) where  $p_2(1, 1) = p_1(1, 1) + \epsilon_1, p_2(8, 8) = p_1(8, 8) + \epsilon_2$ ; and we repeat with (b) where  $p_2(1, j) = p_1(1, j) + \epsilon_1/2, j \in T_1$  and  $p_2(8, j) = p_1(8, j) + \epsilon_2/2, j \in T_2$ ; in method (c)  $p_2(i, j) = p_1(i, j) + \epsilon_1/4$ , if  $i, j \in L_1$  and  $p_2(i, j) = p_1(i, j) + \epsilon_2/5$ , if  $i, j \in L_2$ .

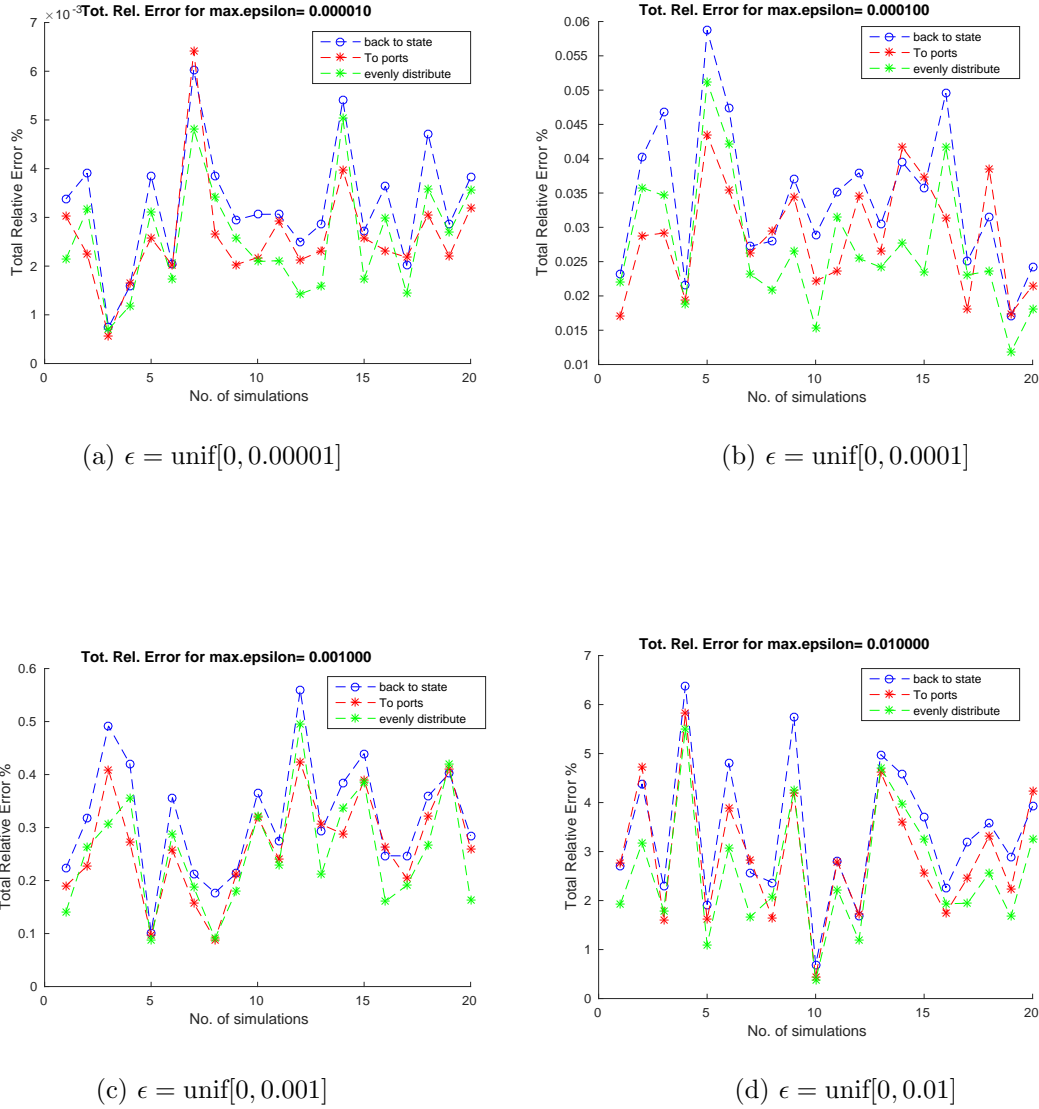


Figure 7: An example of redistribution of small probabilities (2 states case)

In Fig. 7a we can see that the total relative error is small when  $\epsilon_1$  and  $\epsilon_2$  are small. For the case with 2 states, we can see that redistributing small transitions evenly among all states in each island consistently gives the lowest relative error. Of course, this must be tested for large state spaces with more states participating in such transitions. From Figures 6 and 7, we can see that the total relative error is about 6% if  $\epsilon_1, \epsilon_2 \in \text{unif}[0, 0.01]$ . For small transitions, the IP model may not be a bad approximation of the original model. We may

choose to use method (b) or (c) for the redistribution of small probabilities.

### Perturbation estimates

In this section, we give a brief survey of the perturbation theory of stochastic matrices. Although we do not present any new results, we use existing results to estimate the effect of perturbation in our models. In Chapter 1, we mentioned that the effect of perturbation on the invariant distributions of stochastic matrices have been rigorously studied in the literature. The results relating to the invariant distributions are particularly relevant to our discussion. First, we mention the two types of perturbations that have been mostly studied:

*Linear perturbation:* A perturbation of this type is of the form  $P_2 = P_1 + E$ , where  $E$  is a matrix such that  $\sum_j e(i, j) = 0$  and the norm  $\|E\|$  is small relative to 1. Both matrices  $P_1$  and  $P_2$  are stochastic. The matrix  $P_2$  is also referred to as the *perturbed* matrix.

*Analytic perturbation:* A matrix  $P_0$  is said to be analytically perturbed if the perturbation is in the form of power series

$$P(\epsilon) = P_0 + \epsilon P_1 + \epsilon^2 P_2 + \dots, \quad (38)$$

where the coefficient matrices  $P_k, k = 1, 2, \dots$ , are known and  $\epsilon \in [0, \epsilon_{max}]$ . The above series is assumed to converge in some non-empty neighborhood of  $\epsilon = 0$ . If the matrix coefficients  $P_2 = P_3 = \dots = 0$ , then the linear perturbation is a special case of the analytical perturbation in (38). Various numerical methods to solve perturbation of type (38) can be found in [3]. Our discussion only involves the case of linear perturbation.

Let us consider the case of linear perturbation  $P_2 = P_1 + E$ , where  $P_2$  and  $P_1$  are stochastic matrices and  $E$  is some small perturbation applied to  $P_1$ . Let  $\pi_1$  and  $\pi_2$  be the invariant distributions for matrices  $P_1$  and  $P_2$ , respectively. Then, we have the following

norm-wise perturbation bounds

$$\|\pi_2 - \pi_1\| \leq k\|E\|, \quad (39)$$

where  $k$  is a condition number. The number  $k$  is different under different norms. A comparison of all condition numbers is given in [5]. We state the bound given in [4] which states that sensitivity to perturbation can be measured using mean passage times.

**Theorem 9.** [4]

$$\|\pi_2 - \pi_1\|_\infty \leq \frac{1}{2} \max_j \left[ \frac{\max_{i \neq j} m_{ij}}{r_j} \right] \|\Delta P\|_\infty \quad (40)$$

where  $m_{ij} = \frac{v_{jj} - v_{ij}}{\pi_j}$ , and  $r(j) = \frac{1}{\pi_j}$ ,  $\Delta P = P_2 - P_1$  and  $V$  is the fundamental matrix of matrix  $P_1$ .

Notice that when we redistribute small probabilities back to each island according to method (b) or (c), we form another stochastic matrix that is slightly perturbed. We may use the bound given by Theorem 9 to estimate a bound for our approximation. Let  $P_2$  be obtained from  $P_1$  by directing all small transitions of the interior states according to (b) or (c). For both the cases, it is easy to see that  $\Delta P = P_2 - P_1$ , row sums of  $\Delta P = 0$  and  $\|\Delta P\|_\infty \leq 2\epsilon_{max}$ , where  $\epsilon_{max} = \max\{\epsilon_1, \epsilon_2, \dots, \epsilon_i, \dots, \epsilon_k\}$  and  $\epsilon_i = \sum_j p_1(i, j), i \neq j, i \in R_i, j \in R_j, 1 \leq i, j \leq k$ .

Then, the bound in (40) gives

$$\|\pi_2 - \pi_1\|_\infty \leq \epsilon_{max} \max_j \left[ \frac{\max_{i \neq j} m_{ij}}{r_j} \right]. \quad (41)$$

There is also an explicit formula to calculate  $\pi_1$  in terms of  $\pi_2$  using the famous theorem by Schweitzer (1968)[24].

**Theorem 10.** [24] Let  $(Z_n^1)$  and  $(Z_n^2)$  be two irreducible MCs defined on the same state

space  $S$  with transition matrices  $P_1$  and  $P_2 = P_1 + E$ , where  $E$  is some perturbation. Suppose  $\pi_1$  and  $\pi_2$  are the invariant distributions of MCs  $(Z_n^1)$  and  $(Z_n^2)$ , respectively. And let  $V_1$  be the fundamental matrix of  $P_1$  (defined in (44)). Then

$$\pi_2^T = \pi_1^T (I - EV_1)^{-1} \quad \text{and} \quad \pi_2^T - \pi_1^T = \pi_2^T EV_1. \quad (42)$$

It is also possible to give another bound to the error given by (42).

**Corollary 11.** [24] *Under the conditions of Theorem 10,*

$$\|\pi_2 - \pi_1\|_1 \leq \|E\|_\infty \|V_1\|_\infty. \quad (43)$$

*Proof.* Using  $\pi_2^T - \pi_1^T = \pi_2^T EV_1$ , we can see that  $\|\pi_2 - \pi_1\|_1 = \|(\pi_2^T EV_1)^T\|_1 \leq \|V_1^T\|_1 \|E^T\|_1 = \|V\|_\infty \|E\|_\infty$ .  $\square$

In the context of our discussion, for small  $E = \Delta P$ , it may be possible to use the invariant distribution  $\pi_2$  of the perturbed model  $M_2$  (an IP model) to approximate the actual distribution,  $\pi_1$ , of model  $M_1$ . Note that this bound still requires the calculation of the fundamental matrix  $Z_1$  for an ergodic MC  $(Z_n^1)$  specified by model  $M_1$ . The fundamental matrix  $Z_1$  is given by

$$V_1 = (I - P_1 - A)^{-1}, \quad (44)$$

where  $I = |P_1| \times |P_1|$  identity matrix,  $A = \lim_{n \rightarrow \infty} P^n = e\pi_1^T$  is the limiting matrix of  $P$ , where  $e$  is vector of size  $|P_1| \times 1$ . We can use the algorithm given in [30] to calculate  $V_1$ . Even though we present an algorithm to calculate matrix  $N$ , which is a fundamental matrix of a substochastic matrix, it is interesting to investigate if we can calculate  $V_1$  for the IP model, in parallel.

CHAPTER 4: THE IP MODEL: THE FUNDAMENTAL MATRIX

10 Fundamental matrix N

Let  $M = (S, P)$  be a homogeneous Markov model, where  $S$  is discrete, finite and  $P$  is the transition matrix. Let  $(Z_n)$  be a MC specified by the model  $M$  with an initial distribution  $\mu_0$ . If the MC  $(Z_n)$  is ergodic (irreducible, aperiodic and positive recurrent), then the fundamental matrix  $V$  is defined as

$$V = (I - P - A)^{-1}, \quad (45)$$

where  $I = |P_1| \times |P_1|$  identity matrix and  $A = \lim_{n \rightarrow \infty} P^n = e\pi^T$  is the limiting matrix of  $P$ . The entries of  $V := \{n(x, y) : x, y \in S\}$  gives the expected number of visits of the MC  $(Z_n)$  to a state  $y$  from a state  $x$ . A recursive algorithm to compute the matrix  $V$ , based on the SR approach, is given in [30].

In some cases, one might be interested in calculating the expected number of visits of the MC  $(Z_n)$  to states inside some non-empty and non-absorbing subset  $D \subset S$  before the time of the first exit to  $S \setminus D$ . Let  $C = S \setminus D$  and  $D = \{1, 2, \dots, k\}$  be the enumeration of states in  $D$ . Let  $Z_0 \in D$  and define the first passage time  $\tau = \min\{n > 0 : Z_n \in C\}$ .

**Definition 1.** *The expected number of visits of the MC  $(Z_n)$  to  $y \in D$  from  $x \in D$  before the time of the first exit  $\tau$  to  $C$  is given by*

$$n(x, y) := E \left[ \sum_{n=0}^{\tau-1} \mathbb{1}_{\{Z_n=y\}} | Z_0 = x \right], \quad x, y \in D. \quad (46)$$



In matrix form, equation (46) can be written as,

$$N = \sum_{n=0}^{\infty} Q^n, \quad (47)$$

where  $Q := \{p(x, y) : x, y \in D\}$  is the transition probability matrix for states in  $D$ . The matrix  $N$  is also referred to as the fundamental matrix of the substochastic matrix  $Q$ .

**Lemma 12.** *The fundamental matrix  $N$  satisfies the equalities*

$$(a) N = I + QN, \quad (b) N = I + NQ. \quad (48)$$

According to (48) we have

$$N = (I - Q)^{-1} \stackrel{(by\ 47)}{=} \sum_{n=0}^{\infty} Q^n. \quad (49)$$

The inverse and the sum in (49) exist. Probabilistically,  $C$  can be thought of as an absorbing state and the MC  $(Z_n)$ , which gets absorbed in  $C$  at time  $\tau$ , is, then, a transient MC. Thus each state  $y \in D$  is visited only finitely many times. Hence, the sum in (49) converges and the middle equality follows from (48). Using the results from linear algebra, it can also be shown that the spectral radius  $\sigma$  of any substochastic matrix  $Q$  is less than 1, i.e.  $\sigma(Q) < 1$ , and this implies that  $\sum_{n=0}^{\infty} Q^n$  converges and is equal to  $(I - Q)^{-1}$  [23].

**Lemma 13.** *The distribution of the MC  $(Z_n)$  at the time of the first exit  $\tau$  to  $C$  is given by  $U = NT$ .*

See Appendix for a short proof.

Let  $(Z_n^1)$  be a MC specified by model  $M_1 = (S_1, P_1)$ . And let  $D \subset S_1, G \subset S_1, G \cap D = \emptyset$  and  $S_2 = S_1 \setminus D$  (see Fig. 8a). Let  $\tau_0 = 0, Z_0^1 \in S_2 \setminus G$  and  $\tau_{n+1} = \min\{n > \tau_n : Z_n^1 \in S_2\}$ . By Lemma 1  $(Z_n^2) = (Z_{\tau_n}^1)$  is a censored MC specified by the reduced-model  $M_2 = (S_2, P_2)$  (see Fig. 8b).

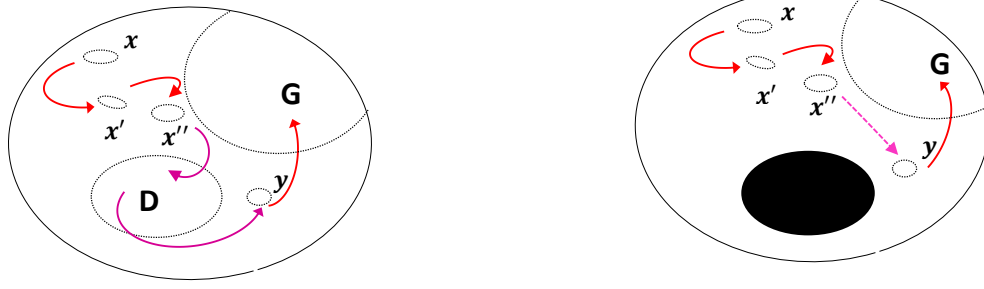
(a) Initial model  $M_1 = (S_1, P_1)$ (b) Censored model  $M_2 = (S_2, P_2)$ 

Figure 8: Censored MCs

Define the Markov times  $\tau^{(i)} := \min\{n : Z_n^i \in G\}$  for  $i = 1, 2$ . Let the matrix  $N_1 := \{n_1(x, y) : x, y \in S_1 \setminus G\}$  be the fundamental matrix of MC  $(Z_n^1)$  and  $N_2 := \{n_2(x, y) : x, y \in S_2 \setminus G\}$  be the fundamental matrix of the censored MC  $(Z_n^2)$ . Then the entries of matrices  $N_1$  and  $N_2$  give the expected number of visits inside  $S_1 \setminus G$  and  $S_2 \setminus G$  by MCs  $(Z_n^1)$  and  $(Z_n^2)$  before the time  $\tau^{(1)}$  and  $\tau^{(2)}$  to set  $G$ , respectively. By Lemma 13, the distributions at time  $\tau^{(i)}$  for MCs  $(Z_n^i)$  are given by  $U_i = N_i T_i$ ,  $i = 1, 2$ , where  $T_i = \{p_i(x, y), x \in S_i \setminus G, y \in G\}$ .

The following Proposition is due to Sonin [31]. It states that the characteristics given by matrices  $N$  and  $U$  are invariant under censoring. See appendix for a short proof.

**Proposition 14.** [31] *Let  $M_1 = (S_1, P_1)$  be a Markov model. And let  $G \subset S_1, G \cap D = \emptyset, D \subset S_1$  and  $S_2 = S_1 \setminus D$ . Let  $M_2 = (S_2, P_2)$  be a  $D$ -reduced Markov model. Let  $N_1$  and  $N_2$  be the fundamental matrices for MCs  $(Z_n^1)$  and  $(Z_n^2)$  defined for  $S_1 \setminus G$  and  $S_2 \setminus G$ , respectively. Then the elements of matrices  $U_1, N_1$  of model  $M_1$  restricted to model  $M_2$  coincide with the corresponding elements of matrices  $U_2, N_2$ , i.e.,  $u_1^G(x, y) = u_2^G(x, y)$ ,  $x \in S_2 \setminus G, y \in G$  and  $n_1^G(x, y) = n_2^G(x, y)$ ,  $x, y \in S_2 \setminus G$ .*

According to Proposition 14, the distributions  $u_1^G(x, y), y \in G$ , of MC  $(Z_n^1)$  at the time of the first exit to  $G$  and the expected number of visits to  $y$  from  $x$  given by  $n_1^G(x, y)$  are

the same for those states  $x$  and  $y$  that remain in model  $M_2$  for MC ( $Z_n^2$ ). This property holds true for any finite number of repeated eliminations, provided that state  $x$  remains in the state space in the calculation of  $u^G(x, y), y \in G$ . Let  $D = \{1, 2, \dots, k-1, k\}$  be the enumeration of states in  $D$ . If we eliminate states  $1, 2, \dots, k-1$  from  $D$ , then, for state  $k$ , Proposition 14 implies

$$u_1^G(k, y) = u_2^G(k, y) = \dots u_{k-1}^G(k, y) = p_k(k, y)/s_k, \quad y \in G,$$

$$n_1^G(k, k) = n_2^G(k, k) = \dots n_{k-1}^G(k, k) = n_k = 1/(1 - p_k(k, k)).$$

Our goal in this chapter is to develop the FUNDQ algorithm to compute the fundamental matrix  $N$ . We will apply the SR approach, use the identities given by (48) and the results of Proposition 14 to develop this algorithm.

## 11 The FUNDQ algorithm

Let  $M = (S, P)$  be a model and suppose  $P$  is a  $r \times r$  transition matrix. Let  $M_1 = (S_1, Q_1)$  be the substochastic model, where  $S_1 \subset S$  is a non-absorbing set,  $n < r$ , and  $Q_1 = \{p(x, y) : x, y \in S_1\} \subset P$  is a  $n \times n$  substochastic matrix for  $S_1$ . Let  $N_1$  be the fundamental matrix of matrix  $Q_1$ . There is also an algorithm presented by [12] to compute  $N_1$ , which was developed in the framework of the SR approach. It has the time complexity of order  $n^3$ . In comparison, the FUNDQ algorithm we develop in the next two sections has time complexity of order  $k(n^2) + (n - k)^3, 1 \leq k < n$ . Because this algorithm is a recursive algorithm, it consists of two stages — a forward stage and a backward stage. In the forward stage, we apply the SR approach on  $Q_1$  to eliminate states from  $S_1$  to obtain a smaller substochastic matrix  $Q_s, 1 \leq s < n$ . We calculate the fundamental matrix  $N_s$  for this matrix  $Q_s$ . In the backward stage, starting with the matrix  $N_s$ , we recursively compute a sequence of fundamental matrices by inserting states that were previously eliminated during

the forward stage, and calculate the matrix  $N_1$  for  $Q_1$ .

We now describe these two stages of the algorithm in detail.

### Stage 1. Forward Stage

In this stage, we apply Lemma 1 Chapter 2 on the substochastic matrix  $Q_1$ , instead of the stochastic matrix  $P_1$ , to eliminate states  $s \in S_1$ ,  $1 \leq s < n$ , in  $n - 1$  iterations, to obtain a matrix  $Q_n$

$$Q_1 \implies Q_2 \cdots \implies Q_k \cdots \implies Q_n = \{p_n(n, n)\}, \quad (50)$$

where the matrix  $Q_n = \{n\}$  has only one state (also see Remark 8). We calculate the fundamental matrix  $N_n$  for  $Q_n$ , which is given by

$$N_n = \frac{1}{1 - p_n(n, n)}.$$

**Remark 8.** Lemma 1 Chapter 2 gives formula (2) to calculate the transition matrix for a censored MC. However, formula (2) can also be applied to a substochastic matrix  $Q$ . Let  $D_0 \subset S_1$ , then the matrix  $Q$  can also be decomposed as

$$Q = \begin{array}{c} \begin{array}{cc} (D_0) & (S_1 \setminus D_0) \end{array} \\ \left[ \begin{array}{cc} A & T \\ R & K \end{array} \right]. \end{array}$$

We can calculate the matrix  $Q_0$ , which describes transitions inside of  $S_1 \setminus D_0$ , by eliminating  $D_0$  using formula (2)

$$Q_0 = K + RN_{D_0}T,$$

where  $N_{D_0} = (I - A)^{-1}$  is the fundamental matrix for the substochastic matrix  $A$ .

## Stage 2. Backward Stage

In the backward stage of the FUNDQ algorithm, we apply formulas given by Proposition 15 to recursively calculate a sequence of matrices

$$N_1 \Leftarrow N_2 \cdots \Leftarrow N_k \cdots \Leftarrow N_n. \quad (51)$$

In each iteration, using Proposition 15 we insert a state  $s \in S_1$  that was eliminated during the forward stage, and compute the sequence in (51). Thus, the matrix  $N_k$  in (51) represents the fundamental matrix for previously inserted states  $\{k + 1, \dots, n - 1, n\}$ , including the state  $k$  that is inserted at this step.

**Remark 9.** *As in Remark 5 Chapter 3, it is also possible to eliminate states  $\{1, 2, \dots, k - 1\} \subset S_1, k < n$ , and compute matrix  $N_k = (I - Q_k)^{-1}$ . Then we can calculate the matrix  $N_1$  by recursively starting from  $N_k$  in  $k - 1$  iterations.*

Remark 9 will be useful in the algorithm to calculate the fundamental matrix for the IP model, later in this chapter.

Proposition 15 gives formulas for one-step recursive calculation of matrices in (51). The sequence in (51) can be calculated from repeated application of Proposition 15 for each previously eliminated state  $s \in S_1$ .

**Proposition 15.** *Let  $M = (S, P)$  be a Markov model and suppose  $M_1 = (S_1, Q_1), S_1 \subseteq S, Q_1 \subseteq P$ , be the substochastic model. Let  $N_1$  be the fundamental matrix for  $Q_1$ . Assume, for simplicity, state  $z \in S_1$  is eliminated from  $Q_1$  and it is located in position  $(1, 1)$ , which we denote by  $A_1$ . Then the matrices  $Q_1$  and  $N_1$  can be represented as below:*

$$Q_1 = \begin{bmatrix} \mathbf{A}_1 & \mathbf{T}_1 \\ \mathbf{R}_1 & \mathbf{K}_1 \end{bmatrix}, \quad N_1 = \begin{bmatrix} \mathbf{u}_1 & \mathbf{n}_1 \\ \mathbf{m}_1 & N_2 \end{bmatrix}. \quad (52)$$

The entries of matrix  $N_1$  are given by

$$\mathbf{n}_1 = \frac{1}{s_z}(T_1 \cdot N_2), \quad \mathbf{m}_1 = \frac{1}{s_z}(N_2 \cdot R_1), \quad \mathbf{u}_1 = \frac{1}{s_z}(1 + n_1 \cdot R_1), \quad (53)$$

where  $s_z = 1 - p_1(z, z)$  and  $N_2 = (I - Q_2)^{-1}$  is the fundamental matrix for the substochastic matrix  $Q_2 = K_1 + \frac{1}{s_z}(R_1 \cdot T_1)$  obtained from matrix  $Q_1$  after eliminating  $\{z\}$ .

We also give a pseudo-code (3) for the FUNDQ algorithm in the Appendix.

*Proof of Proposition 15:*

Suppose state  $z \in S_1$  is eliminated. Then the row vector  $\mathbf{n}_1$  can be obtained from the equality (a) in (48)

$$n_1(z, y) = p_1(z, z)n_1(z, y) + \sum_{x \in S_1 \setminus \{z\}} p_1(z, x)n_1(x, y), \quad y \neq z, y \in S_1. \quad (54)$$

Proposition 14 implies that even after the elimination of state  $z$ ,  $n_1(x, y) = n_2(x, y)$  for all  $x, y \in S_1 \setminus \{z\}$ . We simplify (54) to obtain

$$n_1(z, y) = \frac{1}{s_z} \sum_{x \in S_1 \setminus \{z\}} p_1(z, x)n_2(x, y), \quad (55)$$

which, in matrix form, is  $\mathbf{n}_1 = \frac{1}{s_z}(T_1 \cdot N_2)$ .

From the equality (b) in (48), the  $z^{\text{th}}$  column in matrix  $N_2$  is given by

$$n_1(x, z) = \mathbf{1}_z(x) + n_1(x, z)p_1(z, z) + \sum_{y \in S_1 \setminus z} n_1(x, y)p_1(y, z), \quad x \in S_1.$$

Solving for  $n_1(x, z)$ , we obtain

$$n_1(x, z) = \frac{1}{s_z} \left( \mathbf{1}_z(x) + \sum_{y \in S_1 \setminus \{z\}} n_1(x, y) p_1(y, z) \right). \quad (56)$$

By proposition (14) and noting  $x \neq z$ ,  $\mathbf{1}_z(x) = 0$ , (56) can be written (we name it  $\mathbf{m}_1$ ) as

$$\mathbf{m}_1(x, z) = \frac{1}{s_z} \left( \sum_{y \in D \setminus \{z\}} n_2(x, y) p_1(y, z) \right).$$

In matrix form, this is  $\mathbf{m}_1 = \frac{1}{s_z} (N_2 \cdot R_1)$ . To calculate  $\mathbf{u}_1$ , we make use of the above results.

If we replace  $z$  for  $x$  in equation (56)

$$u_1(z, z) = \frac{1}{s_z} \left( 1 + \sum_{y \in S_1 \setminus \{z\}} n_1(z, y) p_1(y, z) \right).$$

Replacing the equality for  $n_1(z, y)$  from equation (55), we obtain

$$u_1(z, z) = \frac{1}{s_z} \left[ 1 + \frac{1}{s_z} \sum_{y \in S_1 \setminus \{z\}} \left( \sum_{x \in S_1 \setminus \{z\}} p_1(z, x) n_2(x, y) \right) p_1(y, z) \right]. \quad (57)$$

In matrix form, we can write (57) as  $\mathbf{u}_1 = \frac{1}{s_z} \left( 1 + \frac{1}{s_z} ((T_1 \cdot N_2) \cdot R_1) \right) = \frac{1}{s_z} (1 + n_1 \cdot R_1)$ ,

where we used the equality for  $\mathbf{n}_1$  in the last equality.

□

## 12 Order Count for the FUNdq algorithm

Let  $Q_1$  be  $n \times n$  matrix. The calculation of  $N_1$  using matrix inversion has the time complexity of order  $n^3$ . Let  $D = \{1, 2, \dots, k\} \subset S_1, k < n$ , be the eliminated set (See Remark 9). In the FUNdq algorithm,

(i) The calculation of  $W$  (or  $\hat{W}$ ) matrices in  $k$  iterations requires approximately  $2k(n^2 + 1)$  operations.

(ii) The calculation of the matrix  $N_{k+1}$  has the complexity of order  $(n - k)^3$ . In each step, the calculation of  $\mathbf{n}$  and  $\mathbf{m}$  vectors requires approximately  $(n - k)^2$  multiplica-

tions,  $(n - k)(n - k - 1)$  additions and 1 division. The calculation of  $\mathbf{u}$  requires about  $(n - k)^2$  multiplications and  $(n - k - 1)$  additions; in  $k$  iterations, the calculation of  $\mathbf{n}$ ,  $\mathbf{m}$  and  $\mathbf{u}$  requires  $3k(n - k)^2 + n$  operations.

The total count for our algorithm is approximately  $4k(n^2 + 1) + (n - k)^3 + 3k(n - k)^2 + n$ . So, the time complexity is approximately of order  $k(n^2) + (n - k)^3$ .

### 13 Numerical Example

Consider a transition matrix  $P_1$  given in Table 4. In this example, state 7 is absorbing. The submatrix in the square box is the matrix  $Q_1$  and the matrix  $N_1$  is calculated directly using the (first) formula in (49). All the numbers are rounded off to 4 decimal places.

		<b>[1]</b>	<b>[2]</b>	<b>[3]</b>	<b>[4]</b>	<b>[5]</b>	<b>[6]</b>	<b>[7]</b>			<b>[1]</b>	<b>[2]</b>	<b>[3]</b>	<b>[4]</b>	<b>[5]</b>	<b>[6]</b>
$P_1 =$	<b>[1]</b>	0	0.4	0	0	0	0	<b>0.6</b>	$, N_1 =$	<b>[1]</b>	1.6149	1.0248	0.6313	0.3691	0.1943	0.0777
	<b>[2]</b>	0.6	0	0.4	0	0	0	<b>0</b>		<b>[2]</b>	1.5372	2.5619	1.5784	0.9228	0.4857	0.1943
	<b>[3]</b>	0	0.6	0	0.4	0	0	<b>0</b>		<b>[3]</b>	1.4206	2.3677	2.9990	1.7533	0.9228	0.3691
	<b>[4]</b>	0	0	0.6	0	0.4	0	<b>0</b>		<b>[4]</b>	1.2457	2.0763	2.6299	2.9990	1.5784	0.6314
	<b>[5]</b>	0	0	0	0.6	0	0.4	<b>0</b>		<b>[5]</b>	0.9835	1.6391	2.0763	2.3677	2.5619	1.0248
	<b>[6]</b>	0	0	0	0	0.6	0	<b>0.4</b>		<b>[6]</b>	0.5901	0.9835	1.2457	1.4206	1.5372	1.6149
	<b>[7]</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>								

Table 4: Matrices  $P_1$  and  $N_1$

We apply the FUNDQ algorithm for the matrix  $Q_1$ . In the forward stage, we eliminate states 1, 3 and 6. This results into the matrix  $Q_4$  of size  $3 \times 3$ , which describes transitions for states 2, 4 and 5. The fundamental matrix for  $Q_4$  is given by  $N_4 = (I - Q_4)^{-1}$ .

		<b>[2]</b>	<b>[4]</b>	<b>[5]</b>		<b>[2]</b>	<b>[4]</b>	<b>[5]</b>	
$Q_4 =$	<b>[2]</b>	0.48	0.16	0	$N_4 =$	<b>[2]</b>	2.5619	0.9228	0.4857
	<b>[4]</b>	0.36	0.24	0.40		<b>[4]</b>	2.0763	2.9990	1.5784
	<b>[5]</b>	0	0.60	0.24		<b>[5]</b>	1.6391	2.3677	2.5619

Table 5: Fundamental matrix  $N_4$  for  $Q_4$

In the backward stage, we insert states 3, 1 and 6 (in this order) to obtain the matrices



$N_3, N_2$  and  $N_1$ , shown in Table 6 respectively. To show which states are inserted, we use a full-size  $6 \times 6$  matrix (same size as the matrix  $Q_1$ ).

$N_4$							$N_3$						
	[1]	[2]	[3]	[4]	[5]	[6]		[1]	[2]	[3]	[4]	[5]	[6]
[1]	0	0	0	0	0	0	[1]	0	0	0	0	0	0
[2]	0	2.5619	0	0.9228	0.4857	0	[2]	0	2.5619	1.5784	0.9228	0.4857	0
[3]	0	0	0	0	0	0	[3]	0	2.3677	2.9990	1.7533	0.9228	0
[4]	0	2.0763	0	2.9990	1.5784	0	[4]	0	2.0763	2.6299	2.9990	1.5784	0
[5]	0	1.6391	0	2.3677	2.5619	0	[5]	0	1.6391	2.0763	2.3677	2.5619	0
[6]	0	0	0	0	0	0	[6]	0	0	0	0	0	0

$N_2$							$N_1$						
	[1]	[2]	[3]	[4]	[5]	[6]		[1]	[2]	[3]	[4]	[5]	[6]
[1]	1.6149	1.0248	0.6314	0.3691	0.1943	0	[1]	1.6149	1.0248	0.6314	0.3691	0.1943	0.0777
[2]	1.5372	2.5619	1.5784	0.9228	0.4857	0	[2]	1.5372	2.5619	1.5784	0.9228	0.4857	0.1943
[3]	1.4206	2.3677	2.9990	1.7533	0.9228	0	[3]	1.4206	2.3677	2.9990	1.7533	0.9228	0.3691
[4]	1.2457	2.0763	2.6299	2.9990	1.5784	0	[4]	1.2457	2.0763	2.6299	2.9990	1.5784	0.6314
[5]	0.9835	1.6391	2.0763	2.3677	2.5619	0	[5]	0.9835	1.6391	2.0763	2.3677	2.5619	1.0248
[6]	0	0	0	0	0	0	[6]	0.5901	0.9835	1.2457	1.4206	1.5372	1.6149

Table 6: Insertion of states 3, 1 and 6

## 14 Application: The fundamental Matrix for the IP Model

### (a) The fundamental matrices from $W$ matrices

In this section, we discuss the fundamental matrix  $QN = N - I$ , given by (48), in the context of the IP model. Note that the fundamental matrix  $N$  appears in the block form (1) of the full matrix  $W$  in Chapter 2. First, we mentioned that in Chapter 2 we performed all our calculations with the  $W$  matrices. To see why this is relevant to our discussion, recall in stage 1 of the IP algorithm in Chapter 3, we eliminated the interior states  $R_i$  from each island  $L_i, 1 \leq i \leq k$ . Let matrix  $W^{(i)}, 1 \leq i \leq k$ , denote the full matrix corresponding to the  $i^{th}$  island. To calculate the block  $P_{ii}^*$  in matrix  $P^*$  in (24), we let  $W_1^{(i)} = P_i$  and eliminate states in  $R_i$  using formula (8) to obtain matrix  $W_{R_i}^{(i)}$ . As in (1), we can represent

the matrix  $W_{R_i}^{(i)}$  as

$$W_{R_i}^{(i)} = \begin{bmatrix} W_{11}^{(i)} & W_{12}^{(i)} \\ W_{21}^{(i)} & W_{22}^{(i)} \end{bmatrix} = \begin{bmatrix} P_{i_0 i_0} N_i & N_i P_{i_0 i} \\ P_{i i_0} N_i & P_{ii}^* \end{bmatrix}, \quad (58)$$

where submatrices  $P_{i_0 i_0}, P_{i_0 i}, P_{i i_0}$  and  $P_{ii}^*$  were defined in the second matrix of (24). Then, the block  $P_{ii}^*$  are obtained from matrices  $W_{R_i}^{(i)}$  in (58), i.e.  $W_{22}^{(i)} = P_{ii}^*$ . All the other blocks  $P_{jj}^*$  were *calculated* from the matrices  $W_{R_j}^{(j)}, 1 \leq j \leq k$ , in a similar manner. Using these  $W$  matrices, and without any extra calculations, we can obtain a sequence of fundamental matrices  $N_i := \{n_i(x, y) : x, y \in R_i\}, 1 \leq i \leq k$ , which gives the expected number of visits of some MC  $(Z_n)$ , specified by the IP model  $M$ , to the interior states  $R_i$  before the time of the first exit  $\tau^i$  to the ports  $T_i$  in island  $L_i, 1 \leq i \leq k$ , i.e.  $\tau^i = \min(n > 0, Z_n \in T_i)$ . To obtain  $N_i$  from  $W_{R_i}^{(i)}$ , we recall that (48) implies the matrix  $N_i$  satisfies

$$N_i = I + P_{i_0 i_0} N_i = I + W_{11}^{(i)}, \quad (59)$$

where  $I$  is an  $|R_i| \times |R_i|$  identity matrix. Using the submatrix  $W_{11}^{(i)}$  from matrix  $W_{R_i}^{(i)}$  (58), we can calculate  $N_i$  using formula (59) for  $R_i, 1 \leq i \leq k$ . That is, for  $x, y \in R_i$ ,

$$n_i(x, y) = \begin{cases} 1 + W_{11}^{(i)}(x, x), & \text{if } x = y, \\ W_{11}^{(i)}(x, y), & \text{otherwise.} \end{cases}$$

Most importantly, we do not require any additional calculations to compute  $N_i, 1 \leq i \leq k$ , because they are already calculated when we calculate  $\pi$  in Chapter 3.

### (b) The IP Fund algorithm

Here, we will present another algorithm to calculate, by applying the FUNDQ algorithm (Proposition 15), to calculate a fundamental matrix for the case we formulate below.

Let  $M = (S, P)$  be an IP model. Let  $D_i$  be the set of non-absorbing states in each

island  $L_i, 1 \leq i \leq k$ . Let  $Q \subset P$  be the substochastic matrix for states in  $S \setminus D$ , where  $D = \bigcup_{i=1}^k D_i$ . For simplicity, we assume that  $D_i \subset R_i$  and  $D_i \cap T_i = \emptyset$  for all  $1 \leq i \leq k$ . Let  $C_i = R_i \setminus D_i$  and  $S_i = L_i \setminus D_i = C_i + T_i$ . Following the notation in (24) Chapter 3. Then the matrix  $Q$  may also be represented as a union of blocks, *ignoring the parts of  $Q$  that are always zero*; Also see Fig.9a for a matrix representation for the case when  $k = 2$

$$Q = \left( \bigcup_{i=1}^k Q_i \right) \cup \left( \bigcup_{\substack{i,j=1 \\ i \neq j}}^k T_{ij} \right), \quad i, j = 1, 2, \dots, k, \quad (60)$$

where  $Q_i := \{p(x, y), x, y \in S_i\}$  and block  $T_{ij} := \{p(x, y), x \in T_i, y \in T_j\}$ . Note that, the matrix  $N = (I - Q)^{-1}$  gives the expected number of visits in  $S \setminus D$  by some MC ( $Z_n$ ) specified by the IP model, before the time of the first exit  $\tau_0$  to  $D$ , i.e.  $\tau_0 = \min\{n > 0 : Z_n \in D\}$ .

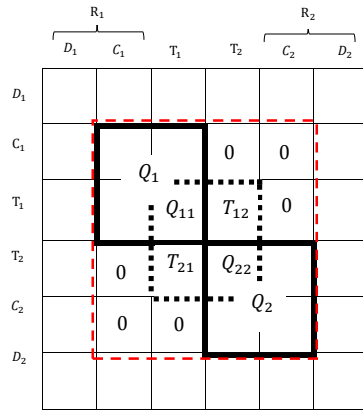
The algorithm to calculate  $N$  consists of three stages. We describe these stages in detail.

#### Stage 1

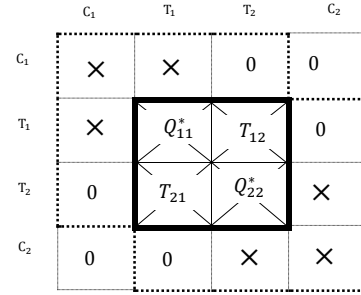
For each  $S_i, 1 \leq i \leq k$ , we eliminate  $C_i$  from the block  $Q_i$  to obtain block  $Q_{ii}^*$ . After  $\bigcup_{i=1}^k C_i$  is eliminated from  $Q$ , we obtain the matrix  $Q^*$ . This matrix can also be represented as

$$Q^* = \left( \bigcup_{i=1}^k Q_{ii}^* \right) \cup \left( \bigcup_{\substack{i,j=1 \\ i \neq j}}^k T_{ij} \right), \quad (61)$$

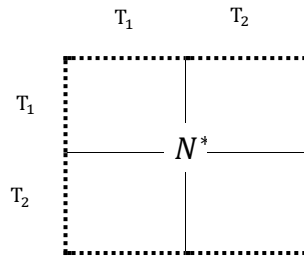
where  $Q_{ii}^* := \{p^*(x, y) : x, y \in T_i\}$ . The matrix  $Q^*$  is the substochastic matrix for the ports  $T$ . According to Lemma 5 and Remark 6 in Chapter 3, the elimination of interior states in  $C_i$  only affects the block  $Q_i$  which results into block  $Q_{ii}^*$ . Thus, we can also calculate each block  $Q_{ii}^*, 1 \leq i \leq k$ , in parallel. The substochastic matrix  $Q^*$  can be computed by aggregating blocks  $Q_{ii}^*$  and the unchanged blocks  $T_{ij}, i \neq j, 1 \leq i, j \leq k$ ; See Figures (9a) and (9b) for an example for the case when  $k = 2$ .



(a) Substochastic matrix  $Q$  (in red)



(b) Parallel elimination of blocks  $C_1$  and  $C_2$



(c) Matrix  $N^*$

Figure 9: Example of stages 1 and 2 for the case  $k = 2$

### Stage 2

We calculate the matrix  $N^* = (I - Q^*)^{-1}$  for  $T$ . This can be done using matrix inversion or by applying the FUNDQ algorithm to  $Q^*$ . According to Remark 9,  $N^*$  is the fundamental matrix for  $Q^*$  obtained after all of the states outside of  $T$  have been eliminated from  $S \setminus D$ ; See Fig.(9c).

### Stage 3

Let us represent the matrix  $N$  as ; See Fig.(10b)

$$N = [F_1 \ F_2 \ \dots \ F_i \ \dots \ F_k]^T, \tag{62}$$

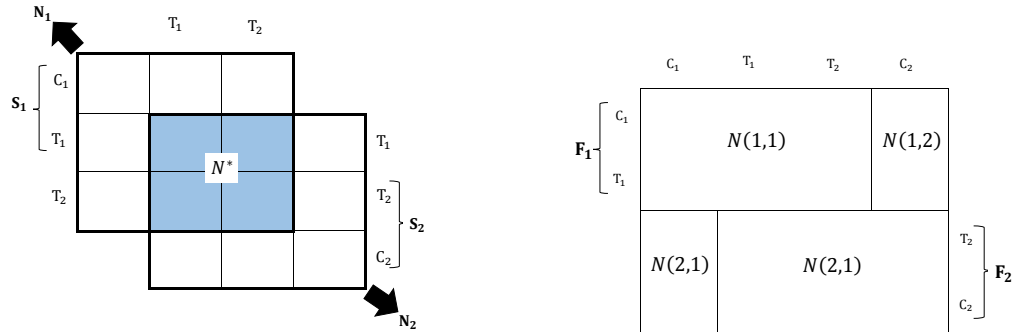
where each row block  $F_i := [N(i, 1) \dots N(i, i) \ N(i, i + 1) \dots N(i, k)]$ ,  $1 \leq i \leq k$ , and each block  $N(i, i) := \{n(x, y) : x \in S_i, y \in C_i + T\}$  and  $N(i, j) := \{n(x, y) : x \in S_i, y \in C_j\}$ ,  $i \neq j$ ,  $1 \leq i, j \leq k$ . We may also represent each row block  $F_i$  as

$$F_i = N(i, i) \cup \left( \bigcup_{\substack{j=1 \\ i \neq j}}^k N(i, j) \right), \quad i = 1, 2, \dots, k.$$

Let us define the block  $N_i := \{n(x, y) : x, y \in C_i + T\}$ . Note that,  $N_{ii} \subset N_i$ . Because of the structure of the IP model and previous results, we have the following result.

**Lemma 16.** *Each block  $N_i$ ,  $1 \leq i \leq k$ , can be computed in parallel.*

*Proof.* Because  $N^* = \bigcap_{i=1}^k N_i$ , the proof follows by replacing  $N_2$  by  $N^*$  in Proposition 15 and recursively inserting eliminated states  $x \in C_i$  for each island  $L_i$ ,  $1 \leq i \leq k$  separately; See Fig.(10a). □



(a) Calculation of blocks  $N_1$  and  $N_2$  in parallel

(b) Row block decomposition of matrix  $N$

Figure 10: Decompositions of the matrix  $N$  for the case  $k = 2$

We now discuss how blocks  $N(i, j)$ ,  $i \neq j$ ,  $1 \leq i, j \leq k$ , can be computed. According to the equality (b) in (48), the expected number of visits from  $x \in S_i$  to  $z \in C_j$ ,  $j \neq i$ , is simply the expected number of visits to  $j \in T_j$  from  $x$ , and then, transitioning from  $j$  to  $z$  with probability  $p(j, z)$ , i.e.  $n(x, z) = \sum_{j \in T_j} n(x, j)p(j, z)$ ,  $x \in S_i$ . Now suppose blocks

$N(i, i), 1 \leq i \leq k$ , were computed using Lemma 16, then let block  $N(T_j) := \{n(x, y) : x \in S_i, y \in T_j\}$  be the part of the block  $N(i, i)$  which gives the expected number of visits to the ports  $T_j$  from states in  $S_i, i \neq j$ . Also let  $z_1^{(j)}, z_2^{(j)}, \dots, z_{c_j}^{(j)}$  be the enumeration of states in  $C_j, |C_j| = c_j$ , in some (arbitrary) order of insertion. Then each column of the block  $N(i, j)$  can be calculated in an iterative manner using Lemma 17.

**Lemma 17.** *Let  $N_j^1 = N(T_j)$ . Then the  $s^{\text{th}}$  column of block  $N(i, j)$ , given by  $\mathbf{m}_s$ , can be calculated for  $s = 1, 2, 3, \dots, c_j$ ,*

$$(i) \mathbf{m}_s = \frac{1}{s_z} \left( N_j^s \cdot \mathbf{R}_s^{(j)} \right) \text{ where } s_z = 1 - p(z_s^{(j)}, z_s^{(j)}),$$

$$(ii) N_j^{s+1} = [N_j^s \quad \mathbf{m}_s],$$

where  $\mathbf{R}_s^{(j)}$  is the probability vector for the insertion of state  $z_s^{(j)}$  state in  $C_j \subset L_j$ , i.e., when  $\{z_{s+1}^{(j)}, z_{s+2}^{(j)}, \dots, z_{c_j}^{(j)}\}$  are still eliminated from  $C_j$ .

*Proof.* The proof follows from Proposition 15 by repeating calculation of vectors  $m_z$  for  $z \in \{z_1^{(j)}, z_2^{(j)}, \dots, z_{c_j}^{(j)}\}$ .  $\square$

In Lemma 17, we can see that the calculation of block  $N(i, j)$  requires knowing the probability vectors  $\mathbf{R}_s^{(j)}$  for each inserted state  $z_s^{(j)}, 1 \leq s \leq c_j$ . In particular, this requires a communication between all of the islands, i.e. probability vector  $\mathbf{R}_s^{(j)}$  is the vector of state  $z_s^{(j)}$  from set  $C_j \subset L_j$  obtained after states  $z_1^{(j)}, z_2^{(j)}, \dots, z_{s-1}^{(j)}$  have been restored from the block  $C_j$ . We suggest a way to obtain the vectors  $\mathbf{R}_s^{(j)}$  without any further calculations. But this requires storing a total of  $c_1 + c_2 + \dots + c_j$  vectors from all the islands, and therefore, can take up lots of memory.

Let  $z_1^{(i)}, z_2^{(i)}, \dots, z_{c_i}^{(i)}$  be the order of elimination of states from set  $C_i \subset L_i, |C_i| = c_i, 1 \leq$

$i \leq k$ . After each state is eliminated from block  $Q_i$  in (60), we obtain a sequence of matrices

$$Q_i = Q_1^{(i)} \rightarrow Q_2^{(i)} \rightarrow \dots \rightarrow Q_s^{(i)} \dots \rightarrow Q_{c_i}^{(i)}, \quad i = 1, 2, \dots, k. \quad (63)$$

If we extract and store the vector  $R_s^{(i)}$ , which is the vector corresponding to the state  $z_s^{(i)} \in C_i, 1 \leq s \leq c_i$ <sup>5</sup>, while eliminating  $C_i$  from matrix  $Q_i$  in stage 1, we obtain the sequence

$$\mathbf{R}_1^{(i)} \rightarrow \mathbf{R}_2^{(i)} \rightarrow \dots \rightarrow \mathbf{R}_s^{(i)} \dots \rightarrow \mathbf{R}_{c_i}^{(i)}, \quad i = 1, 2, \dots, k.$$

Then, for each block  $N(i, j), i \neq j$ , we can simply insert the states in  $C_j$  in the same order as we eliminated them in stage 1 using the probability vectors  $\mathbf{R}_s^{(i)}, 1 \leq s \leq c_i$ . In this way, we can now apply Lemma 17 to calculate the block  $N(i, j), i \neq j$

$$\mathbf{m}_1 \rightarrow \mathbf{m}_2 \dots \rightarrow \mathbf{m}_{c_i}. \quad (64)$$

After calculating all of the blocks  $N(i, j), j \neq i, j = 1, 2, \dots, k$ , we aggregate these blocks, together with the block  $N(i, i)$ , to obtain the row block  $F(i), i = 1, 2, \dots, k$ ; and finally, we obtain the matrix  $N$  according to the representation in (62).

We give a pseudo-code (4) in the Appendix.

## 15 Numerical Example

Consider an IP model  $M = (S, P)$  with the state space consisting of 14 states, i.e.  $S = \{1, 2, 3, \dots, 11, 12, 13, 14\}$ , and the transition matrix  $P$  is given in Table 7. Let  $L_1 = \{1, 2, 3, 4\}, L_2 = \{5, 6, 7, 8\}$  and  $L_3 = \{9, 10, 11, 12, 13, 14\}$  be the three islands of this model with the ports  $T_1 = \{1, 2\}, T_2 = \{5, 6\}$  and  $T_3 = \{9, 10\}$ , respectively. Let  $D = \{3, 13, 14\}$  where  $D_1 = \{3\}, D_2 = \emptyset$  and  $D_3 = \{13, 14\}$ . Define the Markov time  $\tau = \min\{n > 0 : Z_n \in$

---

<sup>5</sup>It will be easier to see this if we let  $Q_i$  to be the matrix  $Q_1$  in Proposition 15, then  $R_s^{(i)}$  is the vector  $R_1$  corresponding to state  $z_s^{(i)}$ .

$D\}$ . Let  $Z_0 \in S \setminus D$ . In this case, we are interested in the expected number of visits to states in islands  $R_1 \setminus D_1$ ,  $R_2$  and  $R_3 \setminus D_3$  by a MC before the time  $\tau$ .

To make our calculations easy to follow, we arrange the substochastic matrix  $Q := \{p(x, y) : x, y \in S \setminus D\}$  (dotted in red) which is shown in Table 7. The fundamental matrix  $N$  for the matrix  $Q$  obtained directly by using formula (49) is also given in Table 8.

		$R_1$		$T$						$R_2$		$R_3$			
				$T_1$		$T_2$		$T_3$							
		3	4	1	2	5	6	9	10	7	8	11	12	13	14
3	10	35	30	25	0	0	0	0	0	0	0	0	0	0	0
4	20	26	10	44	0	0	0	0	0	0	0	0	0	0	0
1	20	10	10	10	10	10	10	20	20	0	0	0	0	0	0
2	23	16	15	4	7	20	10	5	0	0	0	0	0	0	0
5	0	0	7	13	10	13	19	10	18	10	0	0	0	0	0
6	0	0	20	10	7	24	5	10	13	11	0	0	0	0	0
9	0	0	31	10	10	8	8	9	0	0	10	3	5	6	
10	0	0	3	18	17	5	5	11	0	0	19	12	2	8	
7	0	0	0	0	27	11	0	0	22	40	0	0	0	0	
8	0	0	0	0	34	26	0	0	23	17	0	0	0	0	
11	0	0	0	0	0	0	21	10	0	0	20	18	1	30	
12	0	0	0	0	0	0	15	30	0	0	20	10	20	5	
13	0	0	0	0	0	0	7	13	0	0	19	20	11	30	
14	0	0	0	0	0	0	15	40	0	0	3	16	21	5	

$\frac{1}{100}$

Table 7: Matrices  $P$  and  $Q$  (dotted in red)

In this example, we will only calculate the row block  $F_1$ . The calculation of the other blocks is similar.

In stage 1, we eliminate states  $\{4\}$ ,  $\{7, 8\}$  and  $\{11, 12\}$  (in this order) to calculate matrix  $Q^*$  for the ports  $T$ . We also store the probability vectors  $R_s^{(i)}$  for every eliminated state  $z_s^{(i)} \in C_i$  for  $1 \leq i \leq k$ . The matrix  $Q^*$  is shown in Table 9.

In stage 2, using formula (49), we compute the matrix  $N^*$  for  $Q^*$ , which is given in (Table 10).



1.7264	0.8483	1.2045	0.6730	0.8276	0.5930	0.4572	0.4074	0.3871	0.2136	0.1234
0.4534	1.9546	0.8754	0.9603	1.0512	0.9175	0.6871	0.5502	0.5202	0.3247	0.1871
0.5278	0.9825	1.8268	0.9136	1.1530	0.7889	0.6127	0.5601	0.5328	0.2854	0.1651
0.4075	1.2247	1.1191	2.5237	1.6116	1.1843	0.9162	1.1699	1.0814	0.4274	0.2471
0.4196	1.3452	1.0997	1.4805	2.7370	1.0489	0.9196	1.1160	1.0789	0.4094	0.2395
0.3463	1.1765	0.8664	1.0066	1.0697	1.8949	0.7467	0.5690	0.5373	0.4788	0.2585
0.3068	0.8377	0.8954	1.0389	0.9920	0.8523	1.7852	0.5604	0.5267	0.6286	0.3922
0.4117	1.2671	1.1123	2.1569	2.0073	1.1367	0.9174	2.6453	1.8008	0.4211	0.2444
0.4124	1.2742	1.1111	2.0953	2.0738	1.1287	0.9176	1.5619	2.4848	0.4200	0.2440
0.1740	0.5479	0.4621	0.5366	0.5466	0.7778	0.6116	0.2977	0.2805	1.5993	0.4273
0.1986	0.5970	0.5456	0.6333	0.6304	0.7728	0.8554	0.3478	0.3275	0.6447	1.3799

Table 8: Fundamental matrix  $N(F_1(\text{bronze}), F_2(\text{sky blue}), F_3(\text{light green}))$

1	2	5	6	9	10
0.1135	0.1595	0.1000	0.1000	0.2000	0.1000
0.1716	0.1351	0.0700	0.2000	0.1000	0.0500
0.0700	0.1300	0.2756	0.2344	0.1900	0.1000
0.2000	0.1000	0.2191	0.3309	0.0500	0.1000
0.3100	0.1000	0.1000	0.0800	0.1187	0.1225
0.0300	0.1800	0.1700	0.0500	0.1384	0.1956

1	2	5	6	9	10
1.9546	0.8754	0.9603	1.0512	0.9175	0.6871
0.9825	1.8268	0.9136	1.1530	0.7889	0.6127
1.2247	1.1191	2.5237	1.6116	1.1843	0.9162
1.3452	1.0997	1.4805	2.7370	1.0489	0.9196
1.1765	0.8664	1.0066	1.0697	1.8949	0.7467
0.8377	0.8954	1.0389	0.9920	0.8523	1.7852

Table 9: Substochastic matrix  $Q^*$

Table 10: Fundamental matrix  $N^*$

Using Lemma 16, we first compute the block  $N(1, 1)$

4	1	2	5	6	9	10	7	8	11	12
<b>1.7264</b>	0.8483	1.2045	0.6730	0.8276	0.5930	0.4572	0	0	0	0
<b>0.4534</b>	1.9546	0.8754	0.9603	1.0512	0.9175	0.6871	0	0	0	0
<b>0.5278</b>	0.9825	1.8268	0.9136	1.1530	0.7889	0.6127	0	0	0	0

Table 11: Insertion of state 4

We now calculate the block  $N(1, 2)$  using Lemma 17. We first insert state 7 from  $C_2 \subset R_2$ . The vector  $R_{7,2}$  in Table 12 represents the probability of transition from states 5, 6, to the inserted state 7.

$p(7,7)$	0.3308	$p(5,7)$	$R_7^2$	0.2077						
		$p(6,7)$	0.1605							

4	1	2	5	6	9	10	7	8	11	12
1.7264	0.8483	1.2045	0.6730	0.8276	0.5930	0.4572	<b>0.4074</b>	0	0	0
0.4534	1.9546	0.8754	0.9603	1.0512	0.9175	0.6871	<b>0.5502</b>	0	0	0
0.5278	0.9825	1.8268	0.9136	1.1530	0.7889	0.6127	<b>0.5601</b>	0	0	0

Table 12: Insertion of state 7

Similarly, we insert state 8 in Table 13.

$p(8,8)$	0.17	$p(5,8)$	$R_8^2$	0.1						
		$p(6,8)$	0.11							
		$p(7,8)$	0.4							

4	1	2	5	6	9	10	7	8	11	12
1.7264	0.8483	1.2045	0.6730	0.8276	0.5930	0.4572	0.4074	<b>0.3871</b>	0	0
0.4534	1.9546	0.8754	0.9603	1.0512	0.9175	0.6871	0.5502	<b>0.5202</b>	0	0
0.5278	0.9825	1.8268	0.9136	1.1530	0.7889	0.6127	0.5601	<b>0.5328</b>	0	0

Table 13: Insertion of state 8

To calculate the block  $N(1,3)$ , we insert states 11 and 12 from  $C_3 \subset R_3$ . In Table 14, we insert state 11.

$p(11,11)$	0.24	$p(9,11)$	$R_{11}^3$	0.1067						
		$p(10,11)$	0.2167							

4	1	2	5	6	9	10	7	8	11	12
1.7264	0.8483	1.2045	0.6730	0.8276	0.5930	0.4572	0.4074	0.3871	<b>0.2136</b>	0
0.4534	1.9546	0.8754	0.9603	1.0512	0.9175	0.6871	0.5502	0.5202	<b>0.3247</b>	0
0.5278	0.9825	1.8268	0.9136	1.1530	0.7889	0.6127	0.5601	0.5328	<b>0.2854</b>	0

Table 14: Insertion of state 11

Finally, we insert state 12 in Table 15 and aggregating  $N(1, 1)$ ,  $N(1, 2)$  and  $N(1, 3)$ , we have calculated the block  $F_1$ .

$p(12,12)$	0.10	$p(9,12)$	$R_{12}^3$	0.03							
		$p(10,12)$		0.12							
		$p(11,12)$		0.18							
4	1	2	5	6	9	10	7	8	11	12	
1.7264	0.8483	1.2045	0.6730	0.8276	0.5930	0.4572	0.4074	0.3871	0.2136	<b>0.1234</b>	
0.4534	1.9546	0.8754	0.9603	1.0512	0.9175	0.6871	0.5502	0.5202	0.3247	<b>0.1871</b>	
0.5278	0.9825	1.8268	0.9136	1.1530	0.7889	0.6127	0.5601	0.5328	0.2854	<b>0.1651</b>	

Table 15: Insertion of state 12

## CHAPTER 5: APPENDIX

### 16 Proofs

*Proof of Lemma 3:* We need to show that  $\tilde{\pi}_C$  is the invariant distribution of  $P_D$ . Let  $\pi = (\pi_D, \pi_C)$  be the stationary distribution for  $D$  and  $C$  respectively. From the relation (15), we have

$$[\pi_D \ \pi_C] \begin{bmatrix} Q & T \\ R & K \end{bmatrix} = [\pi_D \ \pi_C], \quad (65)$$

Therefore, we have

$$\pi_D = \pi_D Q + \pi_C R, \quad (66)$$

and

$$\pi_C = \pi_D T + \pi_C K. \quad (67)$$

Solving these system of equations, we obtain

$$\pi_D = \pi_C R(I - Q)^{-1} = \pi_C R N_D. \quad (68)$$

Substituting equation (68) in (67), we obtain

$$\pi_C = \pi_C R N_D T + \pi_C K = \pi_C (K + R N_D T) = \pi_C P_D, \quad (69)$$

where the last equality follows from Lemma 1. Normalizing  $\pi_C$  by  $\sum_{i \in C} \pi_i$ , we obtain  $\bar{\pi}_C = \bar{\pi}_C P_D$ . Since the invariant distributions of ergodic MCs are unique,  $\pi_C$  is the invariant distribution of  $P_D$ .  $\square$

*Proof of Lemma 2:* If we apply formula 8 for  $x = z$ , we obtain the equality

$$w_{k+1}(z, y) = \frac{w_k(z, y)}{(1 - w_k(z, z))}. \quad (70)$$

We replace  $w_{k+1}(z, y)$  in (70) in formula (8) and solve for  $w_k(\cdot, y)$ ,

$$w_k(\cdot, y) = w_{k+1}(\cdot, y) - w_{k+1}(\cdot, z)w_{k+1}(z, y), \quad y \in S. \quad (71)$$

Now we applying formula (71) for  $y = z$  and obtain

$$w_k(\cdot, z) = \frac{w_{k+1}(\cdot, z)}{(1 + w_{k+1}(z, z))}. \quad (72)$$

By substituting formula (72) in (8), we obtain insertion formula (12).  $\square$

*Proof of Lemma 4:* (a) Since  $\pi(S_1) = 1$  and  $\pi(S_1) = \sum_{y \in S_1} \pi_1(y) = \pi_1(z) + \sum_{y \in S_2} \pi_1(y)$ .

Let  $\alpha_1 = \sum_{y \in S_2} \pi_1(y)$ , then  $\pi_1(z) = 1 - \alpha_1$ . The equality  $\pi_1(y) = \alpha_1 \pi_2(y), y \in S_2$  follows

from Lemma 3. To prove formula (19), we consider the balance equation at  $z$ ,

$$\begin{aligned}
\pi_1(z) &= \sum_{y \in S_1} \pi_1(y) p_1(y, z), \\
&= \sum_{y \in S_2} \pi_1(y) p_1(y, z) + \pi_1(z) p_1(z, z), \quad y \neq z, \\
&= \frac{1}{1 - p_1(z, z)} \left( \sum_{y \in S_2} \pi_1(y) p_1(y, z) \right), \\
&\stackrel{\text{(by 18)}}{=} \alpha_1 \frac{1}{s_1(z)} \left( \sum_{y \in S_2} \pi_2(y) p_1(y, z) \right), \quad \text{where } s_1(z) = 1 - p_1(z, z), \\
&= \alpha_1 \frac{R_1(z)}{s_1(z)}, \\
&\stackrel{\text{(by 18)}}{=} (1 - \pi_1(z)) \frac{R_1}{s_1(z)}, \\
&= \frac{R_1(z)}{R_1(z) + s_1(z)}, \tag{73} \\
&= \beta_1 R_1.
\end{aligned}$$

Identity for  $\alpha_1$  can easily be derived from (18),

$$\alpha_1 = 1 - \frac{s_1(z)}{R_1(z)} \stackrel{\text{(by 73)}}{=} \frac{s_1}{R_1 + s_1}.$$

□

*Proof of Lemma 12:* The proof for equality (a) follows from Definition 1 by conditioning on an initial transition to some intermediate state  $z \in D$ ,  $z \neq y$ , and counting the first visit as 1 if  $x = y$ ,

$$n(x, y) = \mathbb{1}_x(y) + E \left[ \sum_{n=0}^{\tau-1} \mathbb{1}_{\{Z_n=y\}} | Z_0 = x \right], \quad (74)$$

$$\begin{aligned} &= \mathbb{1}_x(y) + \sum_{n=0}^{\tau-1} P(Z_n = y | Z_0 = x), \\ &= \mathbb{1}_x(y) + \sum_{n=0}^{\tau-1} \sum_{z \in D} P(Z_n = y | Z_1 = z, Z_0 = x) p(x, z), \end{aligned}$$

$$\text{(By MC property } P(Z_n = y | Z_1 = z, Z_0 = x) = P(Z_n = y | Z_1 = z)) \quad (75)$$

$$\begin{aligned} &= \mathbb{1}_x(y) + \sum_{z \in D} p(x, z) E \left[ \sum_{n=0}^{\tau-1} \mathbb{1}_{\{Z_n=y\}} | Z_1 = z \right], \\ &= \mathbb{1}_x(y) + \sum_{z \in D} p(x, z) n(z, y). \end{aligned}$$

In matrix form the last equation can be written as  $N = I + QN$ .

Equality (b) can be proved, in a similar way, by conditioning on the time of the last exit to  $y$ ,

$$\begin{aligned} n(x, y) &= \mathbb{1}_x(y) + E \left[ \sum_{n=0}^{\tau-1} \mathbb{1}_{\{Z_n=y\}} | Z_0 = x \right], \\ &= \mathbb{1}_x(y) + \sum_{n=0}^{\tau-1} P(Z_n = y | Z_0 = x), \\ &= \mathbb{1}_x(y) + \sum_{n=0}^{\tau-1} \sum_{z \in D} P(Z_n = y | Z_{n-1} = z, Z_0 = x) P(Z_{n-1} = z | Z_0 = x), \\ &= \mathbb{1}_x(y) + \sum_{z \in D} p(z, y) \sum_{n=0}^{\tau-1} P(Z_{n-1} = z | Z_0 = x), \\ &= \mathbb{1}_x(y) + \sum_{z \in D} p(z, y) E \left[ \sum_{n=0}^{\tau-1} \mathbb{1}_{\{Z_{n-1}=z\}} | Z_0 = x \right], \\ &= \mathbb{1}_x(y) + \sum_{z \in D} n(x, z) p(z, y). \end{aligned}$$

In matrix form, the last equation can be written as  $N = I + NQ$ . □

**Remark 10.** *It is also possible to prove the equalities in (48) algebraically. Starting with*

(47)  $N = I + Q + Q^2 + \dots$  and left multiplication by  $Q$  gives  $QN = Q(I + Q + Q^2 + \dots) = Q + Q^2 + Q^3 + \dots = (I + Q + Q^2 + \dots)Q = NQ$ .

*Proof of Lemma 13:* For any state  $x \in D$  and  $y \in C$ , MC  $(Z_n)$  can either transition to  $C$  in one step, in which case  $\tau = 1$ , or can transition to state  $z \in D$ , so,

$$\begin{aligned} u(x, y) &= P(Z_\tau = y | Z_0 = x), \\ &= P(Z_\tau = y, \tau = 1 | Z_0 = x) + P(Z_\tau = y, \tau > 1 | Z_0 = x), \\ &= p(x, y) + \sum_{z \in D} p(Z_\tau = y, \tau > 1 | Z_1 = z) p(x, z), \\ &= p(x, y) + \sum_{z \in D} u(z, y) p(x, z). \end{aligned}$$

In matrix form, the last equation is  $U = T + QU = (I - Q)^{-1}T \stackrel{\text{(by 49)}}{=} NT$ .

□

*Proof of Proposition 14:* Let  $x, y \in S_2 \setminus G$ . And let

$$V_{(x,y)}^{(i)} = \sum_{n=0}^{\tau^{(i)}-1} \mathbf{1}_{\{Z_n=y | Z_n=x\}}$$

be the number of visits of MC  $(Z_n^i)$ ,  $i = 1, 2$ , to state  $y$  from state  $x$ . Notice that the elimination of set  $D$  only reduces the length visits to  $y$  but number of visits  $V_{(x,y)}^{(1)} = V_{(x,y)}^{(2)}$  for MCs  $(Z_n^1)$  and  $(Z_n^2)$ , respectively. Let  $\tau^D = \tau^{(1)} - \tau^{(2)}$ . We now need to show  $n_1(x, y) = n_2(x, y)$ . By definition 1, expectation under the probability distribution  $P_1$  gives

$$\begin{aligned} n^1(x, y) &= E^1 \left[ \sum_{n=0}^{\tau^{(1)}-1} \mathbf{1}_{\{Z_n^1=y\}} | Z_0 = x \right], \\ &= \sum_{n=0}^{\tau^{(2)}-1} P_1(Z_n^1 = y | Z_0 = x) + \sum_{z \in D} P_1(Z_{\tau^D}^1 = y | Z_1 = z) p_1(x, z), \\ &= \sum_{n=0}^{\tau^{(2)}-1} P_2(Z_n^2 = y | Z_0 = x) = E^2 \left[ \sum_{n=0}^{\tau^{(2)}-1} \mathbf{1}_{\{Z_n^2=y\}} | Z_0 = x \right] = n^2(x, y), \end{aligned}$$



where the second last equality follows because the probability distribution  $P_2 = K + RU$  as defined in Lemma 1 and  $V_{(x,y)}^{(1)} = V_{(x,y)}^{(2)}$ . Proof of  $u_1^G(x, y) = u_2^G(x, y)$  follows from Lemma 13 for all  $x \in S_1 \setminus G$  and  $y \in G$  since  $p_1(x, y) = p_2(x, y)$ .  $\square$

## 17 Algorithms

---

**Algorithm 1** The GTH/S Algorithm
 

---

```

1: procedure (FORWARD STAGE)
2:   for  $i = r, r - 1 \dots 2$  do                                 $\triangleright$  Assume P is  $r \times r$  matrix
3:      $P_2 \leftarrow \dots P_i \dots \leftarrow P_{r-1} \leftarrow P_r$            $\triangleright |P_2| = 2$  (two states)
4:   end for
5: end procedure
6: procedure (BACKWARD STAGE)
7:    $a_2 = (1 - p_{22})^{-1} p_{12} a_1 \leftarrow a^T P_{r-1} = a^T$            $\triangleright a^T = [a_1 \ a_2]$ 
8:    $k_2 \leftarrow (1 - p_{22})^{-1} p_{22}$ 
9:   for  $j = 3, 4, \dots, r$  do
10:    Compute  $k_j = (1 - p_{jj})^{-1} (p_{1j} + \sum_{i=2}^{j-1} p_{ij} k_i)$ 
11:  end for
12:  Compute the normalization condition  $a_1 = (1 + \sum_{i=2}^r k_i)$ 
13:  for  $i=1, 2, \dots, r$  do
14:     $\pi_i = k_i a_1$ 
15:  end for
16: end procedure

```

---

---

**Algorithm 2** The IP Algorithm
 

---

```

1: procedure (STAGE 1)
2:   for  $i, j = 1, 2 \dots k, i \neq j$  do
3:      $P_{ii}^* \leftarrow P_i$  ▷ eliminate set  $R_i$  from  $L_i$ 
4:      $P_{ij} \leftarrow P$  ▷ Obtain (unchanged) blocks  $P_{ij}$ 
5:   end for
6:   Obtain matrix  $P^*$  ▷ Obtain model  $M^* = (T, P^*)$ 
7: end procedure
8: procedure (STAGE 2)
9:    $\pi^* \leftarrow P^*$  ▷ invariant distribution for model  $M^*$ 
10: end procedure
11: procedure (STAGE 3)(step 1)
12:   for  $i = 1, 2 \dots k$ , do
13:      $\pi_i \leftarrow P_i^*$  ▷ invariant distribution for model  $M_i = (L_i^*, P_i^*)$ 
14:      $\lambda_i \leftarrow w_i \leftarrow \pi_i$  ▷ coefficients for model  $M_i$ 
15:   end for
16: end procedure
17: procedure (STAGE 3)(step 2)
18:   for  $i = 1, 2 \dots k$  do
19:      $e_i \leftarrow \lambda = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$  ▷  $e_i = \pi(L_i^*)$ 
20:      $\pi \leftarrow e_i$  ▷ invariant distribution for model  $M = (S, P)$ 
21:   end for
22: end procedure

```

---



---

**Algorithm 3** The FUNDQ Algorithm
 

---

```

1: Let  $S_1 = \{1, 2, \dots, k-1, k, \dots, n\}$ .
2: procedure (STAGE 1. FORWARD STAGE)
3:    $W_k \leftarrow \dots W_2 \leftarrow W_1 = Q_1$  ▷ (eliminate states  $(1, 2, \dots, k-1)$ )
4:    $N_k \leftarrow Q_k \leftarrow W_k$  ▷  $N_k = (I - Q_k)^{-1}$ 
5: end procedure
6: procedure (STAGE 2. BACKWARD STAGE)
7:    $N_1 \leftarrow N_2 \leftarrow \dots N_{k-1} \leftarrow N_k$  ▷ (insert states  $(1, 2, \dots, k-1)$ )
8: end procedure

```

---

---

**Algorithm 4** The IP FUND Algorithm
 

---

```

1: procedure (STAGE 1)
2:   for  $i = 1, 2, \dots, k$  do
3:      $Q_{ii}^* \leftarrow Q_i$ , store  $\{\mathbf{R}_1^{(i)}, \mathbf{R}_2^{(i)}, \dots, \mathbf{R}_{c_i}^{(i)}\}$ 
4:     Obtain matrix  $Q^*$   $\triangleright Q^*$  is substochastic matrix for  $T$ 
5:   end for
6: end procedure
7: procedure (STAGE 2)
8:    $N^* \leftarrow Q^*$   $\triangleright N^* = (I - Q^*)^{-1}$ 
9: end procedure
10: procedure (STAGE 3)
11:   for  $i = 1, 2, \dots, k$  do
12:      $N(i, i) \leftarrow N_i \leftarrow N^*$   $\triangleright N(i, i) \subset N_i$ 
13:     for  $j = 1, 2, \dots, k, j \neq i$ , do
14:        $N(i, j) \leftarrow \mathbf{m}_1 \leftarrow \mathbf{m}_2 \cdots \leftarrow \mathbf{m}_{c_i} \cdot \leftarrow N(i, i)$ 
15:     end for
16:     Compute  $F_i$   $\triangleright F_i$  is a row block of matrix  $N$  for island  $L_i$ 
17:   end for
18:   Obtain matrix  $N$ 
19: end procedure

```

---

## REFERENCES

- [1] K. Avrachenkov, N. Litvak, and K. S. Pham. A singular perturbation approach for choosing the PageRank damping factor. *Internet Mathematics*, 5(1-2):47–69, 2008.
- [2] K. Avrachenkov, D. Nemirovsky, and K. S. Pham. A survey on distributed approaches to graph based reputation measures. In *The proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, page 82. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.
- [3] K. E. Avrachenkov, J. A. Filar, and P. G. Howlett. *Analytic perturbation theory and its applications*, volume 135. SIAM, 2013.
- [4] G. E. Cho and C. D. Meyer. Markov chain sensitivity measured by mean first passage times. *Linear Algebra and its Applications*, 316(1-3):21–28, 2000.
- [5] G. E. Cho and C. D. Meyer. Comparison of perturbation bounds for the stationary distribution of a markov chain. *Linear Algebra and its Applications*, 335(1-3):137–150, 2001.
- [6] T. Dayar and W. J. Stewart. On the effects of using the Grassmann-Taksar-Heyman method in iterative aggregation-disaggregation. *SIAM Journal on Scientific Computing*, 17(1):287–303, 1996.
- [7] A. Gambin and P. Pokarowski. A new combinatorial algorithm for large Markov chains. In *Computer Algebra in Scientific Computing CASC 2001*, pages 195–211. Springer, 2001.
- [8] W. K. Grassmann and D. A. Stanford. Matrix analytic methods. In *Computational probability*, pages 153–203. Springer, 2000.
- [9] W. K. Grassmann, M. I. Taksar, and D. P. Heyman. Regenerative analysis and steady state distributions for Markov chains. *Operations Research*, 33(5):1107–1116, 1985.
- [10] R. Hassin and M. Haviv. Mean passage times and nearly uncoupled markov chains. *SIAM Journal on Discrete Mathematics*, 5(3):386–397, 1992.
- [11] M. Haviv and L. Van der Heyden. Perturbation bounds for the stationary probabilities of a finite markov chain. *Advances in Applied Probability*, 16(04):804–818, 1984.
- [12] D. P. Heyman. Accurate computation of the fundamental matrix of a markov chain. *SIAM journal on matrix analysis and applications*, 16(3):954–963, 1995.
- [13] J. J. Hunter. Accurate calculations of stationary distributions and mean first passage times in markov renewal processes and markov chains. *Special Matrices*, 4(1):151–175, 2016.
- [14] S. Kamvar, T. Haveliwala, C. Manning, and G. Golub. Exploiting the block structure of the web for computing PageRank. *Stanford University Technical Report*, 2003.
- [15] J. G. Kemeny and J. L. Snell. *Finite Markov chains: With a New Appendix “Generalization of a Fundamental Matrix”*. Springer-Verlag New York, 1976.

- [16] J. Koury, D. McAllister, and W. J. Stewart. Iterative methods for computing stationary distributions of nearly completely decomposable Markov chains. *SIAM Journal on Algebraic Discrete Methods*, 5(2):164–186, 1984.
- [17] A. N. Langville and C. D. Meyer. *Google’s PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2011.
- [18] N. Liu and W. J. Stewart. Markov chains and spectral clustering. In *Performance Evaluation of Computer and Communication Systems. Milestones and Future Challenges*, pages 87–98. Springer, 2011.
- [19] C. D. Meyer. Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Review*, 31(2):240–272, 1989.
- [20] C. D. Meyer and C. D. Wessell. Stochastic data clustering. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1214–1236, 2012.
- [21] C. A. O’Cinneide. Entrywise perturbation theory and error analysis for Markov chains. *Numerische Mathematik*, 65(1):109–120, 1993.
- [22] K. Reichel, V. Bahier, C. Midoux, N. Parisey, J.-P. Masson, and S. Stoeckel. Interpretation and approximation tools for big, dense Markov chain transition matrices in population genetics. *Algorithms for Molecular Biology*, 10(1):1, 2015.
- [23] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [24] P. J. Schweitzer. Perturbation theory and finite markov chains. *Journal of Applied Probability*, 5(02):401–413, 1968.
- [25] T. J. Sheskin. A Markov chain partitioning algorithm for computing steady state probabilities. *Operations Research*, 33(1):228–235, 1985.
- [26] I. Sonin. The elimination algorithm for the problem of optimal stopping. *Mathematical Methods of Operations Research*, 49(1):111–123, 1999.
- [27] I. Sonin. The state reduction and related algorithms and their applications to the study of Markov chains, graph theory, and the optimal stopping problem. *Advances in Mathematics*, 145(2):159–188, 1999.
- [28] I. Sonin and C. Steinberg. Continue, quit, restart probability model. *Annals of Operations Research*, 241(1-2), 2012,2016.
- [29] I. Sonin and C. Steinberg. Elimination and insertion operations for finite Markov chains. In *Modern Trends in Controlled Stochastic Processes: Theory and Applications*. The University of Liverpool, 2015.
- [30] I. Sonin and J. Thornton. Recursive algorithm for the fundamental/group inverse matrix of a Markov chain from an explicit formula. *SIAM Journal on Matrix Analysis and Applications*, 23(1):209–224, 2001.
- [31] I. M. Sonin. The optimal stopping of a Markov chain and recursive solution of Poisson and Bellman equations. In *From Stochastic Calculus to Mathematical Finance: The Shiryaev Festschrift*, pages 609–621. Springer Berlin Heidelberg, 2006.

- [32] I. M. Sonin. A generalized Gittins index for a Markov chain and its recursive calculation. *Statistics & Probability Letters*, 78(12):1526–1533, 2008.
- [33] W. J. Stewart. *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*. Princeton University Press, 2009.
- [34] Y. Q. Zhao. Censoring technique in studying block-structured markov chains. *Advances in Algorithmic Methods for Stochastic Models*, 417:433, 2000.