Computing the renaming functions of a $\rho$-reversible Markov chain

Università Ca’ Foscari Venezia, Italy
Software Dependability and Cyber Security

Matteo Sottana 841883
Supervisor Prof. Sabina Rossi

Academic year 2016/2017
Abstract

A Markov process is a stochastic process where the probability distribution of the future states depend only on the information that we have about the current time, not on the information we have on the past states. Markov models find use in many areas, they are applied to model communication networks, they can be used to prove many of the theorems of the queuing theory and are also applied to study cruise control systems, lines of customers, search engines and much more. In this thesis we focus on the application of $\rho$-reversibility to Markov models. A Markov process is $\rho$-reversible if it is stochastically identical to its reversed process modulo a renaming on the state space. The importance of $\rho$-reversible Markov models is related to the fact that this type of chains don’t require the solution of the system of global balance equation for the computation of their stationary distribution. We propose an heuristic algorithm that is able to find all the possible renaming functions of a $\rho$-reversible continuous time Markov chain (CTMC). We show an application of the algorithm to a real problem, the fair allocation of resources in a Wireless Sensor Network (WSN). The algorithm is used to show that the underlying Markov chain of a distributed algorithm for bandwidth allocation (FACW) in a WSN is dynamically reversible. We also use a stochastic simulation to show that the model of the FACW protocol can be accurately used to estimate performance indices.
Dedicated to my mother and my sister. You’ve always been there for me, when I needed you the most.
Contents

1 Introduction 1
  1.1 Related work .................................................. 2
  1.2 Contributions .................................................. 3
  1.3 Structure of the thesis ........................................ 4

2 Theoretical Background 5
  2.1 Preliminaries on Markov processes ......................... 5
  2.2 Global Balance Equations (GBE) .............................. 6
  2.3 Reversibility .................................................... 6
  2.4 Reversed process ............................................... 8
  2.5 $\rho$-Reversibility ............................................. 9
  2.6 Discrete time Markov chains ................................. 14
  2.7 From CTMC to DTMC ............................................ 15

3 Properties of $\rho$-reversible Markov chains 17

4 Computing a $\rho$-reversible Markov chain’s renaming functions 23
  4.1 Time complexity ................................................ 26

5 Examples 28
  5.1 Example 1 ....................................................... 28
  5.2 Example 2 ....................................................... 29
  5.3 Example 3 ....................................................... 30
  5.4 Example 4 ....................................................... 32

6 The Fair Allocation Control Window protocol 34
  6.1 The FACW protocol .......................................... 34
  6.2 The stochastic model ....................................... 35
  6.3 Testing the FACW model .................................... 35
  6.4 Closed form stationary distribution ....................... 36
6.5 Performance indices

7 Simulations
   7.1 Validation of the simulator
   7.2 Simulation of general (non-mesh) networks

8 Conclusion

9 Acknowledgments
1 Introduction

Markov chains have been widely used to study the performance of computer systems and software architectures. In the past decades several formalism have been developed with the goal to allow a stochastic model to be specified in a compact way by using features such as compositions and hierarchical approach. Despite the availability of compact representations, a stochastic process does not necessarily allow to be analyzed efficiently; even a simple high-level model may suffer from the so called state space explosion problem that makes the computation of the steady state numerically intractable.

Discrete and continuous time Markov processes are the foundations of many approaches to the performance evaluation of computer systems, for this reason, many formalisms that define performance and reliability models have an underlying stochastic process that is a Markov chain. The analysis of the underlying process allows to derive the performance indices of the model. These indices are often computed in steady-state (if it exists), when the time elapsed from the initial instant tends to infinity.

The problem of deriving the stationary probability distribution, of a stochastic process, requires to solve a set of linear equations, called global balance equations (GBE), that has an order equal to the cardinality of the state space of the process. The analysis of Markov chains, both at continuous and discrete time, can sometimes be simplified, allowing for a numerical or analytical tractability of the computation of some performance indices, by the application of techniques based upon: internal symmetries, for instance reversibility [1, 2]; state aggregation, such as lumpability [3]; composition, like product forms [4, 5].

The time-reversibility of Markov stochastic processes has been introduced and applied to the analysis of stochastic networks and Markov chains by Kelly [1]. A time-reversible Markov process has the property that the process we obtain by reversing the direction of time has the
same probabilistic behavior of the original one. Applications of these results has lead to the characterization of product-form solutions for models with underlying time-reversible Markov chains. Product-forms solutions allow to study the sub-components of a system, reducing the state space, and then compute the metrics of the whole system by computing the product of the metrics of all the sub-components. In particular the product-form theory allows for the derivation of the steady-state distribution of a system as the normalized product of the steady-state distributions of the systems sub-components, each considered in isolation and opportunely parametrized.

In this thesis we focus on testing the $\rho$-reversibility of continuous time Markov chains. A Markov process is $\rho$-reversible if it is stochastically identical to its reversed process modulo a renaming on the state space. The notion of $\rho$-reversibility allows to efficiently compute the stationary probabilities since the system of GBE does not need to be solved and each unnormalized state probability can be computed with an explicit formula in terms of the Markov chain’s transition probabilities/rates.

We propose an heuristic algorithm for the computation of all the state space renaming functions of a $\rho$-reversible continuous time Markov chain (CTMC), it can also work with discrete time Markov chains (DTMC), with minor adjustments, but we are more interested in the former case. We prove the correctness of the proposed algorithm and indicate its time complexity; we show that, if applied to test Markov chains obtained from the models of real systems, the time complexity of the algorithm decrease considerably. Finally, we show an application of our algorithm to a model of a wireless sensor network.

1.1 Related work

In [1] the author deeply discusses the notion of time reversibility both for DTMCs and CTMCs, and several applications are illustrated. In [6]
the authors introduce the notion of dynamic reversibility to study physical systems such as the growth of two dimensional crystals. The authors explore the relations among different definitions of lumpability and the notion of time-reversed Markov chain in [7]. In [2] the authors introduce the concept of $\rho$-reversibility, a notion of reversibility modulo a renaming of the states. In [8] the authors introduce the definition of auto-reversibility that allows one to exploit the symmetrical structures of a class of CTMCs to derive the steady-state probabilities in an efficient way. In [3] the authors propose the idea of using a general permutation of states for comparing forward and reversed processes at continuous time. The authors review the main results about time-reversible Markov processes and discusses how to apply them to tackle the problem of the quantitative evaluation of reversible computations in [9]. In [10] the authors show new results on both reversed stationary Markov processes as well as Markovian process algebra. A connection between exact lumping and time reversibility is proposed in [11]. In [12] the authors focus on the problem of defining quantitative stochastic models for concurrent and cooperating reversible computations

1.2 Contributions

In this thesis we propose an heuristic algorithm that allows us to compute all the renaming functions of a continuous time Markov chain. The algorithm can be used to test the reversibility, dynamic reversibility and $\rho$-reversibility for CTMC in a single execution. Despite the existence of algorithms that can test the reversibility of a Markov chain, there is a lack of literature, as far as we know, about the test of the $\rho$-reversibility of a chain. Indeed, given a stationary Markov chain, deciding whether there exists a permutation of it’s nodes such that the chain is $\rho$-reversible is a NP problem. Our algorithm can compute all the valid renaming functions of a continuous time Markov chain
and, when applied to chains that model real systems, it can often do that in polynomial time. The algorithm can also work with discrete time Markov chains with minor adjustments. We apply our algorithm to a real case of study, the FACW protocol. To show a validation of the model, underlying the FACW protocol, we resort to a stochastic simulation implemented with the Java language.

1.3 Structure of the thesis

The thesis is structured as follows. Section 2 reviews the theory of continuous and discrete time Markov chains which will be required in the following sections. Section 3 discusses the properties of $\rho$-reversible Markov chains. Here, we also prove some necessary conditions for the $\rho$-reversibility of a Markov chain that will be used in the proposed algorithm. In Section 4 we describe our algorithm, we also state its time complexity. Section 5 proposes some examples of application of our algorithm. In Section 6 we recall the the definition of the model proposed in [13] and show the application of our algorithm to a CTMC that models the FACW protocol. We show that the tested chain is dynamically reversible. In Section 7 we resort to a stochastic simulation to show that the model proposed in [13] can be used to estimate the system throughput and fairness accurately. Section 8 concludes the thesis.
2 Theoretical Background

In this section we recall the theory and the notions behind Markov processes which will be used in the following sections. The proposed arguments apply to continuous time Markov processes with a discrete state space, also called continuous time Markov chains (CTMC). The following theory may be formulated also for discrete time Markov chains (DTMC) [14, 1].

2.1 Preliminaries on Markov processes

Let $X(t)$ be a stochastic process taking values in a countable state space $S$ for $t \in \mathbb{R}^+$. The stochastic process $X(t)$ is said to be stationary if $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same distribution as the process $(X(t_1 + \tau), X(t_2 + \tau), \ldots, X(t_n + \tau)$ for all $t_1, t_2, \ldots, t_n, \tau \in \mathbb{R}^+$. The stochastic process $X(t)$ is a Markov process if for $t_1 < t_2 < \cdots < t_n < t_{n+1}$ the joint distribution of $(X(t_1), X(t_2), \ldots, X(t_n), X(t_{n+1}))$ is such that

$$P(X(t_{n+1}) = x_{n+1} \mid X(t_1) = x_1, X(t_2) = x_2, \ldots, X(t_n) = x_n) = P(X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n).$$

In a Markov process the past evolution does not influence the conditional probability distribution of future behavior. A continuous time Markov chain (CTMC) is a continuous time Markov process with a discrete state space $S$.

A Markov process is said to be time homogeneous if the conditional probability $P(X(t + \tau) = b \mid X(t) = a)$ does not depend upon $t$, and is irreducible if every state in $S$ can be reached from every other state. A state in a Markov process is called recurrent if it’s guaranteed that the process will eventually return to the same state (the probability of going back to the state is one). A recurrent state is called positive-recurrent if the expected number of steps until the process returns to
it is finite. A Markov process is \textit{ergodic} if it is irreducible and all its states are positive recurrent. A process satisfying all these assumptions possesses a \textit{steady-state distribution}, that is the unique collection of positive numbers \(\pi_k\) with \(k \in S\) summing to unity such that:

\[
\lim_{t \to \infty} P(X(t) = k \mid X(0) = x) = \pi_k. \tag{1}
\]

The transition rate between two states \(a\) and \(b\) is denoted by \(q_{ab}\). The infinitesimal generator \(Q\) of a Markov process is a matrix where the \(q_{ab}\)’s are the off-diagonal elements while the diagonal elements are formed as the negative sum of the non-diagonal elements of each row, \(q_{aa} = -\sum_{b \in S, b \neq a} q_{ab}\).

\section*{2.2 Global Balance Equations (GBE)}

The steady-state distribution \(\pi\) is the unique vector of positive numbers \(\pi_k\) with \(k \in S\) satisfying Equation (1). The vector \(\pi\) sums to unity and satisfies the system of the global balance equations (GBE):

\[
\pi Q = 0.
\]

Any non-trivial solution of the GBE differs by a constant but only one satisfies the normalizing condition

\[
\sum_{k \in S} \pi_k = 1.
\]

\section*{2.3 Reversibility}

The computation of the global balance equation of an ergodic CTMC can be greatly simplified if the chain satisfies the property that when the direction of time is reversed then the behavior of the process remains the same.
Given an ergodic CTMC in steady-state, \( X(t) \) with \( t \in \mathbb{R}^+ \), we call \( X(\tau - t) \) its reversed process. In the following we denote by \( X^R(t) \) the reversed process of \( X(t) \). It can be shown that \( X^R(t) \) is also a stationary CTMC.

We say that \( X(t) \) is reversible if it is stochastically identical to \( X^R(t) \), i.e., \( (X_{t_1}, \ldots, X_{t_n}) \) has the same distribution as \( (X_{\tau-t_1}, \ldots, X_{\tau-t_n}) \) for all \( t_1, t_2, \ldots, t_n, \tau \in \mathbb{R}^+ \) [1]. For an ergodic CTMC there exist simple necessary and sufficient conditions for reversibility expressed in terms of the equilibrium distribution \( \pi \) and the transition rates \( q_{ij} \).

**Proposition 1.** *(Detailed balance equations [1]) A stationary Markov process with state space \( S \) and infinitesimal generator \( Q \) is reversible if and only if the following system of detailed balance equations are satisfied for a set of positive \( \pi_a, a \in S \) summing to unity:

\[
\pi_a q_{ab} = \pi_b q_{ba}
\]

for all states \( a, b \in S \), with \( a \neq b \). If such a set of \( \pi_a \) exists, then it is the equilibrium distribution of the reversible chain.

Since a reversible CTMC \( X(t) \) and its dual \( X^R(t) \) are stochastically identical they have the same steady-state distribution.

Kolmogorov’s criterion states that the reversibility of a process can be established directly from its transition rates. In particular the following proposition can be proved:

**Proposition 2.** *(Kolmogorov’s criterion [1]) A stationary Markov process with state space \( S \) and infinitesimal generator \( Q \) is reversible if and only if its transition rates satisfy the following equation: for every finite sequence of states \( x_1, x_2, \ldots, x_n \in S \),

\[
q_{x_1x_2}q_{x_2x_3} \cdots q_{x_{n-1}x_n}q_{x_nx_1} = q_{x_1x_n}q_{x_nx_{n-1}} \cdots q_{x_3x_2}q_{x_2x_1}.
\] (2)
2.4 Reversed process

Even when $X(t)$ is not reversible the reversed process $X^R(t)$ of a stationary Markov process $X(t)$ can always be defined. In [15] the author shows that the reversed process $X^R(t)$ is a CTMC and proves that the transition rates can be defined in terms of the equilibrium distribution of the process $X(t)$.

**Proposition 3.** (Reversed process transition rates [15]) Given the stationary Markov process $X(t)$ with state space $S$, infinitesimal generator $Q$, and stationary distribution $\pi$, the transition rates of the reversed process $X^R(t)$, forming its infinitesimal generator $Q^R$, are defined as follows:

$$q^R_{ba} = \frac{\pi_a}{\pi_b} q_{ab},$$

(3)

where $q^R_{ba}$ denotes the transition rate from state $b$ to state $a$ in the reversed process. The equilibrium distribution $\pi$ is the same for both the forward and the reversed process.

Equation (3) holds also for any non-trivial solution of the GBE. In [15] the Kolmogorov’s criteria is generalized in order to be applicable also to non-reversible CTMCs. For a given state $a$ we denote by $q_a$ the quantity $\sum_{h \in S, h \neq a} q_{ab}$, similarly we denote $q^R_a$ the quantity $\sum_{h \in S, h \neq a} q^R_{ab}$ for the corresponding state in the reversed process.

**Proposition 4.** (Kolmogorov’s generalised criteria [15]) For a given a stationary Markov process with state space $S$ and infinitesimal generator $Q$, $Q^R = (q^R_{ab})_{a,b \in S}$ is the infinitesimal generator of its reversed process if and only if the following conditions hold:

1) $q^R_x = q_x$ for every state $x \in S$;

2) for every finite sequence of states $x_1, x_2, \ldots, x_n \in S$,

$$q_{x_1} q_{x_2} q_{x_3} \cdots q_{x_n} q_{x_{n+1}} = q^R_{x_1} q^R_{x_2} q^R_{x_3} \cdots q^R_{x_n} q^R_{x_{n+1}}.$$  

(4)
Proposition 4 can be used to build a proof method for verifying whether a vector $\pi$ satisfies the GBE system $\pi Q = 0$ for a given process $X(t)$. It consists of:

1. defining the reversed process $X^R(t)$ using Proposition 3 and assuming $\pi$,

2. verifying the generalised Kolmogorov’s criteria of Proposition 4.

If the generalised Kolmogorov’s criteria are verified and $\sum_{k \in S} \pi_k = 1$ then, by uniqueness of the steady-state distribution, we can conclude that $\pi$ is the steady-state distribution of the process.

2.5 $\rho$-Reversibility

Many stochastic processes are not reversible, however they may be reversible modulo some renaming of the state names. In this section we generalize the definition of reversibility and introduce the notion of $\rho$-reversibility [3, 7].

A renaming function $\rho$ over the state space of a Markov process is a bijection on $S$. For a Markov process $X(t)$ with state space $S$ we denote by $\rho(X)(t)$ the same process where the state names are changed according to $\rho$. More formally, let $Q$ and $\pi$ be the infinitesimal generator and the equilibrium distribution of $X(t)$; $Q'$ and $\pi'$ be the infinitesimal generator and the equilibrium distribution of $\rho(X)(t)$. For all $a, b \in S$ it holds that,

$$q_{ab} = q'_{\rho(a)\rho(b)} \quad \text{and} \quad \pi_a = \pi'_{\rho(a)}.$$ 

**Definition 1.** ($\rho$-reversibility) A CTMC $X(t)$ with state space $S$ is $\rho$-reversible if there exists a renaming $\rho$ on $S$ such that $X(t)$ and $\rho(X^R)(t)$ are stochastically identical. In this case we say that $X(t)$ is $\rho$-reversible with respect to $\rho$. 

9
Lemma 1. Let $X(t)$ be a CTMC with state space $S$ and infinitesimal generator $Q$ and $\rho$ be a renaming of the states such that $q_a = q_{\rho(a)}$ for all $a \in S$. If there exists a set of positive real numbers $\pi_a$, $a \in S$, summing to unity satisfying:

$$\pi_a q_{ab} = \pi_b q_{\rho(b)\rho(a)} \quad \forall a, b \in S,$$

then $\pi_a$ is the unique equilibrium distribution of $X(t)$.

Proof. We carry out the proof by substitution of the expression of $\pi_a$ given by Equation (5) in the system of global balance equations of $X(t)$. We have:

$$\pi_a \sum_{b \in S, a \neq b} q_{ab} = \sum_{b \in S, a \neq b} \pi_b q_{ba},$$

that divided by $\pi_a$ gives:

$$\sum_{b \in S, a \neq b} q_{ab} = \sum_{b \in S, a \neq b} \frac{\pi_b}{\pi_a} q_{ba} = \sum_{b \in S, a \neq b} q_{\rho(a)\rho(b)}.$$

Since $\rho$ is a bijection, this reduces to $q_a = q_{\rho(a)}$ which is an identity by hypothesis. \qed

Proposition 5. (Detailed balance equations for $\rho$-reversible processes) Let $X(t)$ be a CTMC with state space $S$ and infinitesimal generator $Q$, let $X^R(t)$ be the reversed process for $X(t)$ such that the infinitesimal generator $Q^R$ of $X^R(t)$ is defined as stated in Proposition 3 and let $\rho$ be a renaming on $S$ such that $q_a = q_{\rho(a)}$ for all $a \in S$. Then $X(t)$ is $\rho$-reversible with respect to $\rho$ if and only if there exists a set of positive real numbers $\pi_a$ summing to unity, with $a \in S$, such that the following system of detailed balance equations are satisfied: for all $a, b \in S$ with $a \neq b$:

$$\pi_a q_{ab} = \pi_b q_{\rho(b)\rho(a)}$$
If such a solution $\pi_a$ exists then it is the equilibrium distribution of $X(t)$, while $\pi_{\rho(a)} = \pi_a$ is the equilibrium distribution of $X^R(t)$.

**Proof.** Let $Q$ be the infinitesimal generator of $X(t)$ and $Q'$ be the infinitesimal generator of $\rho(X^R)(t)$. We prove that a $\rho$-reversible CTMC satisfies the system of detailed balance equations. Observe that by definition of renaming and the fact that $\rho(X^R)(t)$ and $X^R(t)$ are stochastically identical, for every $a, b \in S$
\begin{equation}
q'_{\rho(b)\rho(a)} = q^R_{ba}.
\end{equation}

Since $X(t)$ and $\rho(X^R)(t)$ are stochastically identical we have $q'_{\rho(b)\rho(a)} = q_{\rho(b)\rho(a)}$ and hence, by Equation (6), $q^R_{ba} = q_{\rho(b)\rho(a)} = q_{\rho(b)\rho(a)}$ with $b' = \rho(b)$. Then, by Proposition 3, we obtain the desired result, that is $\pi_a q_{ab} = \pi_b q_{b'\rho(a)}$. In order to prove that $q_a = q_{\rho(a)}$ it suffices to observe that since $X(t)$ and $\rho(X^R)(t)$ are stochastically identical then the residence time in state $a$ and $\rho(a)$ must be the same, hence $q_a = q^R_{\rho(a)}$ and by Proposition 4 the residence time in the forward and the reversed processes must be the same for each state, i.e., $q^R_{\rho(a)} = q_{\rho(a)}$. Observe that $\pi_a$ is the equilibrium distribution of $X(t)$ by Lemma 1.

Now we prove that the detailed balance equations and the condition $q_a = q_{\rho(a)}$ imply the fact that $X(t)$ is $\rho$-reversible with respect to $\rho$. By Proposition 3 we write:
\begin{equation}
q^R_{\rho(a),\rho(b)} = \frac{\pi_b}{\pi_a} q_{\rho(b)\rho(a)} = \frac{\pi_b}{\pi(a)} q_{\rho(b)\rho(a)}.
\end{equation}

By using the detailed balance equations, we have $q_{\rho(b)\rho(a)} = \pi_a / \pi_b q_{ab}$, obtaining $q^R_{\rho(a),\rho(b)} = q_{ab}$.

By applying Proposition 4 we obtain the following characterization of $\rho$-reversibility.
Proposition 6. Let $X(t)$ be a stationary Markov process with state space $S$ and infinitesimal generator $Q$, and let $\rho : S \to S$ be a renaming on $S$. $X(t)$ is $\rho$-reversible if and only if for every finite sequence of states $x_1, x_2, \ldots, x_{n-1}, x_n \in S$,

$$q_{x_1x_2}q_{x_2x_3} \cdots q_{x_{n-1}x_n}q_{x_nx_1} = q_{\rho(x_1)}q_{\rho(x_2)}q_{\rho(x_3)}q_{\rho(x_4)}q_{\rho(x_5)}q_{\rho(x_6)}q_{\rho(x_7)}q_{\rho(x_8)}q_{\rho(x_9)}q_{\rho(x_{10})}$$

(7)

Proof. Consider a finite sequence of states $x_1, x_2, \ldots, x_n \in S$. By Proposition 5

$$q_{x_1x_2}q_{x_2x_3} \cdots q_{x_{n-1}x_n}q_{x_nx_1} = q_{\rho(x_1)}q_{\rho(x_2)}q_{\rho(x_3)}q_{\rho(x_4)}q_{\rho(x_5)}q_{\rho(x_6)}q_{\rho(x_7)}q_{\rho(x_8)}q_{\rho(x_9)}q_{\rho(x_{10})}$$

that, by simplifying yields

$$q_{x_1x_2}q_{x_2x_3} \cdots q_{x_{n-1}x_n}q_{x_nx_1} = q_{\rho(x_1)}q_{\rho(x_2)}q_{\rho(x_3)}q_{\rho(x_4)}q_{\rho(x_5)}q_{\rho(x_6)}q_{\rho(x_7)}q_{\rho(x_8)}q_{\rho(x_9)}q_{\rho(x_{10})}$$

Observe that, since $X(t)$ is irreducible, for all $x, k \in S$ we can find a chain $x = x_1 \to x_2 \to \ldots \to x_{n-1} \to x_n = k$ (for $n \geq 1$) of one-step transitions. From the hypothesis that $X(t)$ and $\rho(X(t))$ are stochastically identical there is also a chain $\rho(k) = \rho(x_n) \to \rho(x_{n-1}) \to \ldots \to \rho(x_2) \to \rho(x_1) = \rho(x)$. Consider an arbitrary state $x_0 \in S$ as a reference state and $x \in S$. Let $x = x_n \to x_{n-1} \to \ldots \to x_1 \to x_0$ and $\rho(x_0) \to \rho(x_1) \to \ldots \to \rho(x_{n-1}) \to \rho(x_n)$ ($n \geq 1$) be two chains of one-step transitions in $X(t)$. we prove that:

$$\pi_x = C_{x_0} \prod_{k=1}^{n} q_{\rho(x_{k-1})}\frac{q_{\rho(x_k)}}{q_{x_kx_{k-1}}}$$

(8)

where $C_{x_0} \in \mathbb{R}^+$. We prove that $\pi_x$ is well-defined. If $x = y_m \to y_{m-1} \to \ldots \to y_1 \to y_0 = x_0$ ($m \geq 1$) is another chain, we can always
find a chain \( x_0 = w_0 \to w_1 \to \ldots \to w_{l-1} \to w_l = x \). By hypothesis we have:

\[
\prod_{k=1}^{m} q_{y_k y_{k-1}} \prod_{k=1}^{l} q_{\rho(w_{k-1})\rho(w_k)} = \prod_{k=1}^{l} q_{\rho(w_k)\rho(w_{k-1})} \prod_{k=1}^{m} q_{\rho(y_{k-1})\rho(y_k)}.
\]

Moreover, consider the one-step chain \( x = x_n \to x_{n-1} \to \ldots \to x_1 \to x_0 = w_0 \to w_1 \to \ldots \to w_{l-1} \to w_l = x \), we have:

\[
\prod_{k=1}^{n} q_{x_k x_{k-1}} \prod_{k=1}^{l} q_{\rho(w_{k-1})\rho(w_k)} = \prod_{k=1}^{l} q_{\rho(w_k)\rho(w_{k-1})} \prod_{k=1}^{n} q_{\rho(x_{k-1})\rho(x_k)}.
\]

From the previous two equations we obtain:

\[
\prod_{k=1}^{m} \frac{q_{\rho(y_{k-1})\rho(y_k)}}{q_{y_k y_{k-1}}} = \prod_{k=1}^{n} \frac{q_{\rho(x_{k-1})\rho(x_k)}}{q_{x_k x_{k-1}}}.
\]

Hence:

\[
\pi_x = C_{x_0} \prod_{k=1}^{n} \frac{q_{\rho(x_{k-1})\rho(x_k)}}{q_{x_k x_{k-1}}},
\]

where \( C_{x_0} \in \mathbb{R}^+ \), is well-defined.

In order to prove that this is the stationary probability of state \( x \in S \) we show that it satisfies the system of global balance equations for \( x \). Indeed,

\[
\pi_x = \sum_{j \in S} \pi_x q_{jx},
\]

which can be written as:

\[
1 = \sum_{j \in S} \frac{\pi_j}{\pi_x} q_{jx}.
\]
By Proposition 5 we have:

$$1 = \sum_{j \in \mathcal{S}} \frac{q_{\rho}(x)\rho(j)}{q_{jx}} = \sum_{j \in \mathcal{S}} q_{\rho}(x)\rho(j)$$

that is an identity.

Now let $a, b \in \mathcal{S}$ such that $q_{ab} > 0$. Then

$$\pi_x = C_{x_0} \prod_{k=1}^{n} \frac{q_{\rho(x_{k-1})\rho(x_k)}}{q_{x_kx_{k-1}}} = \pi_x \frac{q_{\rho}(x)\rho(j)}{q_{jx}}.$$

Hence, by Proposition 5, $X(t)$ is $\rho$-reversible. \hfill \Box

**Proposition 7.** Let $X(t)$ be an ergodic CTMC with state space $\mathcal{S}$ and infinitesimal generator matrix $Q$, let $\rho : \mathcal{S} \to \mathcal{S}$ be a renaming on $\mathcal{S}$, and $x_0, x_1, x_2, \ldots, x_n = x \in \mathcal{S}$ be a finite sequence of states. If $X(t)$ is $\rho$-reversible then for all $x \in \mathcal{S}$,

$$\pi_x = C_{x_0} \prod_{k=1}^{n} \frac{q_{\rho(x_{k-1})\rho(x_k)}}{q_{x_kx_{k-1}}} \quad (9)$$

where $x_0 \in \mathcal{S}$ is an arbitrary reference state and $C_{x_0} \in \mathbb{R}^+.$

**Proof.** See the proof of Proposition 6. \hfill \Box

### 2.6 Discrete time Markov chains

We review some notions of the theory of discrete time Markov chains [14] that will be required in the following sections. A stochastic process $X(t)$ in discrete time is a sequence of random variables taking values in a countable state space $\mathcal{S}$ for $t \in \mathbb{Z}$. If $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same distribution as $(X(t_1 + \tau), X(t_2 + \tau), \ldots, X(t_n + \tau))$ for all $t_1, t_2, \ldots, t_n, \tau \in \mathbb{Z}$ then the process $X(t)$ is said to be stationary. A
Discrete Time Markov Chain (DTMC) is a stochastic process $X(t)$ in discrete time with $t \in \mathbb{Z}$, such that for all $t_1 < t_2 < \ldots < t_n < t_{n+1}$ the joint distribution of $(X(t_1), X(t_2), \ldots, X(t_n), X(t_{n+1}))$ is such that

$$P(X(t_{n+1}) = x_{n+1} \mid X(t_1) = x_1, X(t_2) = x_2, \ldots, X(t_n) = x_n) = P(X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n).$$

The value $P(X(t + 1) = b \mid X(t) = a) = p_{ab}$ denotes the probability that the chain, whenever in state $a$, makes a transition to state $b$ and is referred to as one-step transition probability. The square matrix $P = (p_{ab})_{a,b \in S}$ is called one-step transition matrix. Since the chain must move to one of the states $b \in S$ when leaving state $a$, each row sums to one: for each $a \in S$

$$\sum_{b \in S} p_{ab} = 1.$$

An ergodic DTMC admits a unique stationary distribution, that is the unique vector $\pi$ of positive numbers $\pi_x$ with $x \in S$ such that:

$$\pi = \pi P \quad \text{and} \quad \sum_{x \in S} \pi_x = 1.$$

### 2.7 From CTMC to DTMC

Let $X(t)$ be a CTMC such that there exists a positive real number $\gamma$ for which $q_{ab} \leq \gamma$ for all $a, b \in S$. In [16, 2] the DTMC $X^U(t)$ with the same state space of $X(t)$ by uniformisation is defined, consider $v = \max \{q_a, a \in S\}$ and define the transition probabilities of $X^U(t)$ as follows:

$$p_{ab} = \begin{cases} 
q_{ab}v & \text{if } a \neq b \\
1 - q_a v & \text{if } a = b.
\end{cases}$$

The steady-state distribution of $X^U(t)$ and $X(t)$ are the same and it will be more convenient to study the uniformised chain rather than
the one at continuous time. From the prospective of \( \rho \)-reversibility, the following result holds:

**Proposition 8.** Let \( X(t) \) be an ergodic CTMC and let \( X^U(t) \) be a DTMC obtained by uniformisation. Then \( X(t) \) is \( \rho \)-reversible if and only if \( X^U(t) \) is \( \rho \)-reversible.

*Proof.* If \( X(t) \) is \( \rho \)-reversible then its stationary solution satisfies the detailed balance equation according to Proposition 5 for all state \( i \neq j \). By definition of uniformisation, also the DTMC satisfies the corresponding equations for \( i \neq j \). However, in the DTMC we have introduced new transitions, namely the self-loops, and we have to check that for each state we have: \( \pi_i p_{ii} = \pi_i p_{\rho(i)\rho(i)} \), which is true if and only if \( p_{\rho(i)\rho(i)} = p_{ii} \). This is true since, by Proposition 5, \( i \) and \( \rho(i) \) have the same residence time distribution.

If \( X^U(t) \) is \( \rho \)-reversible then the detailed balance equations of Proposition 5 hold for all the state transitions \( i \rightarrow j \). If \( i \neq j \), clearly the detailed balance equations hold also for the CTMC. It remains to prove that the residence time distribution for state \( i \) and \( \rho(i) \) are the same. Indeed, let us take the transition \( i \rightarrow i \) in the DTMC and observe that the corresponding detailed balance equation implies \( p_{ii} = p_{\rho(i)\rho(i)} \) that, by the definition of uniformisation, implies \( q_i = q_{\rho(i)} \) in \( X(t) \). \( \square \)
3 Properties of $\rho$-reversible Markov chains

In this section we introduce the properties of $\rho$-reversible Markov chains on which we will base our algorithm for the computation of all the state space renaming functions of a $\rho$-reversible CTMC. The following theory may be formulated also for discrete time Markov chains (DTMC).

**Definition 2.** *(One step transition rate matrix) Let $X(t)$ be an ergodic Markov process with state space $S$ and infinitesimal generator matrix $Q$. We define the one step transition rate matrix $P = (p_{ab})$ where $p_{aa} = 0$ for all $a \in S$ and $p_{ab} = q_{ab}$ with $a \neq b$ for all $a, b \in S$.*

Let $X(t)$ be a $\rho$-reversible ergodic Markov process taking values in a countable state space $S = \{1, 2, ..., n\}$ with $n = |S|$ and one step transition rate matrix $P$, let $X^R(t)$ be the reversed process for $X(t)$. Let $\rho : S \to S$ be a renaming function on $S$ such that $X(t)$ and $\rho(X^R(t))$ are stochastically identical. From Proposition 6 $\forall x_1, \ldots, x_n \in S$ we can write:

$$p_{x_1 x_2} p_{x_2 x_3} \cdots p_{x_n x_1} = p_{\rho(x_1) \rho(x_2)} p_{\rho(x_2) \rho(x_3)} \cdots p_{\rho(x_{n-1}) \rho(x_n)}$$

which can be written as:

$$p_{x_1 x_2} p_{x_2 x_3} \cdots p_{x_n x_1} = p_{\rho(x_1) \rho(x_2)}^T p_{\rho(x_2) \rho(x_3)}^T \cdots p_{\rho(x_{n-1}) \rho(x_n)}^T$$

\[(10)\]

**Definition 3.** We define the matrix $P^\alpha = (p_{ab}^\alpha)$ and $P^\beta = (p_{ab}^\beta)$, where:

$$p_{ab}^\alpha = \begin{cases} -\ln\left(\frac{q_{ab}}{v}\right) & \text{if } a \neq b \\ -\ln\left(1 - \frac{q_b}{v}\right) & \text{if } a = b \end{cases}$$

$$p_{ab}^\beta = \begin{cases} -\ln\left(\frac{q_{ba}}{v}\right) & \text{if } a \neq b \\ -\ln\left(1 - \frac{q_a}{v}\right) & \text{if } a = b \end{cases}$$

with $v = \max\{q_a, a \in S\}$. 

17
Now we can rewrite Equation (10) as follow:

\[ p_{x_1 x_2}^\alpha + p_{x_2 x_3}^\alpha + \cdots + p_{x_n x_1}^\alpha + p_{x_1}^\alpha = p_{\rho(x_1)\rho(x_2)}^\beta + p_{\rho(x_2)\rho(x_3)}^\beta + \cdots + p_{\rho(x_{n-1})\rho(x_n)}^\beta + p_{\rho(x_n)\rho(x_1)}^\beta. \]  

(11)

By Proposition 8 and the rules of algebra, Equation 10 holds if and only if Equation 11 holds.

**Definition 4.** *(In-degree function)* Let \( X(t) \) be a CTMC with state space \( S \) and one step transition rate matrix \( \mathcal{P} \), we define the out-degree function

\[ \sigma_{\text{in}} : \{\mathcal{P}\} \times S \rightarrow \mathbb{N} \]

as follow:

\[ \sigma_{\text{in}}(\mathcal{P}, a) = \sum_{b \in S | p_{ab} > 0} 1. \]

The function \( \sigma_{\text{in}} \) represent the in-degree of a state of a CTMC.

**Definition 5.** *(Out-degree function)* Let \( X(t) \) be an ergodic CTMC with state space \( S \) and one step transition rate matrix \( \mathcal{P} \), we define the out-degree function

\[ \sigma_{\text{out}} : \{\mathcal{P}\} \times S \rightarrow \mathbb{N} \]

as follow:

\[ \sigma_{\text{out}}(\mathcal{P}, a) = \sum_{b \in S | p_{ab} > 0} 1. \]

The function \( \sigma_{\text{out}} \) represent the out-degree of a state of a CTMC.

**Proposition 9.** *(Topology)* Let \( X(t) \) be an ergodic Markov process with state space \( S \) and one step transition rate matrix \( \mathcal{P} \), and let \( \rho : S \rightarrow S \) be a renaming on \( S \). If \( X(t) \) is \( \rho \)-reversible then

\[ \forall x \in S : \sigma_{\text{in}}(P^\alpha, x) = \sigma_{\text{in}}(P^\beta, \rho(x)) \land \sigma_{\text{out}}(P^\alpha, x) = \sigma_{\text{out}}(P^\beta, \rho(x)). \]

This is a necessary condition for the \( \rho \)-reversibility of a Markov process.
Proof. Since by hypothesis \(X(t)\) is \(\rho\)-reversible, we know from Definition 1 that \(X(t)\) and \(\rho(X^R(t))\) are stochastically identical. Hence the in-degree and out-degree of \(x\) and \(\rho(x)\) must be identical for all \(x \in S\).

The Floyd-Warshall algorithm proposed in [17] is an algorithm for the computation of the shortest paths between all pairs of nodes of a weighted graph. The algorithm can work with positive or negative edge weights but requires that there are no negative cycles in the graph. The Floyd-Warshall algorithm has a time complexity of \(\Theta(n^3)\) where \(n\) is the number of the nodes in the graph. The algorithm takes as input the adjacency matrix representing the directed graph and returns as output a matrix \(M = (m_{xy})\) where \(m_{xy}\) contains the value of the shortest path between the node \(x\) and \(y\) in the graph, a value of zero means that there is no path going from \(x\) to \(y\). We are going to use the application of Floyd-Warshall algorithm in the following definition.

**Definition 6.** We define \(\Delta^\alpha = \text{Floyd-Warshall}(P^\alpha)\) and \(\Delta^\beta = \text{Floyd-Warshall}(P^\beta)\) as the matrices representing the shortest path between all pairs of vertices, calculated using the Floyd-Warshall algorithm proposed in [17] applied to the adjacency matrix \(P^\alpha\) for \(\Delta^\alpha\) and \(P^\beta\) for \(\Delta^\beta\). More formally:

\[
\Delta^\alpha_{ab} = p^\alpha_{ai_1} + p^\alpha_{i_1i_2} + \ldots + p^\alpha_{i_{n-1}b}
\]

with \(a \rightarrow i_1 \rightarrow i_2 \rightarrow \ldots \rightarrow i_{n-1} \rightarrow b\) representing the shortest path between \(a\) and \(b\) computed by the Floyd-Warshall algorithm applied on the matrix \(P^\alpha\) and:

\[
\Delta^\beta_{ab} = p^\beta_{aj_1} + p^\beta_{j_1j_2} + \ldots + p^\beta_{j_{n-1}b}
\]

with \(a \rightarrow j_1 \rightarrow j_2 \rightarrow \ldots \rightarrow j_{n-1} \rightarrow b\) representing the shortest path between \(a\) and \(b\) computed by the Floyd-Warshall algorithm applied on the matrix \(P^\beta\) for all \(a, b \in S\).
We give an example of the computation of \( P^\alpha, P^\beta, \Delta^\alpha \) and \( \Delta^\beta \); values are approximated for readability.

\[
P = \begin{bmatrix}
0 & 0 & 0 & 0.2 & 0.8 \\
0.5 & 0 & 0 & 0.5 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0.7 & 0.3 & 0 & 0 \\
1 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
P^\alpha = (P^\beta)^T = \begin{bmatrix}
\infty & \infty & \infty & 1.609 & 0.223 \\
0.693 & \infty & \infty & 0.693 & \infty \\
\infty & 0 & \infty & \infty & \infty \\
\infty & 0.357 & 1.2 & \infty & \infty \\
0 & \infty & \infty & \infty & \infty
\end{bmatrix}
\]

\[
\Delta^\alpha = (\Delta^\beta)^T = \begin{bmatrix}
0.223 & 1.966 & 2.813 & 1.609 & 0.223 \\
0.693 & 1.05 & 1.897 & 0.693 & 0.916 \\
0.693 & 0 & 1.897 & 0.693 & 0.916 \\
1.05 & 0.357 & 1.2 & 1.05 & 1.273 \\
0 & 1.966 & 2.813 & 1.609 & 0.223
\end{bmatrix}
\]

**Proposition 10.** *(Cycles)* Let \( X(t) \) be an ergodic Markov process with state space \( S \) and one step transition rate matrix \( P \), and let \( \rho : S \to S \) be a renaming on \( S \). Let \( \Delta^\alpha = \text{Floyd-Wharshall}(P^\alpha) \) and \( \Delta^\beta = \text{Floyd-Wharshall}(P^\beta) \) as defined before. If \( X(t) \) is \( \rho \)-reversible then

\[
\forall a, b \in S : \Delta^\alpha_{ab} + \Delta^\alpha_{ba} = \Delta^\beta_{\rho(a)\rho(b)} + \Delta^\beta_{\rho(b)\rho(a)}.
\]

This is a necessary condition for the \( \rho \)-reversibility of a Markov process.
Proof. Since by hypothesis $X(t)$ is $\rho$-reversible, we know from Proposition 6 that for every finite sequence of states $x_1, x_2, \ldots, x_{n-1}, x_n \in S$, we have:

$$p_{x_1 x_2} p_{x_2 x_3} \cdots p_{x_n x_1} = p_{\rho(x_1)\rho(x_n)} p_{\rho(x_n)\rho(x_{n-1})} \cdots p_{\rho(x_3)\rho(x_2)} p_{\rho(x_2)\rho(x_1)}.$$ 

We also know by hypothesis that $X(t)$ is ergodic, so it’s irreducible, this means that for every $a, b \in S$ there is at least one finite chain of one-step transition starting from $a$ that reaches $b$:

$$a = i_1 \rightarrow i_2 \rightarrow \ldots \rightarrow i_{n-1} \rightarrow i_n = b$$

and there is at least one finite chain of one-step transitions starting from $b$ that reaches $a$:

$$b = i_n \rightarrow i_{n+1} \rightarrow \ldots \rightarrow i_{n+m-1} \rightarrow i_1 = a$$

We know that holds:

$$p_{ai_2p_{i_2i_3} \cdots p_{i_{n-1}b}p_{i_{n+1}} \cdots p_{i_{m-1}a}} =$$

$$p_{\rho(a)\rho(i_{m-1})} \cdots p_{\rho(i_{n+1})\rho(b)} p_{\rho(b)\rho(i_{n-1})} \cdots p_{\rho(i_3)\rho(i_2)} p_{\rho(i_2)\rho(a)}$$

which can be rewritten as:

$$p_{a_{i_2}^\alpha + p_{i_2i_3}^\alpha + \cdots + p_{i_{n-1}}^\alpha b + p_{i_{n+1}}^\alpha + \cdots + p_{i_{m-1}}^\alpha a} =$$

$$p_{\rho(a)\rho(i_2)} + p_{\rho(i_2)\rho(i_3)} + \cdots + p_{\rho(i_{n-1})\rho(b)} + p_{\rho(b)\rho(i_{n+1})} + \cdots + p_{\rho(i_{n+m-1})\rho(a)}$$

Equation 12 holds for every $a, b \in S$ and for every finite sequence starting from $a$, passing through $b$ and going back to $a$.

Now consider the finite chain, going from $a$ to $b$ and back to $a$, with the smallest sum of $p_\alpha$: 

$$a = i_1 \rightarrow i_2 \rightarrow \ldots \rightarrow i_{n-1} \rightarrow i_n = b \rightarrow i_{n+1} \rightarrow \ldots \rightarrow i_{n+m-1} \rightarrow i_1 = a.$$
By Definition 6 we know that:

$$\Delta_\alpha^{ab} + \Delta_\alpha^{ba} = p_{ai_2}^\alpha + p_{i_2i_3}^\alpha + \ldots + p_{i_{n-1}b}^\alpha + p_{bi_{n+1}}^\alpha + \ldots + p_{i_m-1a}^\alpha \quad (13)$$

Since by hypothesis $X(t)$ and $\rho(X(t))$ are stochastically identical we know that by Definition 3 and Definition 6 the chain

$$\rho(a) \rightarrow \rho(i_{n+m-1}) \rightarrow \ldots \rightarrow \rho(i_{n+1}) \rightarrow \rho(b) \rightarrow \rho(i_{n-1}) \rightarrow \ldots \rightarrow \rho(i_2) \rightarrow a$$

must have the smallest sum of $p^\beta$. Hence we can write:

$$\Delta_\beta^{\rho(a)\rho(b)} + \Delta_\beta^{\rho(b)\rho(a)} =$$

$$p_{\rho(a)\rho(i_2)}^\beta + p_{\rho(i_2)\rho(i_3)}^\beta + \ldots + p_{\rho(i_{n-1})\rho(b)}^\beta + p_{\rho(b)\rho(i_{n+1})}^\beta + \ldots + p_{\rho(i_{n+m-1})\rho(a)}^\beta. \quad (14)$$

By Equation 12, Equation 13 and Equation 14 holds that:

$$\Delta_\alpha^{ab} + \Delta_\alpha^{ba} = \Delta_\beta^{\rho(a)\rho(b)} + \Delta_\beta^{\rho(b)\rho(a)}$$

for all $a, b \in S$. \qed
4 Computing a $\rho$-reversible Markov chain’s renaming functions

In this section we introduce an algorithm for the computation of all the state space renaming functions of a $\rho$-reversible CTMC.

The algorithm takes in input the state space $\mathcal{S}$ and the infinitesimal generator matrix $Q$ of the Markov chain. The output is the set $\mathcal{F}$ which contains all the permutations of the state space $\mathcal{S}$ that constitute valid renaming functions for the given chain.
Algorithm 1: ComputeRenamings(S, Q)

\{Initialize P, P^\alpha, P^\beta, \Delta^\alpha and \Delta^\beta\}

for i ∈ S do
    for j ∈ S do
        if \(\sigma_{in}(P^\alpha, i) = \sigma_{in}(P^\beta, j) \land \sigma_{out}(P^\alpha, i) = \sigma_{out}(P^\beta, j)\) then
            \(I_i \leftarrow j\)
        end
    end
for i ∈ S do
    for \(\rho(i) \in I_i\) do
        if \(\forall j \in S \exists \rho(j) \in I_j\) s.t.
            \(\Delta_i^\alpha + \Delta_j^\alpha = \Delta_{\rho(i)\rho(j)}^\beta + \Delta_{\rho(j)\rho(i)}^\beta \land \rho(i) = \rho(j) \rightarrow i = j\)
        then
            \(R_i \leftarrow \rho(i)\)
        end
    end
    \(I_i = R_i\)
end
for \(\rho \in R_1 \times R_2 \times \ldots \times R_n\) do
    if NaiveReversibleUpTo(S, P, \rho) then
        \(\mathcal{F} \leftarrow \rho\)
    end
end
return \(\mathcal{F}\)
**Algorithm 2:** NaiveReversibleUpTo($S, P, \rho$)

```
for $u \in S$ do
    $color[u] = white$
    $x[u] = +\infty$
end
$x[1] = 1$
return DFS-NaiveReversibleUpTo($S, P, 1, \rho$)
```

**Algorithm 3:** DFS-NaiveReversibleUpTo($S, P, u, \rho$)

```
bool = true
for $bool \land v \in Adj[u]$ do
    if $(P[\rho[v], \rho[u]] = 0)$ then
        $bool = false$
    end
    if $(color[v] \neq white \land x[u]P[u, v] \neq x[v]P[\rho[v], \rho[u])$ then
        $bool = false$
    end
    if $(color[v] = white)$ then
        $color[v] = grey$
        $x[v] = x[u]P[u, v]P[\rho[v], \rho[u]]$
        $bool = bool \land DFS-NaiveReversibleUpTo(S, P, v, \rho)$
    end
end
return $bool$
```
Proposition 11. Given a stationary DTMC \((S, P)\) the Algorithm NaiveReversibleUpTo returns true if and only if \((S, P)\) is \(\rho\)-reversible.

NaiveReversibleUpTo\((S, P, \rho)\) always terminates in time \(O(n+m)\), where \(n\) is the number of states of \(S\) and \(m\) is the number of edges of the chain.

Proposition 12. If after the execution of the algorithm ComputeRenamings we have that \(\mathcal{F} = \emptyset\) then the chain is not \(\rho\)-reversible nor reversible.

Proof. The proof is trivial since the proposed algorithm rejects all the renaming that don’t satisfy the conditions showed in Proposition 9 and 10 which are necessary conditions for a function to be a valid renaming for the chain. If \(\mathcal{F} = \emptyset\) then there are no valid renaming functions and the chain cannot be \(\rho\)-reversible nor reversible.

Proposition 13. If after the execution of the algorithm ComputeRenamings we have that \(\exists R_i = \emptyset, i \in S\) then we have \(\mathcal{F} = \emptyset\) and the chain is not \(\rho\)-reversible nor reversible.

Proof. As seen in the proof of Proposition 12 recalling that if \(\exists R_i = \emptyset, i \in S\) we cannot build any valid permutation function in the state space \(S\) and \(\mathcal{F} = \emptyset\).

4.1 Time complexity

The algorithm NaiveReversibleUpTo has a time complexity of \(O(n + m)\), since a CTMC is a directed graph we have that \(m_{\text{max}} = n(n - 1)\) thus the time complexity can be written as \(O(n) + O(n^2) = O(n^2)\).

Let \(\gamma\) be the number of possible renaming function that we can build from the sets \(R_x\) with \(x \in S\), in particular \(\gamma \leq \prod_{x \in S} r_i\) with \(r_x = |R_x|\) for all \(x \in S\). The time complexity of the algorithm ComputeRenamings is \(\Theta(n^4) + \Theta(\gamma n^2)\).
To be able to discover if a chain admits valid renaming functions we should check all the possible $\gamma = n!$, with $n = |S|$, permutations of the states. Even if the number of possible renaming functions is huge, only a few comply with the properties of a valid rename of the states. Applying our algorithm to continuous time Markov chains that model real systems, we’ve seen that the number of renaming functions, that must be checked, rapidly decrease and thus also the global complexity of the test is reduced. If applied to chains that represent real systems, we expect to have $\gamma \ll n!$ where $n = |S|$ and $n!$ is the number of all possible renaming functions obtained from the permutation of the states of $X(t)$. In particular, if applied to such chains, the complexity of the heuristic algorithm can often decrease to polynomial time.
5 Examples

In this section we show the application of our algorithm to some theoretical examples.

5.1 Example 1

Let $\mathcal{S} = \{0, 1, 2, 3, 4\}$ be the state space of the Markov chain showed in Figure 1. For convenience the chain has already been uniformised and Figure 1 shows the $X^U(t)$ obtained from $X(t)$. The execution of the algorithm $\text{ComputeRenamings}$ on the proposed chain gives us the following results:

\[
\mathcal{R}_0 = \{0\}
\]
\[
\mathcal{R}_1 = \{1, 3\}
\]
\[
\mathcal{R}_2 = \{2\}
\]
\[ \mathcal{R}_3 = \{1, 3\} \]
\[ \mathcal{R}_4 = \{4\} \]
\[ \mathcal{F} = \{[0, 3, 2, 1, 4]\} \]

### 5.2 Example 2

![Diagram of a \(\rho\)-reversible DTMC](image)

**Figure 2: A \(\rho\)-reversible DTMC**

Let \(\mathcal{S} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}\) be the state space of the Markov chain showed in Figure 2. For convenience the chain has already been uniformised and Figure 2 shows the \(X^U(t)\) obtained from \(X(t)\). The execution of the algorithm \textit{ComputeRenamings} on the proposed chain gives us the following results:

\[ \mathcal{R}_0 = \{1, 3\} \]
\[ \mathcal{R}_1 = \{0, 2\} \]
\[ \mathcal{R}_2 = \{1, 3\} \]
\[ \mathcal{R}_3 = \{0, 2\} \]
\[ \mathcal{R}_4 = \{5, 7\} \]
\[ \mathcal{R}_5 = \{4, 6\} \]
\[ \mathcal{R}_6 = \{5, 7\} \]
\[ \mathcal{R}_7 = \{4, 6\} \]
\[ \mathcal{R}_8 = \{8, 9\} \]
\[ \mathcal{R}_9 = \{9\} \]

\[ \mathcal{F} = \{[1, 2, 3, 0, 5, 6, 7, 4, 8, 9], [3, 0, 1, 2, 7, 4, 5, 6, 8, 9]\} \]

### 5.3 Example 3

![A \rho\text{-reversible Markov chain diagram}](image)

**Figure 3:** A $\rho$-reversible Markov chain
Let \( S = \{0, 1, 2, 3, 4, 5, 6, 7\} \) be the state space of the Markov chain showed in Figure 3. For convenience the chain has already been uniformised and Figure 3 shows the \( X_U(t) \) obtained from \( X(t) \). The execution of the algorithm \textit{ComputeRenamings} on the proposed chain gives us the following results:

\[
\mathcal{R}_0 = \{0, 2, 4, 6\} \\
\mathcal{R}_1 = \{1, 3, 5, 7\} \\
\mathcal{R}_2 = \{0, 2, 4, 6\} \\
\mathcal{R}_3 = \{1, 3, 5, 7\} \\
\mathcal{R}_4 = \{0, 2, 4, 6\} \\
\mathcal{R}_5 = \{1, 3, 5, 7\} \\
\mathcal{R}_6 = \{0, 2, 4, 6\} \\
\mathcal{R}_7 = \{1, 3, 5, 7\}
\]

\[
\mathcal{F} = \{[0, 1, 6, 7, 4, 5, 2, 3], [2, 3, 4, 5, 6, 7, 0, 1], [4, 5, 2, 3, 0, 1, 6, 7], [6, 7, 0, 1, 2, 3, 4, 5]\}
\]
5.4 Example 4

Figure 4: A $\rho$-reversible Markov chain

Let $\mathcal{S} = \{0,1,2,3,4,5,6,7\}$ be the state space of the Markov chain showed in Figure 4. For convenience the chain has already been uniformised and Figure 4 shows the $X^U(t)$ obtained from $X(t)$. The execution of the algorithm $\text{ComputeRenamings}$ on the proposed chain gives us the following results:

\[ \mathcal{R}_0 = \{0,1,2,3\} \]
\[ \mathcal{R}_1 = \{0,1,2,3\} \]
\[ \mathcal{R}_2 = \{0,1,2,3\} \]
\[ \mathcal{R}_3 = \{0,1,2,3\} \]
\[ \mathcal{R}_4 = \{4,5,6,7\} \]
\[ \mathcal{R}_5 = \{4,5,6,7\} \]
\( \mathcal{R}_6 = \{4, 5, 6, 7\} \)
\( \mathcal{R}_7 = \{4, 5, 6, 7\} \)

\[ \mathcal{F} = \{[1, 0, 3, 2, 6, 5, 4, 7], [1, 2, 3, 0, 5, 6, 7, 4], [3, 0, 1, 2, 7, 4, 5, 6], [3, 2, 1, 0, 4, 7, 6, 5]\} \]
6 The Fair Allocation Control Window protocol

In the following section we show how the proposed algorithm can be applied to a model of a real system. The algorithm is used to verify the dynamic reversibility of the Markov chain underlying the analytical model used for the performance evaluation of the FACW protocol [13].

6.1 The FACW protocol

The main idea of FACW is that data traffic in a WSN can be classified into a finite set of $M$ classes $\mathcal{K} = \{c_1, c_2, \ldots, c_M\}$. Each sensor maintains a control window of size $N$ in which the classes of the latest $N$ transmissions (listened or performed) are stored. In the window, at most $h_c$ entries of class $c$ can appear. We stress the fact that the window stores only the class identifier of a transmission and not the sent packet. So, if we assume a practical situation with 16 classes, each class can be encoded by 4 bits and hence a window can be stored in few bytes. In case the sensor generates a packet of class $c$ when in its window there are already $h_c$ entries of class $c$, the packet is rescheduled for transmission after a back-off time or is simply dropped. Otherwise, in case of generation of a class $c$ packet and the number of $c$-entries in the window is strictly lower than $h_c$, then the packet is sent and the window is updated according to a FIFO policy. It should be clear that larger window sizes imply a lower bandwidth usage, whereas lower values of $N$ make the transmission more aggressive. The role of $h_c$ is that of modeling class priority. Allowing more entries of a class $c$ in the control window reduces the probability of $c$-packet dropping/delaying and hence its priority is larger than that of a traffic class $d$ with $h_d < h_c$. The initialization of the window is arbitrary. If necessary, we can assume that there exists a class $c$ of data traffic (e.g., the packets used
for controlling the routing) that has $h_c = N$ and whose rate is slow. The presence of this class ensures that the starvation of all the other traffic classes never occurs because of the control window.

6.2 The stochastic model

We consider a set $\mathcal{K} = \{c_1, c_2, \ldots, c_M\}$ of $M$ distinct traffic classes and assume that each node maintains a window $\mathcal{W}$ of size $N$ storing the transmission classes of the most recent sensed data according to a FIFO policy. An arrival can be due to a sensor data harvesting or to a listening to another node transmission. We denote the state of the window by $\vec{x} = (x_1, x_2, \ldots, x_N)$, where $x_i \in \mathcal{K}$, and let $|\vec{x}|_c = \sum_{i=1}^{N} \delta_{x_i=c}$ be the total number of occurrences of class $c$ in $\mathcal{W}$. We assume that data of different traffic classes are generated according to independent Poisson processes whose rates $\lambda_c(j)$, with $c \in \mathcal{K}$ and $1 \leq j \leq N$, depend on the number of objects $j = |\vec{x}|_c$ of class $c$ that are present in the window. Clearly, the process $X(t)$ that describes the state of $\mathcal{W}$ is a homogeneous continuous time Markov chain (CTMC) with finite state space. In the window there can be at most $h_c$ objects of class $c$, with $c \in \mathcal{K}$. If $h_c = N$ then there is no constraint on the maximum number of objects of the same class in the window. Let $\vec{x} = (x_1, \ldots, x_N)$ be the state of the control window, then the transition rates in the CTMC infinitesimal generator are: for $\vec{x} \neq \vec{x}'$,

$$q(\vec{x}, \vec{x}') = \begin{cases} 
\lambda_c(|\vec{x}|_c) & \text{if } \vec{x}' = (c, x_1, \ldots, x_{N-1}) \text{ and } |\vec{x}|_c < h_c \\
0 & \text{otherwise.}
\end{cases}$$

6.3 Testing the FACW model

We show the application of our algorithm to a continuous time Markov chain that models the FACW protocol. Assume to have $N = 3$, $h_c = 2$
for all $c$ and three classes with rates $\lambda_1 = 1.0$, $\lambda_2 = 2.0$ and $\lambda_3 = 3.0$. The state space of the CTMC is the following:

$$S = \{112, 211, 121, 113, 311, 131, 221, 122, 212, 223, 322, 232, 331, 133, 313, 332, 233, 323, 123, 213, 231, 132, 312, 321\}.$$ 

For readability purposes we label each state as follows:

$$S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23\}.$$ 

The resulting set of renaming functions is:

$$F = \{[1, 0, 2, 4, 3, 5, 7, 6, 8, 10, 9, 11, 13, 12, 14, 16, 15, 17, 23, 22, 21, 20, 19, 18]\}$$ 

The chain admits only one renaming function. From the result we can see that the given CTMC is dynamically reversible.

### 6.4 Closed form stationary distribution

In [13] the authors derive the stationary distribution of process $X(t)$. The state space of $X(t)$ is $S = \{\vec{x} \in \mathcal{K}^N : |\vec{x}|_c \leq h_c \text{ for all } c \in \mathcal{K}\}$. Since the state space of $X(t)$ is finite and its transition graph is irreducible, the CTMC has a unique limiting distribution independent of its initial state.

**Proposition 14.** The stationary distribution $\pi(\vec{x})$ of $X(t)$ for the FIFO (First In First Out) policy is given by the following expression:

$$\pi(\vec{x}) = \frac{1}{G} \prod_{c \in \mathcal{K}} \prod_{i=0}^{|[\vec{x}]_c-1} \lambda_c(i), \quad (15)$$

where $G = \sum_{\vec{x} \in S} \prod_{c \in \mathcal{K}} \prod_{i=0}^{|[\vec{x}]_c-1} \lambda_c(i).$
Lemma 2. Let \( n = (n_{c_1}, n_{c_2}, \ldots, n_{c_M}) \) with \( 0 \leq n_c \leq h_c \) for all \( c \in \mathcal{K} \) and \( \sum_{c \in \mathcal{K}} n_c = N \). The stationary probability of observing the aggregated state with \( n_c \) elements of class \( c \) for all \( c \in \mathcal{K} \) is:

\[
\pi_A(n) = \frac{1}{G\left(n_{c_1}, n_{c_2}, \ldots, n_{c_M}\right)} \prod_{c \in \mathcal{K}} \prod_{i=0}^{n_c-1} \lambda_c(i),
\]

where \( n \) belongs to the set of aggregated states

\[
\mathcal{S}_{\mathcal{K},N} = \{ n : \sum_{c \in \mathcal{K}} n_c = N \text{ and } 0 \leq n_c \leq h_c \ \forall c \in \mathcal{K} \}.
\]

Lemma 3. The marginal stationary probability of observing exactly \( \delta \) objects of class \( d \in \mathcal{K} \) in the window, with \( 0 \leq \delta \leq h_d \), is:

\[
\pi_{\mathcal{K},N}^d(\delta) = \binom{N}{\delta} \left( \prod_{j=0}^{\delta-1} \lambda_d(j) \right) \frac{G_{\mathcal{K}\setminus\{d\},N-\delta}}{G_{\mathcal{K},N}}.
\]
6.5 Performance indexes

In [13] the authors introduce some performance indexes and show how to compute them efficiently.

**Definition 7** (Admission rate). The admission rate for a class \( c \in \mathcal{K} \) is the rate associated with the event of transition from a state \( \vec{x} \) with \(|\vec{x}|_c = 0\) to a state \( \vec{x}' \) with \(|\vec{x}'|_c = 1\) when the model is in steady-state. The global admission rate is the sum of the admission rates for each \( c \in \mathcal{K} \).

**Definition 8** (Rejection rate). The rejection rate for a traffic class \( c \in \mathcal{K} \) is the rate associated with the event of rejecting the arrival of class \( c \) because the number of objects of class \( c \) in the window is \( h_c \). The global rejection rate is the sum of the rejection rates for each traffic class \( c \in \mathcal{K} \).

**Proposition 15.** In steady-state, the admission rate for a traffic class \( d \in \mathcal{K} \) is:

\[
X^d_{\mathcal{K},N} = \lambda_d(0) \frac{G_{\mathcal{K}\setminus\{d\},N}}{G_{\mathcal{K},N}},
\]

and the global admission rate is:

\[
X_{\mathcal{K},N} = \sum_{c \in \mathcal{K}} \lambda_c(0) \frac{G_{\mathcal{K}\setminus\{c\},N}}{G_{\mathcal{K},N}},
\]

**Proposition 16.** In steady-state, the rejection rate for a traffic class \( d \in \mathcal{K} \) is:

\[
Y^d_{\mathcal{K},N} = \lambda_d(h_d) \binom{N}{h_d} \prod_{j=0}^{h_d-1} \lambda_d(j) \frac{G_{\mathcal{K}\setminus\{d\},N-h_d}}{G_{\mathcal{K},N}},
\]

and the global rejection rate is:

\[
Y_{\mathcal{K},N} = \sum_{c \in \mathcal{K}} \lambda_c(h_c) \binom{N}{h_c} \prod_{j=0}^{h_c-1} \lambda_c(j) \frac{G_{\mathcal{K}\setminus\{c\},N-h_c}}{G_{\mathcal{K},N}}.
\]
Lemma 4. In steady-state, the expected number of objects of class $d \in \mathcal{K}$ in the window is:

$$
\overline{N}_{K,N}^d = \sum_{\delta=1}^{h_d} \binom{N}{\delta} \frac{\prod_{j=0}^{\delta-1} G_{K \setminus \{d\},N-\delta}}{G_{K,N}}.
$$

(20)

Lemma 5. In steady-state the throughput for a traffic class $d \in \mathcal{K}$ is:

$$
\lambda_d^* = \sum_{\delta=0}^{h_d-1} \lambda_d(\delta) \binom{N}{\delta} \left( \prod_{j=0}^{\delta-1} \lambda_d(j) \right) \frac{G_{K \setminus \{d\},N-\delta}}{G_{K,N}}.
$$

(21)
7 Simulations

The results shown in [13] are based on the assumption that the WSN is a mesh network, i.e., every node is in the transmission range of every other node. However, in practice, this is hard to achieve especially in large scale WSNs. In this section we study the sensitivity of the analytical results under the hypothesis of dealing with a mesh network. We resort to a stochastic simulation to show that in a connected ad-hoc network the average performance indices are not very sensitive to this assumption and hence the model proposed in [13] can be used to estimate the system throughput and fairness quite accurately.

All the simulations, unless differently specified, consist of 30 independent experiments whose warm-up phase has been removed according to the Welch’s procedure [18]. The confidence intervals have a confidence level of 98%. All the proposed simulation are based on the assumption that we have 20 classes with the arrival rates showed in the following table.

<table>
<thead>
<tr>
<th>Class $c$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_c$ ($S_1$)</td>
<td>1.00</td>
<td>1.30</td>
<td>1.50</td>
<td>1.80</td>
<td>3.80</td>
<td>1.20</td>
<td>1.50</td>
<td>1.72</td>
<td>1.12</td>
<td>8.00</td>
</tr>
<tr>
<td>$\lambda_c$ ($S_2$)</td>
<td>1.20</td>
<td>2.30</td>
<td>1.50</td>
<td>2.00</td>
<td>3.80</td>
<td>2.40</td>
<td>2.20</td>
<td>3.30</td>
<td>2.62</td>
<td>3.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class $c$</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_c$ ($S_1$)</td>
<td>1.00</td>
<td>1.30</td>
<td>1.35</td>
<td>6.78</td>
<td>4.10</td>
<td>1.20</td>
<td>1.66</td>
<td>1.70</td>
<td>1.44</td>
<td>20.0</td>
</tr>
<tr>
<td>$\lambda_c$ ($S_2$)</td>
<td>3.21</td>
<td>2.25</td>
<td>4.35</td>
<td>5.00</td>
<td>4.10</td>
<td>1.64</td>
<td>1.66</td>
<td>2.70</td>
<td>2.44</td>
<td>6.78</td>
</tr>
</tbody>
</table>

Table 1: Arrival rates for scenarios S1 and S2

7.1 Validation of the simulator

The first step consists in validating the simulator. To this aim, we simulate a mesh network that satisfies all the assumptions of the model proposed in [13]. Table 2 shows that the comparison of the results
given by the analytical model and the estimates of the simulation. We observe that the analytical values always fall in the confidence intervals whose relative error is very small. Thus, we consider the simulator validated.

<table>
<thead>
<tr>
<th>$h_c$</th>
<th>$N$</th>
<th>$\lambda^*$</th>
<th>$\lambda^s$</th>
<th>$\lambda^s_{lower}$</th>
<th>$\lambda^s_{upper}$</th>
<th>$\Phi^*$</th>
<th>$\Phi^s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>24.2832</td>
<td>24.2814</td>
<td>24.2618</td>
<td>24.3011</td>
<td>1.2786</td>
<td>1.2788</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>24.5006</td>
<td>24.5021</td>
<td>24.4861</td>
<td>24.5181</td>
<td>0.9897</td>
<td>0.9896</td>
</tr>
<tr>
<td>3</td>
<td>38</td>
<td>24.1746</td>
<td>24.1746</td>
<td>24.1534</td>
<td>24.1958</td>
<td>0.8513</td>
<td>0.8513</td>
</tr>
<tr>
<td>4</td>
<td>54</td>
<td>24.5124</td>
<td>24.5141</td>
<td>24.4959</td>
<td>24.5323</td>
<td>0.7537</td>
<td>0.7536</td>
</tr>
<tr>
<td>5</td>
<td>70</td>
<td>24.9674</td>
<td>24.9649</td>
<td>24.9463</td>
<td>24.9835</td>
<td>0.6807</td>
<td>0.6808</td>
</tr>
<tr>
<td>6</td>
<td>87</td>
<td>24.9928</td>
<td>24.9909</td>
<td>24.9748</td>
<td>25.0071</td>
<td>0.6249</td>
<td>0.6250</td>
</tr>
<tr>
<td>7</td>
<td>105</td>
<td>24.7372</td>
<td>24.7368</td>
<td>24.7173</td>
<td>24.7563</td>
<td>0.5800</td>
<td>0.5800</td>
</tr>
<tr>
<td>8</td>
<td>123</td>
<td>24.6059</td>
<td>24.6069</td>
<td>24.5895</td>
<td>24.6242</td>
<td>0.5430</td>
<td>0.5430</td>
</tr>
<tr>
<td>9</td>
<td>140</td>
<td>24.8422</td>
<td>24.8413</td>
<td>24.8214</td>
<td>24.8612</td>
<td>0.5117</td>
<td>0.5118</td>
</tr>
</tbody>
</table>

Table 2: Validation of the simulator for a mesh network. $N$ identical nodes are deployed and all the classes have the same $h_c$. The packet generation rate per class are shown in Table 1 (Scenario 1), the throughput and the fairness index obtained by the model are $\lambda^*$ and $\Phi^*$, respectively, and those estimated by the simulation are $\lambda^s$ and $\Phi^s$, respectively.

### 7.2 Simulation of general (non-mesh) networks

We now consider a WSN consisting of 300 sensors deployed in an area of $1000 \text{m} \times 1000 \text{m}$ whose transmission range is $100 \text{m}$. The location of the sensors is random with uniform distribution and we assume that the network is connected, i.e., there is at least one single- or multi-hop route connecting each pair of motes of the net. In this experiment, the transmission rates of the motes for each class of data
are sampled from exponential distributions whose means are the values
given in the description of Scenario 1 (see Table 1). The throughput
and the fairness index are shown in Tables 3. We observe that the
analytical model provides a high level of accuracy for the estimation of
the performance indices in this case, with a relative error lower than
2%, even if the packet generation rates for the same class are different
for the various motes. We conclude that if the WSN is sufficiently
dense to be connected the total throughput of the proposed admission
control algorithm depends mainly on the first moment of the harvesting
times and is not very sensitive to their distributions.

<table>
<thead>
<tr>
<th>$h_c$</th>
<th>$N$</th>
<th>$\lambda^*$</th>
<th>$\lambda^s$</th>
<th>$\lambda^s_{lower}$</th>
<th>$\lambda^s_{upper}$</th>
<th>$\Phi^*$</th>
<th>$\Phi^s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>24.2832</td>
<td>24.1775</td>
<td>24.1587</td>
<td>24.1962</td>
<td>1.2786</td>
<td>1.3114</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>24.5006</td>
<td>24.2029</td>
<td>24.1822</td>
<td>24.2235</td>
<td>0.9897</td>
<td>1.0178</td>
</tr>
<tr>
<td>3</td>
<td>38</td>
<td>24.1746</td>
<td>23.9317</td>
<td>23.9085</td>
<td>23.9549</td>
<td>0.8513</td>
<td>0.8521</td>
</tr>
<tr>
<td>4</td>
<td>54</td>
<td>24.5124</td>
<td>24.5962</td>
<td>24.5727</td>
<td>24.6197</td>
<td>0.7537</td>
<td>0.7525</td>
</tr>
<tr>
<td>5</td>
<td>70</td>
<td>24.9674</td>
<td>25.0125</td>
<td>24.9970</td>
<td>25.0281</td>
<td>0.6807</td>
<td>0.6897</td>
</tr>
<tr>
<td>6</td>
<td>87</td>
<td>24.9928</td>
<td>24.8864</td>
<td>24.8680</td>
<td>24.9047</td>
<td>0.6249</td>
<td>0.6046</td>
</tr>
<tr>
<td>7</td>
<td>105</td>
<td>24.7372</td>
<td>24.7002</td>
<td>24.6768</td>
<td>24.7235</td>
<td>0.5800</td>
<td>0.5938</td>
</tr>
<tr>
<td>8</td>
<td>123</td>
<td>24.6059</td>
<td>24.6298</td>
<td>24.6109</td>
<td>24.6488</td>
<td>0.5430</td>
<td>0.5509</td>
</tr>
<tr>
<td>9</td>
<td>140</td>
<td>24.8422</td>
<td>25.0960</td>
<td>25.0738</td>
<td>25.1183</td>
<td>0.5117</td>
<td>0.5261</td>
</tr>
</tbody>
</table>

Table 3: Simulated results with confidence interval at 98% confidence
level.

Figures 5 and 6 (7 and 8) show the comparison of the simulation es-
timates with the analytical results for S1 (S2). The confidence intervals
are too small to be displayed in the figures. We can see that, despite
the strictest hypotheses required by the analytical model are violated,
the results that we can obtain by its analysis are very accurate. The
same conclusion holds for the analysis of the total throughput and the
throughput of the fastest and slowest classes as shown by Figures 9-20.
Simulated and analytical fairness index $S_1 \; h_c = 1$

Figure 5: Fairness index in Scenario 1 with $h_c = 1$.

Simulated and analytical fairness index $S_1 \; h_c = 2$

Figure 6: Fairness index in Scenario 1 with $h_c = 2$. 
Figure 7: Fairness index in Scenario 2 with $h_c = 1$.

Figure 8: Fairness index in Scenario 2 with $h_c = 2$. 
Simulated and analytical throughput $S_1 \ h_c = 1$

Figure 9: Total throughput of the slowest class in Scenario 1 with $h_c = 1$. 

Simulated and analytical throughput $S_1 \ h_c = 1$

Figure 10: Total throughput of the fastest class in Scenario 1 with $h_c = 1$. 

45
Simulated and analytical throughput $S_1$ $h_c = 2$

Figure 11: Total throughput of the slowest class in Scenario 1 with $h_c = 2$.

Simulated and analytical throughput $S_1$ $h_c = 2$

Figure 12: Total throughput of the fastest class in Scenario 1 with $h_c = 2$. 
Simulated and analytical throughput $S_2 h_c = 1$

\[ \lambda_{20}, h_c = 1 \quad \times \]
\[ \lambda_{20}, h_c = 1 \quad - \text{line} \]

Figure 13: Total throughput of the fastest class in Scenario 2 with $h_c = 1$.

Simulated and analytical throughput $S_2 h_c = 2$

\[ \lambda_{s1}, h_c = 2 \quad \times \]
\[ \lambda^{*}_{s1}, h_c = 2 \quad - \text{line} \]

Figure 14: Total throughput of the slowest class in Scenario 2 with $h_c = 2$. 
Simulated and analytical throughput $S_2 h_c = 2$

Figure 15: Total throughput of the fastest class in Scenario 2 with $h_c = 2$.

Simulated and analytical total throughput $S_1 h_c = 1$

Figure 16: Total throughput in Scenario 1 with $h_c = 1$. 
Figure 17: Total throughput in Scenario 1 with $h_c = 1$.

Figure 18: Total throughput in Scenario 1 with $h_c = 2$. 
Figure 19: Total throughput in Scenario 2 with $h_c = 1$.

Figure 20: Total throughput in Scenario 2 with $h_c = 2$. 
8 Conclusion

In this thesis we have proposed an heuristic algorithm that allows the computation of all the renaming functions of a continuous time Markov chain; the algorithm can also work with discrete time Markov chains with minor adjustments. With this algorithm we try to reduce the number of renaming functions that need to be tested, to do so we exploit: properties about the structure of the chain, rules of algebra on the logarithmic function applied to the Kolmogorov’s criterion and the application of the Floyd-Warshall algorithm. We have introduced necessary conditions for the $\rho$-reversibility of a Markov process and shown the application of our algorithm to verify the dynamic reversibility of a Markov chain underlying the FACW protocol. We have also showed a stochastic validation of the model proposed in [13], thus it can be used to estimate the system performance indeces quite accurately.

If applied to a CTMC that models a real system, the complexity of the heuristic algorithm can often decrease to polynomial time. Our algorithm can be used to test the reversibility, dynamic reversibility and $\rho$-reversibility in a single execution being a handy tool for the computation of all the renaming functions of a given chain.
9 Acknowledgments

I would like to acknowledge all the people that helped me during my journey at the university.

I owe my gratitude to Prof. Sabina Rossi, Prof. Andrea Marin, Prof. Andrea Albarelli and Prof. Carla Piazza for their support and help throughout the thesis work. I would like to thank my mother, my sister and my father for their support, they allowed me to accomplish all my achievements. I would like to show my gratitude to all of my friends, they helped and endured me when many other would have already given up.
References


