

Efficient Computation of Renaming Functions for ρ -reversible Discrete and Continuous Time Markov Chains

Matteo Sottana
University Ca' Foscari Venice
matteo.sottana@unive.it

Carla Piazza
University of Udine
carla.piazza@uniud.it

Andrea Albarelli
University Ca' Foscari Venice
albarelli@unive.it

ABSTRACT

With the introduction of ρ -reversibility, the basic notion of reversible Markov chain has been relaxed by allowing a wider range of scenarios. Specifically, the reversibility properties are not just sought on the chain itself, but also on all the possible topology-preserving renamings of its state space. Such renamings, called *Renaming Functions*, exhibit many interesting properties which can be exploited in different contexts. Unfortunately, finding a renaming function for a Markov chain is a very computationally intensive task. Using a naive approach it could require to check for all the possible state space permutations, which is unfeasible for all but the most trivial chains. As a matter of fact, we prove that the corresponding decision problem is polynomially equivalent to Graph Isomorphism. Nevertheless, we introduce an algorithm that, exploiting some necessary conditions for ρ -reversibility, is able to efficiently prune the search space and then verify the remaining renaming candidates. The correctness of the method is theoretically demonstrated and its practical effectiveness is shown over a significant set of discrete and continuous ρ -reversible Markov chains.

Keywords

Reversibility modulo Renaming, Discrete and Continuous Time Markov Chains, Algorithms

1. INTRODUCTION

Markov chains have been widely used to study the performance of computer systems and software architectures. In the past decades several formalisms have been developed with the goal of allowing a stochastic model to be specified in a compact way by using features such as composition and hierarchical approaches. Despite the availability of compact representations, a stochastic process does not necessarily admit an efficient analysis. For instance, 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 basis for modeling a wide range of real-world contexts, ranging from chemical reaction dynamics to economic models and computer systems. The analysis of the Markov process underlying this kind of systems, allows us to derive the performance indices of the model, and thus of the modeled system itself. These are often computed at the steady-state (if it exists), when the time elapsed from the initial instant tends to infinity.

The analysis of Markov chains, both at continuous and discrete time, can in many cases be simplified by applying techniques based upon: internal symmetries, for instance reversibility [6, 8]; state aggregation, such as lumpability [9, 5]; composition, like product forms [2, 3]. Most of such simplifications can be exploited in order to allow the numerical or analytical tractability of the computation of performance indices, related by the common concept of time-reversibility.

The time-reversibility of Markov stochastic processes has been introduced for the first time and applied to the analysis of stochastic networks and Markov chains by Kelly [6]. 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 have led to the characterization of product-form solutions for queuing models with underlying time-reversible Markov chains. Product-form solutions allow the study of the sub-components of a system, reducing the state space, and then obtain 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 system's sub-components, each considered in isolation and opportunely parametrized.

However, time-reversibility is a very restrictive condition since it requires the chain to exhibit definitely the same stochastic behavior presented by its reverse. Indeed, this is a quite rare event and it happens only for a narrow class of chains, thus limiting the aptness of simplification and analysis approaches. In order to broaden the scope, a more flexible definition, called ρ -reversibility, has been proposed in [10, 8]. A Markov process is said to be ρ -reversible if it is stochastically identical to its reversed process modulo a renaming of the state space. The availability of a *renaming function* making a chain ρ -reversible allows, for instance, the efficient computation of the stationary probabilities or to easily decompose a process to characterize them as a product form.

Contribution. The main contributions of this paper are two algorithms for testing the ρ -reversibility of both DTMCs and CTMCs. The first is a linear time procedure that can be used to test whether a given renaming is valid for a process. The second one exploits the first and solves the more general problem of finding all renaming functions of a process. Of course, its computation can be stopped as soon as the first valid renaming is produced. The problem—in its decision version—is equivalent to Graph Isomorphism, provided that the stationary distribution has been exactly computed. Despite this complexity result, our algorithm proves its effectiveness on both synthetic and real-world examples. The key ingredients of its efficiency are two necessary conditions that in polynomial time prune the set of possibly valid renamings. To the best of our knowledge there was no previous feasible approach to solve this problem and the naive check of all the possible maps would require a number of verification steps which is factorial with respect to the size of the state space.

Related work. In [6] the author deeply discusses the notion of time reversibility both for DTMCs and CTMCs, and several applications are illustrated. Dynamic reversibility is introduced in [14] as a tool for the study of physical systems such as the growth of two dimensional crystals. An exploration of the relations among different definitions of lumpability and the notion of time-reversed Markov chain can be found in [10]. The idea of ρ -reversibility is introduced for the first time in [8] and its applications in the embedded and uniformized chains of continuous time processes are discussed. In [7] 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 [9] the authors propose the idea of using a general permutation of states for comparing forward and reversed processes at continuous time. A review covering the main results regarding time-reversible Markov processes and a discussion about how to apply them to tackle the problem of the quantitative evaluation of reversible computations can be found in [1]. In [4] 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 [13]. Finally, in [12] the authors focus on the problem of defining quantitative stochastic models for concurrent and cooperating reversible computations.

Structure of the paper. The remainder of this paper is organized as follows. Section 2 briefly recalls the general notions about Markov chains and supplies the definition and notation for both reversibility and ρ -reversibility that will be used throughout the following sections. Section 3 discusses in depth the conditions for a Markov chain to be ρ -reversible at the basis of our algorithm and studies the complexity of the *Reversibility modulo Renaming* decision problem and its relation with Graph Isomorphism. Section 4 introduces the algorithm we propose for recovering all the feasible *renaming functions* mapping a Markov chain to its reversible isomorphic form. Its correctness and complexity are also discussed. In Section 5 the performance and effectiveness of this novel algorithm are demonstrated by applying it to both continuous and discrete Markov chains representing respectively synthetic examples and processes related to a real case study. Finally, Section 6 concludes the paper.

2. SETTING THE CONTEXT

In this section, we briefly recall some basic notions about discrete and continuous time Markov chain. For more details the reader should refer to [6].

2.1 Markov Chains

A *discrete time stochastic process* $X(t)$ is a sequence of random variables taking values in state space \mathcal{S} for $t \in \mathbb{Z}$. Similarly, a *continuous time stochastic process* $X(t)$ is a sequence of random variables taking values in state space \mathcal{S} for $t \in \mathbb{R}$. In this paper we will only deal with finite state spaces. A stochastic process is *stationary* if it has the same distribution for all time t . A *Markov Chain* is a stochastic process such that for all $t_1 < \dots < t_n < t_{n+1}$ it holds that

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

i.e., the past evolution does not influence the conditional probability of the future behaviour. If $X(t)$ is a discrete (continuous) time stochastic process, then the Markov Chain is a *Discrete Time Markov Chain (DTMC)* (*Continuous Time Markov Chain (CTMC)*, respectively). A Markov chain is said to be *time homogeneous* if the conditional probability $P(X(t + \tau) = j \mid X(t) = i)$ does not depend upon t .

In the case of time homogeneous DTMC for each $i, j \in \mathcal{S}$ we use the notation

$$P(X(t + 1) = j \mid X(t) = i) = p_{ij}$$

The value p_{ij} denotes the probability that the chain, whenever in state i , next makes a transition into state j , and is referred to as *(one-step) transition probability*. The square matrix $\mathbf{P} = (p_{ij})_{i,j \in \mathcal{S}}$ is called *(one-step) transition matrix*, and since when leaving state i the chain must move to one of the states $j \in \mathcal{S}$, each row sums to one (i.e., forms a probability distribution).

In the case of time homogeneous CTMC for each $i, j \in \mathcal{S}$ and for each $t \geq 0$ we use the notation

$$P(X(s + t) = j \mid X(s) = i) = p_{ij}(t)$$

This value represents the probability that the chain from state i after t time instants ends up in state j . Hence, we can consider the square matrices $\mathbf{P}(t) = (p_{ij}(t))_{i,j \in \mathcal{S}}$, where $\mathbf{P}(0) = Id$ is the identity matrix. It is possible to prove that

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{Q}\mathbf{P}(t), \text{ with } \mathbf{Q} = \frac{d\mathbf{P}(t)}{dt}(0) = (q_{ij})_{i,j \in \mathcal{S}}$$

\mathbf{Q} is the *infinitesimal generator*, all its rows sum to 0 and its diagonal elements are negative. Intuitively, q_{ij} with $i \neq j$ represents the transition rate from i to j . Given \mathbf{Q} the CTMC is completely defined, since $\mathbf{P}(t) = e^{t\mathbf{Q}}$.

A Markov chain is *irreducible* if every state in \mathcal{S} can be reached from every other state. A state in a Markov process is called *recurrent* if it is 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 chain is *ergodic* if it is irreducible and all its states are positive recurrent. An ergodic Markov chain has a unique *steady-state distribution* (*stationary distribution*), i.e., a distribution of probability $\pi = (\pi_i)_{i \in \mathcal{S}}$ which remains invariant with respect to time.

In the case of DTMC this means that

$$\pi = \pi \mathbf{P} \quad (1)$$

Similarly, in the case of CTMC the meaning is that $\pi = \pi \mathbf{P}(t)$, for all $t \geq 0$, which can be proven to be equivalent to

$$\pi \mathbf{Q} = 0 \quad (2)$$

Eq. (1) and (2) are called *Global Balance Equations (GBE)*.

In general, the knowledge of the steady-state for a process modeling a system is a key information in order to compute its performance indices, such as throughput, expected number of customers at a queue, admission rates and many others.

2.2 Reversibility

The most obvious method to obtain π is, of course, by solving algebraic problems (1) and (2). However, this approach could suffer from ill-conditioning or numerical instability, since it requires either an iterative computation or a possibly large number of substitutions. Luckily the analysis of ergodic Markov chains can be greatly simplified if the behavior of the chain remains the same when the direction of time is reversed.

DEFINITION 1 (REVERSIBILITY [6]). *A process $X(t)$ is reversible if $X(t)$ and $X^R(t) = X(\tau - t)$ are stochastically identical for all τ .*

It is easy to see that reversible processes are stationary. Moreover, if $X(t)$ is a Markov chain, then $X^R(t)$ is a Markov chain too. Hence, in the rest of this paper we will consider only ergodic stationary Markov chains.

A characterization of reversibility over (ergodic stationary) DTMC is given by the following result.

THEOREM 1 (DETAILED BALANCE [4]). *A DTMC $X(t)$ defined by \mathbf{P} is reversible iff there is a distribution of probability $\pi = (\pi_i)_{i \in \mathcal{S}}$ such that for each $i, j \in \mathcal{S}$ it holds that:*

$$\pi_i p_{ij} = \pi_j p_{ji} \quad (3)$$

The stationary distribution of both $X(t)$ and $X^R(t)$ is π .

The above theorem can be exploited to efficiently compute the stationary distribution of reversible processes. The same result holds for CTMCs replacing \mathbf{P} with \mathbf{Q} .

Unfortunately, reversibility is a strong requirement which is not usually satisfied by real systems. In order to relax the condition, still guaranteeing the efficient computation of the stationary distribution, the notion of reversibility modulo state renaming has been introduced in [8]. A *renaming function* ρ over the state space of a Markov process is a bijection on \mathcal{S} . For a Markov chain $X(t)$ with state space \mathcal{S} we denote by $\rho(X)(t)$ the same process where the state names are changed according to ρ .

DEFINITION 2 (ρ -REVERSIBILITY). *A Markov Chain $X(t)$ over \mathcal{S} is ρ -reversible if there exists a renaming ρ on \mathcal{S} such that $X(t)$ and $\rho(X^R)(t)$ are stochastically identical. In this case we say that $X(t)$ is ρ -reversible.*

THEOREM 2 (ρ -DETAILED BALANCE EQUATIONS[8]). *Let ρ be a renaming on \mathcal{S} . A DTMC $X(t)$ defined by \mathbf{P} is ρ -reversible iff there is a distribution of probability $\pi = (\pi_i)_{i \in \mathcal{S}}$ such that for all $i, j \in \mathcal{S}$ it holds that:*

$$\pi_i p_{ij} = \pi_j p_{\rho(j)\rho(i)} \quad (4)$$

The stationary distribution of $X(t)$ and $\rho(X^R)(t)$ is π and $\pi_i = \pi_{\rho(i)}$, for all $i \in \mathcal{S}$.

We will also exploit a characterization of ρ -reversibility which generalizes Kolmogorov's criterion.

THEOREM 3 (ρ -KOLMOGOROV'S CRITERION [8]). *Let ρ be a renaming on \mathcal{S} . A DTMC $X(t)$ defined by \mathbf{P} is ρ -reversible iff for every finite sequence $i_1, i_2, \dots, i_n \in \mathcal{S}$,*

$$p_{i_1 i_2} p_{i_2 i_3} \cdots p_{i_{n-1} i_n} p_{i_n i_1} = p_{\rho(i_1)\rho(i_n)} p_{\rho(i_n)\rho(i_{n-1})} \cdots p_{\rho(i_3)\rho(i_2)} p_{\rho(i_2)\rho(i_1)}. \quad (5)$$

The above theorems holds also for CTMCs replacing \mathbf{P} with \mathbf{Q} .

3. COMPUTATION OF ρ -REVERSIBILITY

In this section we study the complexity of the *Reversibility modulo Renaming* decision problem and its relations with graph isomorphism. We focus on DTMCs, but all the results can be restated on CTMCs.

3.1 Verifying a given Renaming ρ

Given a renaming ρ on \mathcal{S} , Equation (4) of Theorem 2 can be used to verify whether $X(t)$ is ρ -reversible and to simultaneously compute its stationary distribution π .

LEMMA 1. *Let ρ be a renaming on \mathcal{S} . A DTMC $X(t)$ defined by \mathbf{P} is ρ -reversible iff the following system in the variables Π has a solution with $\Pi_1 = 1$:*

$$\bigwedge_{i,j \in \mathcal{S}} \Pi_i p_{ij} = \Pi_j p_{\rho(j)\rho(i)} \quad (6)$$

PROOF. \Rightarrow If $X(t)$ is ρ -reversible then, by Theorem 2 System (6) has a solution π which is the stationary distribution of $X(t)$. Since $X(t)$ is ergodic its stationary distribution has $\pi_i > 0$ for all $i \in \mathcal{S}$. Hence, by multiplying π for the factor π_1^{-1} we get a solution of System (6) whose first component is 1.

\Leftarrow If System (6) has a solution $\pi^* = (\pi_i^*)_{i \in \mathcal{S}}$ whose first component is 1, then by multiplying such solution for the factor $1/\sum_{i \in \mathcal{S}} \pi_i^*$ we get a solution π which satisfies the conditions of Theorem 2. \square

Notice that, since $X(t)$ is ergodic, once the value of X_1 has been fixed it is immediate to determine whether System (6) has a solution or not. We replace the value of Π_1 in the system and we determine the value of Π_j for each j such that $p_{1j} \neq 0$. We proceed by replacing and computing until either we find a contradiction or we have a solution. Moreover, the following lemma proves that it is sufficient to consider the equations whose left hand side is not null (i.e., $p_{ij} \neq 0$).

LEMMA 2. *Let ρ be a renaming on \mathcal{S} . Let $X(t)$ be a DTMC defined by \mathbf{P} . System (6) has a solution with $\Pi_1 = 1$ iff the following system has a solution with $\Pi_1 = 1$:*

$$\bigwedge_{i,j \in \mathcal{S} \text{ } p_{ij} \neq 0} \Pi_i p_{ij} = \Pi_j p_{\rho(j)\rho(i)} \quad (7)$$

PROOF. \Rightarrow It is trivial, since System (7) is a subset of System (6).

\Leftarrow) Let π^* be a solution of System (7) with $\pi_1^* = 1$. We have to prove that π^* is a solution of System (6). First we notice that since $X(t)$ is ergodic, π^* has only strictly positive components. The equations of System (6) with $p_{ij} \neq 0$ are also in System (7), so they are satisfied. We have to prove that if $p_{ij} = 0$, then also $p_{\rho(j)\rho(i)} = 0$, i.e., the equations of System (6) that are not in System (7) are trivially satisfied. Let us assume by contradiction that $p_{\rho(j)\rho(i)} \neq 0$. Then, since $\pi_{\rho(j)}^* \neq 0$, it has to be $p_{\rho^2(i)\rho^2(j)} \neq 0$. More in general this implies that $p_{\rho^{2k+1}(j)\rho^{2k+1}(i)} \neq 0$ and $p_{\rho^{2k}(i)\rho^{2k}(j)} \neq 0$ for all $k \geq 0$. Since ρ is a bijection over a finite set, there exists m and n such that $\rho^m(i) = i$ and $\rho^n(j) = j$. Hence, $\rho^{2*m*n}(i) = i$ and $\rho^{2*m*n}(j) = j$. So we have $p_{\rho^{2*m*n}(i)\rho^{2*m*n}(j)} = p_{ij} \neq 0$ which is a contradiction. \square

The above lemma can be exploited to define Algorithm (1) for verifying ρ -reversibility. The algorithm exploits a Depth First Search (DFS) on the labeled graph induced by \mathbf{P} on \mathcal{S} . It starts from the first element of \mathcal{S} initializing $\Pi[1]$ to 1 and it proceeds initializing all the $\Pi[j]$'s thanks to the equations of System (7). If the process is ρ -reversible, then it returns *true* and at the end of the computation the array Π contains the stationary distribution of the process. Otherwise an equation which cannot be satisfied is found during the computation and *false* is returned.

Algorithm 1: ReversibleUpTo($\mathcal{S}, \mathbf{P}, \rho$)

```

for  $i \in \mathcal{S}$  do
  |  $color[i] = white$ 
  |  $\Pi[i] = +\infty$ 
end
 $\Pi[1] = 1$ 
return DFS-ReversibleUpTo( $\mathcal{S}, \mathbf{P}, 1, \rho$ )

```

Algorithm 2: DFS-ReversibleUpTo($\mathcal{S}, \mathbf{P}, i, \rho$)

```

 $bool = true$ 
for  $bool \wedge p[i, j] \neq 0$  do
  | if  $(p[\rho[j], \rho[i]] = 0)$  then
  | |  $bool = false$ 
  | end
  | if  $(color[j] \neq white \wedge \Pi[i]p[i, j] \neq \Pi[j]p[\rho[j], \rho[i]])$ 
  | | then
  | | |  $bool = false$ 
  | | end
  | | if  $(color[j] = white)$  then
  | | |  $color[j] = grey$ 
  | | |  $\Pi[j] = \Pi[i] \frac{p[i, j]}{p[\rho[j], \rho[i]]}$ 
  | | |  $bool = bool \wedge \text{DFS-ReversibleUpTo}(\mathcal{S}, \mathbf{P}, j, \rho)$ 
  | | end
end
return  $bool$ 

```

THEOREM 4 (CORRECTNESS AND COMPLEXITY). *Given a DTMC $X(t)$ over \mathcal{S} defined by \mathbf{P} and a renaming ρ , algorithm ReversibleUpTo($\mathcal{S}, \mathbf{P}, \rho$) returns true iff $(\mathcal{S}, \mathbf{P})$ is ρ -reversible. If $X(t)$ is ρ -reversible, at the end of the computation the normalized vector $\frac{1}{\sum_{i \in \mathcal{S}} \Pi[i]} \Pi$ contains the stationary distribution of $X(t)$.*

ReversibleUpTo($\mathcal{S}, \mathbf{P}, \rho$) can be implemented so as to run in time $O(n + m)$, where n is the number of states of \mathcal{S} and m is the number of strictly positive elements of \mathbf{P} .

PROOF. The correctness of the algorithm immediately follows from Lemma 2.

The algorithm has time and space complexities of a DFS-visit. Hence, if \mathbf{P} is stored using weighted adjacency lists, the complexity thesis follows. \square

3.2 Deciding Reversibility modulo Renaming

Let us now consider the *Reversibility modulo Renaming* decision problem, i.e.: given a (ergodic stationary) DTMC $X(t)$ over a finite state space \mathcal{S} decide whether there exists a renaming ρ such that $X(t)$ is ρ -reversible. This is the problem we would like to solve. Moreover, in case of affirmative answer we would like to produce a valid renaming ρ and use it to compute the stationary distribution.

As a consequence of the results in the previous section, the Reversibility modulo Renaming problem is in the class NP. We can say more than that. The following two lemmas prove that our problem is polynomially equivalent to Graph Isomorphism.

LEMMA 3. *Unlabeled Undirected Graph Isomorphism can be polynomially reduced to Reversibility modulo Renaming.*

PROOF. Given two unlabeled undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ we should compute a DTMC $X(t)$ defined by \mathbf{P} such that G_1 is isomorphic to G_2 if and only if $X(t)$ is ρ -reversible for some ρ .

It is not restrictive to assume that G_1 and G_2 are connected, $V_1 \cap V_2 = \emptyset$, and $|V_1| = |V_2| = n$.

With a slight abuse of notation we denote as $G_1 (G_2)$ the oriented graph obtained considering both $\langle u, v \rangle$ and $\langle v, u \rangle$ for each $\{u, v\} \in E_1 (E_2, \text{ respectively})$. G_1 and G_2 are isomorphic if and only if these two oriented graphs are isomorphic.

Let us consider the following oriented graph

$$G = (V, E) = G_1 \cup G_2 \cup (V_3, E_3),$$

where $V_3 = \{a_1, a_2, a_3, b_1, b_2, b_3\}$ are new nodes and

$$E_3 = \{\langle a_1, a_2 \rangle, \langle a_2, a_3 \rangle, \langle a_3, a_1 \rangle\} \cup \{\langle b_1, b_3 \rangle, \langle b_3, b_2 \rangle, \langle b_2, b_1 \rangle\} \cup \{\langle a_2, b_2 \rangle, \langle b_2, a_2 \rangle\} \cup \{\langle a_1, u \rangle, \langle u, a_1 \rangle \mid u \in G_1\} \cup \{\langle b_1, v \rangle, \langle v, b_1 \rangle \mid v \in G_2\}$$

G has $2n + 6$ nodes.

Let \mathbf{P} over V be defined as follows:

- for each $u \in V_1 \cup V_2 \cup \{a_2, a_3, b_2, b_3\}$, for each edge $\langle u, v \rangle$ let $p_{uv} = \frac{1}{deg(u)}$, where $deg(u)$ is outgoing degree of u ;
- for each edge $\langle a_1, u \rangle$ with $u \in V_1$ let $p_{a_1 u} = \frac{1}{n} - \frac{1}{n^3}$;
- let $p_{a_1 a_2} = \frac{1}{n^2}$;
- for each edge $\langle b_1, v \rangle$ with $v \in V_2$ let $p_{b_1 v} = \frac{1}{n} - \frac{1}{n^3}$;
- let $p_{b_1 b_3} = \frac{1}{n^2}$;
- otherwise $p_{ij} = 0$.

\mathbf{P} is a probability matrix.

If G_1 is isomorphic to G_2 , let $\sigma : G_1 \rightarrow G_2$ be an isomorphism from G_1 to G_2 . We consider the following permutation of the nodes of G :

$$\rho(u) = \begin{cases} \sigma(u) & \text{if } u \in V_1 \\ \sigma^{-1}(u) & \text{if } u \in V_2 \\ b_i & \text{if } u = a_i \text{ for } i = 1, 2, 3 \\ a_i & \text{if } u = b_i \text{ for } i = 1, 2, 3 \end{cases}$$

Notice that for each $u \in V_1 \cup V_2 \cup \{a_2, a_3, b_2, b_3\}$ it holds that $\deg(u) = \deg(\rho(u))$.

We prove that $X(t)$ defined by \mathbf{P} is ρ -reversible. As a consequence of Theorem 3 it is sufficient to prove that for each simple cycle C of G it holds $p(C) = p((\rho(C))^{-1})$, where $P(C)$ is the product of the probabilities of the edges occurring in C .

If $C = a_1ua_1$ with u in G_1 , $(\rho(C))^{-1} = \rho(a_1)\rho(u)\rho(a_1) = b_1\rho(u)b_1$ and

$$P((\rho(C))^{-1}) = \left(\frac{1}{n} - \frac{1}{n^3}\right) \frac{1}{\deg(\rho(u))} = \left(\frac{1}{n} - \frac{1}{n^3}\right) \frac{1}{\deg(u)} = P(C)$$

If $C = a_2b_2a_2$, then $(\rho(C))^{-1} = b_2a_2b_2$. Hence,

$$P((\rho(C))^{-1}) = \frac{1}{\deg(b_2)} \frac{1}{\deg(a_2)} = \frac{1}{\deg(a_2)} \frac{1}{\deg(b_2)} = P(C)$$

If $C = a_1a_2a_3a_1$, then $(\rho(C))^{-1} = b_1b_3b_2b_1$. In this case

$$P((\rho(C))^{-1}) = \frac{1}{n^2} \frac{1}{\deg(b_3)} \frac{1}{\deg(b_2)} = \frac{1}{n^2} \frac{1}{\deg(a_3)} \frac{1}{\deg(a_2)} = P(C)$$

Similarly the thesis holds for the cycles b_1vb_1 , $b_2a_2b_2$ and $b_1b_3b_2b_1$.

Now we only have to consider cycles among nodes of G_1 and G_2 . If $C = u_1u_2 \dots u_s u_1$ is a cycle in G_1 , then since σ is an isomorphism $\rho(u_1)\rho(u_2) \dots \rho(u_s)\rho(u_1)$ is a cycle in G_2 . Since $G_2 = G_2^+$, also $\rho(u_1)\rho(u_s) \dots \rho(u_2)\rho(u_1) = (\rho(C))^{-1}$ is a cycle in G_2 . Moreover,

$$P((\rho(C))^{-1}) = \frac{1}{\deg(\rho(u_1))} \frac{1}{\deg(\rho(u_s))} \dots \frac{1}{\deg(\rho(u_2))} = \frac{1}{\deg(u_1)} \frac{1}{\deg(u_s)} \dots \frac{1}{\deg(u_2)} = \frac{1}{\deg(u_1)} \frac{1}{\deg(u_2)} \dots \frac{1}{\deg(u_s)} = P(C)$$

Similarly, the thesis holds if we consider a cycle C over G_2 .

If $X(t)$ defined by \mathbf{P} is ρ -reversible for some renaming ρ , then we consider $\sigma : G_1 \rightarrow G_2$ defined as $\sigma(u) = \rho(u)$. We prove that σ is a graph isomorphism between G_1 and G_2 . First we prove that for each $u \in V_1 \cup V_2$ it holds $\deg(u) = \deg(\rho(u)) = \deg(\sigma(u))$. For each edge $\langle u, w \rangle$ in $E_1 \cup E_2$ we also have $\langle w, u \rangle$ in $E_1 \cup E_2$. Hence, uwu is a cycle in $G_1 \cup G_2$. Since, G is ρ -reversible it corresponds to a cycle $\rho(u)\rho(w)\rho(u)$ in G and vice-versa. Hence, since ρ is permutation (i.e., a bijection), it has to be $\deg(u) = \deg(\rho(u))$. Now we prove that $\rho(a_1)$ has to be b_1 . If by contradiction $\rho(a_1) \neq b_1$, then the probability of any outgoing edge from $\rho(a_1)$ is at least $\frac{1}{n}$. Hence, if we consider $u \in V_1$ we

have $P(a_1ua_1) = \left(\frac{1}{n} - \frac{1}{n^3}\right) \frac{1}{\deg(u)} = P(\rho(a_1)\rho(u)\rho(a_1)) \geq \frac{1}{n} \frac{1}{\deg(\rho(u))} = \frac{1}{n} \frac{1}{\deg(u)}$ which is a contraddiction. For the same reason $\rho(b_1)$ is a_1 . Since the only cycle of length 3 in which a_1 is involved is $a_1a_2a_3a_1$ and the only cycle of length 3 in which b_1 is involved is $b_1b_3b_2b_1$, $\rho(a_1) = b_1$ implies $\rho(a_2) = b_2$ and $\rho(a_3) = b_3$. Similarly, from the fact that $\rho(b_1) = a_1$ we get $\rho(b_2) = a_2$ and $\rho(b_3) = a_3$. Moreover, since all the cycles of length 2 in which a_1 is involved are of the form a_1ua_1 with $u \in V_1$ and the only cycles of length 2 in which b_1 is involved are of the form b_1vb_1 with $v \in V_2$, we get that for each $u \in V_1$ it holds $\sigma(u) = \rho(u) \in V_2$. Moreover, since ρ is a permutation and $|V_1| = |V_2| = n$, σ is a bijection. Finally if $\langle u, w \rangle \in E_1$, then uwu is a cycle in G_1 , this implies that $\sigma(u)\sigma(w)\sigma(v) = \rho(u)\rho(w)\rho(u)$ is a cycle in G_2 . Hence, $\langle \sigma(u)\sigma(w) \rangle \in E_2$. On the other hand, if $\langle \sigma(u), \sigma(w) \rangle \in E_2$, then $\sigma(u)\sigma(w)\sigma(v) = \rho(u)\rho(w)\rho(v)$ is a cycle in G_2 , hence uwv has to be a cycle in G_1 , which implies $\langle u, w \rangle \in E_1$. \square

LEMMA 4. *Reversibility modulo a Renaming can be reduced to Labeled Directed Graph Isomorphism.*

PROOF. Given a DTMC $X(t)$ defined by \mathbf{P} we should exhibit two graphs G_1 and G_2 such that G_1 is isomorphic to G_2 if and only if $X(t)$ is reversible modulo a renaming. Let G_1 be the labeled oriented graph having \mathbf{P} as adjacency matrix, i.e., each edge $\langle u, v \rangle$ is labeled with the value p_{uv} . Let π be the stationary distribution of $X(t)$. Notice that π can be computed from \mathbf{P} in polynomial time. We consider the graph G_2 obtained by labeling each edge $\langle u, v \rangle$ with the value $\frac{\pi_v}{\pi_u} p_{vu}$. If there exists an isomorphism σ from G_1 to G_2 , let $\rho = \sigma$. It holds that for each $u, v \in G$, $\pi_u p_{uv} = \pi_u p_{\rho(u)\rho(v)} = \pi_v p_{\rho(v)\rho(u)}$, i.e., $X(t)$ is ρ -reversible. Similarly, if $X(t)$ is ρ -reversible, then $\sigma = \rho$ is an isomorphism from G_1 to G_2 . \square

Hence, the Reversibility modulo Renaming problem is complete for the class GI (Graph Isomorphism). Currently, Graph Isomorphism is not known to be polynomially solvable nor NP-complete. A large number of heuristics and specialized solvers have been developed to efficiently solve Graph Isomorphism over a large class of graphs. Unfortunately, we cannot exploit the reduction presented in the proof of Lemma 4 together with such solvers to efficiently solve Reversibility modulo Renaming without first computing the stationary distribution. So we are in a vicious circle and we need to develop direct algorithms for solving our problem without having to compute the stationary distribution.

4. ρ -REVERSIBILITY ALGORITHM

In the case of ergodic Markov chains over a finite state space \mathcal{S} , the uniformization method allows to transform a CTMC $X(t)$ into a DTMC $X(t)^U$ having the same stationary distribution. Let \mathbf{Q} be the infinitesimal generator of $X(t)$ and $\nu = \max\{-q_{ii} \mid i \in \mathcal{S}\}$ we define $X(t)^U$ as the DTMC having transition matrix $\mathbf{P}^U = Id + \frac{1}{\nu} \mathbf{Q}$. In view of this uniformization technique, a CTMC $X(t)$ is ρ -reversible if and only if the DTMC $X^U(t)$ is ρ -reversible. Hence, with respect to the definition of a reversibility algorithm, we can focus on DTMCs.

A naive algorithm for Reversibility modulo Renaming could generate all the possible renamings and for each of them exploit Algorithm 1 to test whether the renaming is valid or

not. This would require time $\Omega(n!)$ for generating the set of valid renamings of a process. Our algorithm exploits two main necessary conditions to drastically reduce the set of possible renamings that need to be validated. Of course, in the worst case all renamings pass the conditions and there is no improvement. However, as we will see in the following section, in practical cases our algorithm is effective, while the naive one is not. The advantage of our approach comes from using the necessary conditions once at the beginning of the computation to determine for each state of \mathcal{S} a set of possible renamings.

The first condition we use concerns the number of strictly positive elements in the rows and columns of the matrices defining a process and its ρ -reversed.

DEFINITION 3. Let $X(t)$ be a DTMC over \mathcal{S} defined by \mathbf{P} . For each $i \in \mathcal{S}$ we define:

$$\begin{aligned} in(i) &= |\{j \mid p_{ji} > 0\}| \\ out(i) &= |\{j \mid p_{ij} > 0\}| \end{aligned}$$

PROPOSITION 1 (TOPOLOGY). Let ρ be a renaming over \mathcal{S} . Let $X(t)$ be a DTMC defined by \mathbf{P} . If $X(t)$ is ρ -reversible, then for each $i \in \mathcal{S}$ it holds that:

$$in(i) = out(\rho(i)) \quad \text{and} \quad out(i) = in(\rho(i))$$

PROOF. This is immediate from Theorem 2. \square

The second condition follows from ρ -Kolmogorov's criterion. Each pair of states $i, j \in \mathcal{S}$ of an ergodic Markov chain is involved in at least one simple cycle. From ρ -Kolmogorov's criterion we get that if i and j are involved in a cycle C having probability $p > 0$, then there must be a cycle in the reversed process involving $\rho(i)$ and $\rho(j)$ and having probability p . Given a matrix $\mathbf{A} = (a_{ij})_{i,j \in \mathcal{S}}$ we use the notation $\mathbf{A}^T = (a_{ij}^T)_{i,j \in \mathcal{S}}$ to denote its transposed, i.e., $a_{ij}^T = a_{ji}$.

LEMMA 5. Let ρ be a renaming over \mathcal{S} . Let $X(t)$ be a ρ -reversible DTMC defined by \mathbf{P} . If

$$p_{ii_2} \cdots p_{i_k j} p_{j i_{k+1}} \cdots p_{i_n i} = p > 0$$

then there exists $a_2, \dots, a_k, a_{k+1}, a_n$ such that

$$p_{\rho(i) a_2}^T \cdots p_{a_k \rho(j)}^T p_{\rho(j) a_{k+1}}^T \cdots p_{a_n \rho(i)}^T = p$$

PROOF. This is a consequence of Theorem 3 taking $a_h = \rho(i_h)$, for each $2 \leq h \leq n$. \square

Now we need an efficient way to compute the probability of simple cycles or more in general of paths. Given a transition matrix $\mathbf{P} = (p_{ij})_{i,j \in \mathcal{S}}$, let $\mathbf{P}^\alpha = (p_{ij}^\alpha)_{i,j \in \mathcal{S}} = (-\log p_{ij})_{i,j \in \mathcal{S}}$ and $\mathbf{P}^\beta = (\mathbf{P}^\alpha)^T$. Thanks to the properties of logarithms, Lemma 5 can be rewritten as follows.

LEMMA 6. Let ρ be a renaming over \mathcal{S} . Let $X(t)$ be a ρ -reversible DTMC defined by \mathbf{P} . If

$$p_{ii_2}^\alpha + \cdots + p_{i_k j}^\alpha + p_{j i_{k+1}}^\alpha + \cdots + p_{i_n i}^\alpha = p' \geq 0$$

then there exists $a_2, \dots, a_k, a_{k+1}, a_n$ such that

$$p_{\rho(i) a_2}^\beta + \cdots + p_{a_k \rho(j)}^\beta + p_{\rho(j) a_{k+1}}^\beta + \cdots + p_{a_n \rho(i)}^\beta = p'$$

PROOF. It immediately follows from Lemma 5 \square

Notice that the elements of \mathbf{P}^α and \mathbf{P}^β are all positive numbers (including 0 and $+\infty$). Hence, we can interpret \mathbf{P}^α and \mathbf{P}^β as adjacency matrices of weighted graphs and exploit an

algorithm for solving the *All Pairs Shortest Paths* problem over them. Let Δ^α and Δ^β be the matrices obtained as output of such computations. In the graph induced by \mathbf{P}^α the simple cycle of minimum weight involving both i and j has weight $\Delta_{ij}^\alpha + \Delta_{ji}^\alpha$. Similarly, in the graph induced by \mathbf{P}^β the simple cycle of minimum weight involving both $\rho(i)$ and $\rho(j)$ has weight $\Delta_{\rho(i)\rho(j)}^\beta + \Delta_{\rho(j),\rho(i)}^\beta$. As a consequence of Lemma 6 these two values have to coincide.

PROPOSITION 2 (CYCLES). Let ρ be a renaming over \mathcal{S} . Let $X(t)$ be a DTMC defined by \mathbf{P} . If $X(t)$ is ρ -reversible, then for all $i, j \in \mathcal{S}$ it holds that

$$\Delta_{ij}^\alpha + \Delta_{ji}^\alpha = \Delta_{\rho(i)\rho(j)}^\beta + \Delta_{\rho(j),\rho(i)}^\beta$$

Notice that the shortest cycles determined by the above proposition corresponds to the cycle of highest probability in the Markov chain.

We are now ready to put together the necessary conditions of Propositions 1 and 2 in our algorithm. For each state i we compute a set \mathcal{I}_i of candidates $\rho(i)$ using Proposition 1, then we refine \mathcal{I}_i in \mathcal{R}_i using Proposition 2, finally we test all the possible combinations of the remaining candidates with Algorithm 1.

Algorithm 3: ComputeRenamings(\mathcal{S}, \mathbf{P})
<pre> {Initialize $P^\alpha, P^\beta, \Delta^\alpha$ and Δ^β} for $i \in \mathcal{S}$ do $\mathcal{I}_i = \emptyset$ for $j \in \mathcal{S}$ do if $in(i) = out(j) \wedge out(i) = in(j)$ then $\mathcal{I}_i = \mathcal{I}_i \cup \{j\}$ end end end for $i \in \mathcal{S}$ do $\mathcal{R}_i = \emptyset$ for $k \in \mathcal{I}_i$ do if $\forall j \in \mathcal{S} \setminus \{i\} \exists h \in \mathcal{I}_j \setminus \{k\}$ s.t. $\Delta^\alpha[i, j] + \Delta^\alpha[j, i] = \Delta^\beta[k, h] + \Delta^\beta[h, k]$ then $\mathcal{R}_i = \mathcal{R}_i \cup \{k\}$ end end end end $\mathcal{F} = \emptyset$ for $\rho \in \mathcal{R}_1 \times \mathcal{R}_2 \times \dots \times \mathcal{R}_n$ do if ReversibleUpTo($\mathcal{S}, \mathbf{P}, \rho$) then $\mathcal{F} = \mathcal{F} \cup \{\rho\}$ end end return \mathcal{F} </pre>

LEMMA 7. Given a DTMC $X(t)$ over \mathcal{S} defined by \mathbf{P} , Algorithm (3) returns the set \mathcal{F} which contains all the valid renaming functions for the ρ -reversibility of the given chain.

PROOF. \Rightarrow) The proof is trivial since the proposed algorithm rejects all the renaming that don't satisfy the conditions of Propositions 1 and 2 which are tested respectively in the first and second for-loop of the algorithm. All the remaining permutations are tested in the last for-loop with Algorithm (1), this ensures that at the end \mathcal{F} will contain only the valid renaming functions for ρ -reversibility, if there exists any. \square

We now evaluate the time complexity of Algorithm (3). Let $|\mathcal{S}| = n$. The first part of the algorithm, which includes the initialization of the data structures and the first two for-loop, has a time complexity of $O(n^4)$. In particular, the initialization of P^α and P^β requires time $\Theta(n^2)$, while the initialization of Δ^α and Δ^β can be done using for instance Floyd-Warshall algorithm in time $\Theta(n^3)$. The first for-loop requires time $\Theta(n^2)$, while the second one requires time $\Theta(n^4)$ only if the first one has generated large sets \mathcal{I}_i . The second part of the algorithm, i.e., the last for-loop, consists of $\gamma = |\mathcal{R}_1 \times \mathcal{R}_2 \times \dots \times \mathcal{R}_n|$ calls to Algorithm 1. Hence, by Theorem 4, the complexity of the second part is $O(\gamma * n^2)$. This give us a total complexity of $O(n^4 + \gamma * n^2)$.

5. EXPERIMENTAL RESULTS

The validation of the proposed algorithm has been performed on a set of 9 different ρ -reversible chains, with sizes ranging from 8 to 24 nodes. While testing the method with larger chains would be interesting for evaluating the computational efficiency, it would be very difficult to synthetically produce large ρ -reversible chains. On the other hand, testing the method with non ρ -reversible graphs would result in an immediate trivial answer by the algorithm. The first 4 processes are modeled by discrete time Markov chains and are not related to any specific real-world scenario. They are just designed in order to exhibit a valid renaming function and to offer a variable number of states. These synthetic chains, labeled with identifiers from DTMC1 to DTMC4, are presented in Fig. 1. They range from a minimum of 5 states to a maximum of 10 states for the most complex case. In addition to these discrete time chains, we also tested the behavior of the algorithm with 5 continuous time Markov chains, by first transforming them as described in Section 4. These latter chains have been obtained by modeling a practical scenario.

In detail, we show how the proposed algorithm can be applied to a model of a real system. To this end, the algorithm is used to verify the ρ -reversibility of the Markov chain underlying the analytical model used for the performance evaluation of the Fair Allocation Control Window (FACW) protocol [11].

The main idea of FACW is that data traffic in a Wireless Sensor Network (WSN) can be classified into a finite set of M classes $\mathcal{K} = \{c_1, c_2, \dots, c_M\}$. 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. 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 either rescheduled for transmission after a back-off time or simply dropped. Otherwise, in case of generation of a class c packet when the number of c -entries in the window is strictly lower than h_c , the packet is sent and the window is updated according to a FIFO policy. The initialization of the window is arbitrary.

We consider a set $\mathcal{K} = \{c_1, c_2, \dots, c_M\}$ 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. The state of the window is denoted by $\vec{x} = (x_1, x_2, \dots, x_N)$, where $x_i \in \mathcal{K}$, and $|\vec{x}|_c = \sum_{i=1}^N \delta_{x_i=c}$ represents the total number of occurrences of class c in \mathcal{W} . Data of different traffic classes are generated according to independent Poisson processes

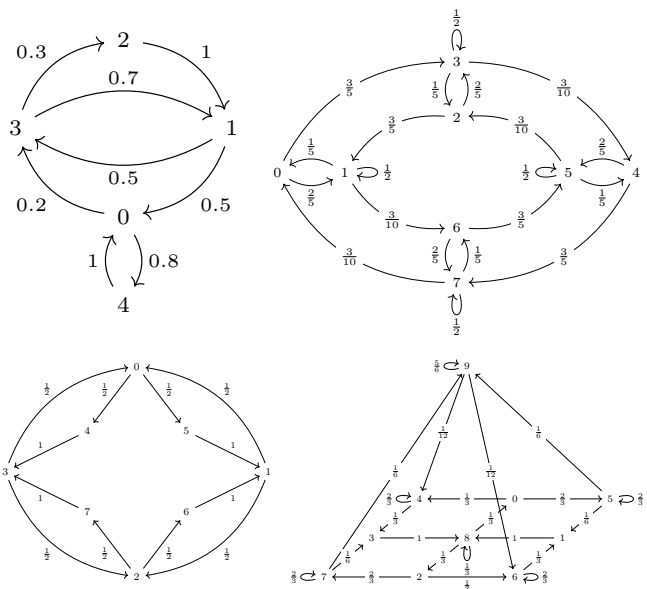


Figure 1: The 4 synthetic discrete time Markov chains used for experimental validation. In the text we will refer to these chains as DTMC1 and DTMC2 for the examples in the first row and DTMC3, DTMC4 for those on the second row.

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 CTMC. 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 $\vec{x} = (x_1, \dots, x_N)$ be the state of the control window, then the transition rates in the CTMC infinitesimal generator are: for $\vec{x} \neq \vec{x}'$,

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

We created a total of 5 different CTMCs based on the FACW model with the parameters showed in Table 3.

Chain	N	h_c	$ \mathcal{S} $	Classes	Rates λ
CTMC1	3	3	8	{1, 2}	{1.0, 2.0}
CTMC2	4	3	14	{1, 2}	{1.0, 2.0}
CTMC3	4	4	16	{1, 2}	{1.0, 2.0}
CTMC4	2	2	16	{1, 2, 3, 4}	{1.0, 2.0, 3.0, 4.0}
CTMC5	3	2	24	{1, 2, 3}	{1.0, 2.0, 3.0}

Table 1: Parameters for the FACW model of the generated CTMCs

5.1 Efficiency of the algorithm

In order to assess the efficiency of the proposed algorithm we compare its execution times with the naive approach. From Table 2 we can notice the improvement in the execution times, the *Renaming Algorithm* (column t_{RA}) is able to verify the ρ -reversibility of the tested chains much faster than the naive approach (column $t_{factorial}$). It should be noted that we were not able to complete the computation using the naive approach for all the chains. In fact, with a size of the state space of 16 or more the naive method did not terminate after a whole 24 hours.

Chain	$ \mathcal{S} $	t_{RA}	$t_{factorial}$
DTMC1	5	2	2
DTMC2	8	6	41
CTMC1	8	7	41
DTMC3	8	12	56
DTMC4	10	4	4948
CTMC2	14	2864	6306581
CTMC3	16	2334	—
CTMC4	16	101	—
CTMC5	24	9168	—

Table 2: Execution times in milliseconds

5.2 Suitability of the results

The congruence of the renaming function obtained is easy to verify in practice and the correctness of the verification method has been proven theoretically. Still, we are interested in showing a practical application of the obtained renaming function and to highlight its suitability with respect to other methods that can be used to obtain the same result.

To this end we decided to exploit the renaming map to compute the stationary distribution of tested chains. In fact the notion of ρ -reversibility, as shown in [8], can be used to efficiently compute the vector π . We also computed the stationary distribution for the same chains by adopting the rather standard power method simulation (i.e., repeated multiplications).

In Tables 3 and 4 we compare the approximation error of the stationary distribution resulting from the direct computation over the renaming function with the power method after respectively 50 and 100 iterations. In detail, Table 3 shows the results for the discrete time examples and Table 4 reports the results for the continuous time chains. Approximations error ϵ for the discrete case has been computed as $\epsilon = |\pi\mathbf{P} - \pi|^2$. Differently, we computed the error ϵ for the continuous case as $\epsilon = |\pi\mathbf{Q}|^2$. It can be observed that, in general, the direct computation through the iterative method requires a large number of iterations to obtain a level of accuracy comparable with that of our method.

Chain	$ \mathcal{S} $	ϵ_{RA}	ϵ_{mult50}	$\epsilon_{mult100}$
DTMC1	5	$7.881E - 17$	$3.429E - 5$	$2.299E - 8$
DTMC2	8	0.0	$5.551E - 17$	0.0
DTMC3	8	0.0	$3.065E - 11$	0.0
DTMC4	10	$5.551E - 17$	$6.218E - 9$	$1.582E - 15$

Table 3: Discrete chains

Chain	$ \mathcal{S} $	ϵ_{RA}	ϵ_{mult50}	$\epsilon_{mult100}$
CTMC1	8	0.0	0.0	0.0
CTMC2	14	0.0	$1.175E - 6$	$1.881E - 9$
CTMC3	16	0.0	0.0	0.0
CTMC4	16	$3.119E - 16$	$2.775E - 17$	$1.387E - 17$
CTMC5	24	$5.846E - 17$	$3.869E - 7$	$1.419E - 10$

Table 4: Continuous chains

6. CONCLUSIONS

In this paper we proposed a practical algorithm that allows the computation of all the renamings functions for both continuous and discrete time Markov chains. Our algorithm reduces the number of renaming functions that need to be tested by exploiting: properties about the structure of the chain; rules of algebra on the logarithmic function applied to the Kolmogorov’s criterion; and the properties of shortest paths connecting states in the chain.

We compared the execution times of our algorithm with those required by the naive approach showing that the former is much faster than the latter, and that it can obtain solutions where the naive approach cannot. As far as we know our is the first proposal for solving this problem.

As future work we plan to implement other heuristics for polynomially pruning the set of possible renamings and to extensively test them on large systems.

Acknowledgments. Research partially supported by University of Udine PRID ENCASE project and by INDAM-GNCS project "Logica e Automi per il Model-Checking Intervallare"

7. REFERENCES

- [1] S. Balsamo, F. Cavallin, A. Marin, and S. Rossi. Applying reversibility theory for the performance evaluation of reversible computations. In *Proc. of ASMTA*, 9845 LNCS:45–59, 2016.
- [2] S. Balsamo and G. Iazeolla. Aggregation and disaggregation in queueing networks: The principle of product-form synthesis. In *Computer Performance and Reliability*, pages 95–109, 1983.
- [3] S. Balsamo and A. Marin. Performance engineering with product-form models: efficient solutions and applications. In *Proc. of ICPE*, pages 437–448, 2011.
- [4] P. G. Harrison. Turning back time in Markovian process algebra. *Theoretical Computer Science*, 290(3):1947–1986, 2003.
- [5] J. Hillston, A. Marin, S. Rossi, and C. Piazza. Contextual lumpability. In *VALUETOOLS*, pages 194–203, 2013.
- [6] F. Kelly. *Reversibility and stochastic networks*. Wiley, New York, 1979.
- [7] A. Marin and S. Rossi. Autoreversibility: exploiting symmetries in Markov chains. In *Proc. of MASCOTS*, pages 151–160, 2013.
- [8] A. Marin and S. Rossi. On discrete time reversibility modulo state renaming and its applications. In *VALUETOOLS*, 2014.
- [9] A. Marin and S. Rossi. On the relations between lumpability and reversibility. In *In Proc. of MASCOTS*, pages 427–432, 2014.
- [10] A. Marin and S. Rossi. On the relations between markov chain lumpability and reversibility. In *Acta Informatica*, pages 1–39, 2016.
- [11] A. Marin and S. Rossi. Priority-based bandwidth allocation in wireless sensor networks. *EAI Endorsed Trans. Wireless Spectrum*, 2(10), 2016.
- [12] A. Marin and S. Rossi. On the relations between Markov chain lumpability and reversibility. *Acta Inf.*, 54(5):447–485, 2017.
- [13] U. Sumita and M. Rieders. Lumpability and time reversibility in the aggregation-disaggregation method for large markov chains. *Communications in Statistics: Stochastic Models*, 5:63–81, 1989.
- [14] P. Whittle. *Systems in stochastic equilibrium*. John Wiley & Sons Ltd., 1986.