ABSTRACT

Title of dissertation: QUANTIFYING FLOWS IN TIME-IRREVERSIBLE MARKOV CHAINS

Danielle Middlebrooks
Doctor of Philosophy, 2020

Dissertation directed by: Professor Maria Cameron
Department of Mathematics

Stochastic networks a.k.a. Markov chains allow us to model phenomena in systems arising in many applications. The appeal of stochastic networks is that they offer a mathematically tractable and robust model focused on the most important features of the system. Nevertheless, stochastic networks approximating complex systems can be huge and unstructured, and an effective description of their dynamics is a challenging mathematical problem.

This dissertation is motivated by our study of two models of a gene regulatory network (GRN), one deterministic [1] and one stochastic [2], which describes the budding yeast cell cycle. A GRN with N nodes can be straightforwardly converted into a Markov chain with 2^N states. Our scientific goal is to understand how the stochasticity affects the stability of the cell cycle in the GRN. This gives rise to our mathematical goal: to develop efficient tools for quantifying dynamics of large time-irreversible Markov chains.

Our methodological developments are built upon the transition path theory

(TPT) [16] which is a general framework for describing transitions in Markov chains between two subsets of states. In TPT, the transition process is described by the so-called effective current. We have realized that the effective current gives a lopsided description of the transition process in the case of time-irreversible networks where elementary cycles of length greater than two are present. Thus, we have introduced the so-called acyclic current that gives a quantitative description of a transition process and proposed an algorithm to compute it. Moreover, we have developed a general recipe to modify the generator matrix of a given Markov chain in order to make the stationary probability current and the invariant distribution in the modified chain coincide with a desired current and a desired invariant distribution in the original chain.

Finally, we have applied these tools to the budding yeast cell cycle GRN. Our results show which edges are essential and which ones are redundant. Our computations eloquently demonstrate that stochasticity makes the GRN much more stable with respect to edge removals. This conclusion is consistent with Q. Nie's statement [26] that stochasticity plays a fundamental role in biological processes.

Quantifying Flows in Time-Irreversible Markov Chains

by

Danielle Middlebrooks

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2020

Advisory Committee:
Professor Maria Cameron, Chair/Advisor
Professor Kasso Okoudjou
Professor Howard Elman
Professor Konstantina Trivisa
Professor Michelle Girvan

Dedication

To my grandparents.

Forever and always in my heart.

Acknowledgments

Over the past six years I have received support and encouragement from a number of individuals. First and foremost, I would like to express my deepest gratitude to my advisor, Professor Maria Cameron. She has gone above and beyond in her role as an advisor and not only has made me a better researcher, but a better person. My success in graduate school and the completion of this dissertation would not be possible without her guidance, support, and patience, and I will forever be indebted to her.

I am also extremely grateful to my committee members. To Kasso Okoudjou for his unwavering support and belief in my abilities, Howard Elman and Konstantina Trivisa for their guidance and helpful advice when I first arrived at the university and throughout years here, and Michelle Girvan for her helpful contributions and suggestions for part of this project.

I wish to thank my loving parents, Gwen and Rodney and my older brother Rodney Jr. I am thankful for their love and support throughout my life. Thank you for always believing in me and encouraging me to always reach for the stars. To my aunts, uncles and many cousins, thank you for your moral support and constant words of encouragement.

To my many friends near and far, thank you for listening, offering me advice, and supporting me through this entire process. Thank you to my AMSC cohort- Addison, Tengfei, Cara and David for your friendship during our time at UMD. My professors at Spelman College for encouraging me to attend graduate school and their words of kindness throughout the years. Thank you to my 2014 EDGE program sisters in mathematics for going through this journey with me and sharing your experiences as well as listening to mine. A special thank you to one of my closest friends Kayla Davie for always being a shoulder to lean on and someone to vent to. I am so glad you came to UMD and I can't wait for you to be up next!

I would like to thank the COMBINE program for their financial support. To the COMBINE staff and students, thank you for listening to my research talks and providing valuable feedback.

Above all, I would like to thank God for giving me the strength, knowledge and ability to complete this dissertation. Without His blessings, this achievement would not have been possible.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

DFS	Depth-first search
GRN	Gene regulatory network
MJP	Markov jump process
SCC	Strongly connected component
TPT	Transition path theory

Chapter 1: Introduction

1.1 Overview

Over recent years, the use of large graphs or networks has emerged as a popular tool for data representation and organization [3]. Through networks, one is able to identify patterns and structure within the data. The use of networks can be applied in many fields of science, including economics [4], chemical physics [5], social sciences [6, 7] and biology [8, 9]. Data points are mapped onto the set of nodes in the network and edges or links in the network indicate pairwise relationships between the nodes. For instance, webpages on the World Wide Web can be linked according to URLs, banks and businesses across the country can be linked according to investments made, and proteins in a cell can be linked according to their action on other proteins.

The size and complexity of such networks exceeds by far those of graphs considered in the classic graph theory [8]. This fact led to the emergence of research in network science. Network science is a relatively new discipline that has only been able to blossom because of computer technologies. With computers, scientists are capable of analyzing large-scale networks such as the World Wide Web which has approximately 500 billion nodes [10]. Network science draws on theories and methods from graph theory, statistical mechanics, inferential modeling and information visualization to solve challenging problems in various real-world applications [3].

While detecting structure in a network is crucial for finding relationships within the data [11, 12], important questions that we would like to answer arise from understanding the dynamics of a network. For instance, in modeling Lennard-Jones particle configurations [5], where nodes correspond to local potential minima and edges represent transition states between them, one can find probabilities from one configuration to another. In this work, we consider dynamical networks arising from gene regulatory networks (GRNs). One way of analyzing the dynamics of these networks is to represent these positive weighted networks as a Markov chain. Hence, tools for studying Markov chains can be used for understanding the dynamics of GRNs and validating the GRN models. Furthermore, in this work we address the question of robustness of the system modeled by a GRN with respect to edge removals.

Several approaches for quantifying the dynamics of complex networks have been introduced. Bovier et al. developed the potential theoretical approach and advanced the mathematical spectral theory of metastability [13, 14]. This theory is built upon an analogy with electric circuits. This work derived sharp estimates for low-lying spectra of time-reversible Markov chains. Another approach is the transition path theory (TPT) which was originally proposed by E and Vanden-Eijnden in the context of stochastic differential equations [15] and then transferred and extended to networks in [16, 17]. Like the potential theoretical approach, TPT was inspired by an analogy with electric circuits. However, TPT does not assume time-reversibility and focuses on the statistical analysis of so-called reactive trajectories (see next paragraph). An alternative approach for analyzing reactive trajectories based on a set of recurrence relationships proposed by Manhart and Morozov [18,19].

After considering several options, we found TPT meets our needs to the largest extent. The basic idea of TPT is to single out two specific subsets of nodes, a "reactant" set A and a "product" set B, and analyze the statistical properties of the trajectories by which transitions between these sets occur. Those trajectories that travel strictly from A to B without returning to A in between are called reactive. Statistical analysis of reactive trajectories gives a quantitative description of the transition process between the sets A and B.

1.2 Goals

In this dissertation, we are mainly interested in applications where the given Markov chain is time-irreversible. Time-irreversible Markov chains often arise in biological models, in particular gene regulatory networks (GRN) [2, 20]. In a GRN, the nodes are a collection of molecular regulators that govern gene expression levels which describe a biological process of a given cell. Edges between nodes represent interactions. These interactions can be considered "activating", with an increase in the concentration of one leading to an increase in the other, or "inhibiting", with an increase in one leading to a decrease in the other. The nodes could also regulate themselves directly, creating feedback loops. Gene regulatory networks have recently attracted a great deal of interest [20–23]. In particular, budding yeast have been under investigation since the underlying molecular machinery regulating this process has been highly studied and mathematical models of how the interacting proteins control each stage of the process already exist [1, 2, 24, 25]. While there are ≈ 800 genes involved in the process of the budding yeast cell cycle, there are only 11 key regulators that are responsible for the control and regulation of this process. Hence, the budding yeast GRN consists of only 11 nodes [1]. Our goal is to develop computational tools in the framework of TPT that would allow us to analyze the dynamics of a stochastic budding yeast GRN. We wish to understand and quantify the effects of stochasticity in this GRN. In particular, we would like to establish which edges are critical for the proper function of the cell cycle, and which are not, and understand how stochasticity effects these subsets of edges. To benchmark the effect of stochasticity, we first answer this question for the corresponding deterministic GRN.

1.3 A Brief Summary of Main Results

In the context of TPT, the effective current is defined as the net average number of reactive trajectories per unit time making a transition from one node i to another node j on their way from A to B. If a Markov chain is time-reversible, the effective current is acyclic and we can effectively describe the transition process of the induced graph. If a Markov chain is time-irreversible, the effective current may be cyclic. As a result, the effective current along a single edge along the path from A to B might be larger than total reactive current coming out of A which is the same as the total reactive current going into B. This was first observed when applying the tools from TPT to the stochastic model of the gene regulatory network in [2]. This dissertation involves three major parts: (1) we propose a general framework for designing modified Markov chains in order to quantify transitions in time-irreversible Markov chains, (2) we introduce an algorithm for obtaining an acyclic current and generating acyclic reaction pathways, and (3) we propose a so-called "mutation analysis" for gene regulatory networks allowing one to access their robustness and use the proposed tools to analyze a stochastic budding yeast gene regulatory network.

We develop a general framework for modifying the original time-irreversible irreducible Markov chain to make it have a desired stationary current and a desired invariant distribution. We apply our recipe to time-irreversible networks so that the stationary currents in the modified Markov chains are equal to the effective currents and acyclic currents in the original ones. Our construction generalized Propositions 1 and 2 in Ref. [17]. We can then sample acyclic trajectories using an acyclic current.

In order to design a modified Markov chain and generate the desired trajectories, we propose a cycle removal algorithm which introduces a so-called acyclic current. While an acyclic current may not be unique, we devoted substantial effort to address this issue in our application to the gene regulatory network. Our algorithm is designed for generating a weighted directed acyclic graph from a graph

$G(S, \{f^+\})$ where S is the set of vertices inherited from the original Markov chain, and $f^+ \geq 0$ is the effective current used for the weights of the arcs.

Our methodology for quantifying transition processes in Markov chains is applied to the stochastic GRN representing the budding yeast cell cycle to determine essential edges in the network. The cell cycle is a process in which one cell grows and then proceeds to divide into two cells. The cell cycle consists of four phases: growth (G1), DNA synthesis (S), gap (G2) and mitosis (M). We consider two models of the gene regulatory network for the budding yeast cell cycle: one deterministic and one stochastic. Li et al. (2004) [1] developed a deterministic model which uses boolean functions as update rules to model the dynamics of the GRN. The GRN consists of only 11 proteins/protein complexes. The dynamics of the gene regulatory network is defined by (i) the influence matrix constructed by the interaction between nodes and (ii) the transition rules defined in [1]. In 2006, Zhang et al. [2] modified the deterministic model [1] by making the transition rules stochastic. The stochastic model builds upon the deterministic model by adding certain degrees of unpredictability or randomness that may happen due to the environment and allows for the model to self-organize. Stochasticity plays a fundamental role in biological processes [26–28]. Due to internal noise, genetically identical cells may assume different fates within a consistent environment. This can be caused by the inherently random nature of the biochemical reactions of gene expression which can cause noisy gene expression levels [29].

The models [1, 2] allow us to describe the dynamics of the GRN through what we call the dynamical network. Each node can be in one of two states, either active or passive. Each phase of the cell cycle is described by certain protein/protein complexes being active. For any GRN with n nodes, the dynamical network is the graph with 2^n vertices representing all possible cell states, and edges representing transitions between the states. This dynamical network produces a discrete-time Markov chain. In particular, one that is time-irreversible. Thus, for the stochastic GRN, we use transition path theory and our cycle removal algorithm to quantify the transitions starting at the excited growth state, through all cell cycle phases, then arriving at the stationary growth state right before the process repeats.

Finally, we perform a "mutation analysis" in order to identify redundant edges in the GRN. For the deterministic GRN, we apply the following steps: We first distort the GRN by removing one edge and adjusting the influence matrix. We then apply the transition rules described in [1] to obtain the dynamical network and explore it using the depth-first search (DFS) algorithm. Finally, we check if there is a pathway corresponding to the cell cycle going from the excited growth phase to the stationary growth phase. If such a pathway exists, we compare it to the one in the original dynamical network and mark changes in it if any. We repeat this process for all edges of the regulatory network in order to determine which edges can be removed without a major effect on the cell cycle.

The stochastic transition rules create a dense stochastic matrix. This results in a complete weighted dynamical network. Since DFS ignores the weights of a weighted graph, this method alone is unsuitable here. To identify essential edges in the stochastic GRN, we remove an edge and use the transition path theory (TPT) and our cycle removal algorithm to the respective dynamical network to obtain pathways between the excited growth phase and the stationary growth phase carrying at least 20% of the normalize acyclic current. We repeat this process for each edge in the GRN to determine essential and non-essential edges.

We found that in the deterministic GRN, about 41% of edges can be removed without a major effect on the cell cycle. In the stochastic one, this is true for 88% of edges. This suggests that stochasticity renders the cell cycle much more robust. This is consistent with Q. Nie's statement that stochasticity plays a fundamental role in biological processes [26].

The rest of this dissertation is organized as follows. Chapter 2 gives an overview of Markov chains and Markov jump processes as well as introduces the key concepts and definitions of transition path theory. Throughout this work we consider Markov jump processes but our results are also applicable to discrete-time Markov chains. In Chapter 3, we introduce our construction for designing modified Markov jump processes to generate desired reactive trajectories. This is accompanied with a theorem and proof justifying this construction. In Chapter 4, we propose our algorithm for generating an acyclic current from the original time-irreversible Markov jump processes. Chapter 5 applies this methodology to the stochastic GRN for the Budding yeast cell cycle to analyze the dynamics of the model under mutations. Chapter 6 summarizes the project and gives avenues for future research.

Chapter 2: Background

Markov chains are among the most well known and established probabilistic models. Dating back to the early 20th century, they have long been developed, studied and applied to real world problems. In Sections 2.1 and 2.2, we present definitions and basic properties of both discrete-time and continuous-time Markov chains.

An important arsenal of tools for analysis of transitions taking place in Markov chains is offered by the transition path theory (TPT). The TPT was originally introduced in 2006 by W.E. and E. Vanden-Eijnden [15] as a framework for analysis of rare reactive events in systems evolving according to stochastic differential equations. It was adopted for continuous-time Markov chains also known as Markov jump processes by Metzner et al. in 2009 [16]. The TPT was motivated by the transition state theory (TST) developed by Eyring in 1935 [30] to quantify chemical reactions. In Section 2.3, we give an overview of TPT and discuss basic definitions and theorems that will be necessary throughout this dissertation.

2.1 Discrete-time Markov Chains

Let S denote the state space which is finite or countable. We consider discrete moments of time 0, 1, 2, ... and denote the state of the system at time n by X_n. A discrete-time Markov chain is defined as a sequence of random variables, $(X_n)_{n\geq 0}$ taking on values in S, that is characterized by the Markov property, which means the future state of the process is only dependent on its present state [31]. In other words, given the present state X_n, any future state X_{n+k} is independent of any past state X_{n-m}. Thus a Markov chain is characterized by a transition matrix P where $P_{i,j}$ is the probability that the Markov chain will be in state j in the next time step given that it is currently in state i. Note that the ith row of P is the probability distribution for X_{n+1} conditioned on the fact that $X_n = i$. Therefore, all entries of the matrix P are nonnegative, and the row sums are equal to one i.e.

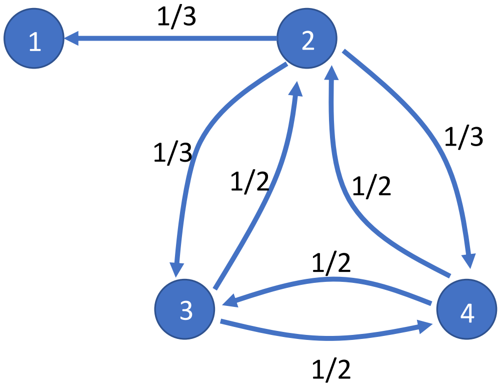
$$\begin{cases} P_{i,j} \geq 0, & \forall i, j \in S, i \neq j \\ \sum_{j\in S} P_{i,j} = 1, & \forall i \in S \end{cases} \tag{2.1}$$

Any matrix P satisfying these conditions in called *stochastic*. A formal definition of a Markov chain is:

Definition 1. *A sequence of random variables $(X_n)_{n\geq 0}$ is a Markov chain with initial distribution λ and stochastic matrix P if*

- *X_0 has distribution $\lambda = \{\lambda_i | i \in S\}$ and*
- *the Markov property holds*

$$\mathbb{P}(X_{n+1} = i_{n+1} | X_n = i_n, ..., X_0 = i_0) = \mathbb{P}(X_{n+1} = i_{n+1} | X_n = i_n) = p_{i_n, i_{n+1}}.$$



$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix}$$

Figure 2.1: Graph representation of a discrete-time Markov chain with 4 nodes. The probability to move from one state to another is given along each arc. The matrix representation is given by matrix P.

A probability distribution in a Markov chain is called equilibrium and denoted by π if

$$\pi P = \pi. \tag{2.2}$$

Markov chains can be divided into time-reversible and time-irreversible. A Markov chain is *time-reversible* if and only if it satisfies the detailed balance conditions

$$\pi_i P_{i,j} = \pi_j P_{j,i} \quad \forall i, j.$$

In other words, for each pair of states i, j, as the length of time interval tends to infinity, the expected numbers of transitions from i to j and from j to i, divided by the length of the time interval, approach the same limit equal to $\pi_i P_{i,j} = \pi_j P_{j,i}$.

In this dissertation, we are mainly interested in applications where the given Markov chain is time-irreversible. The stochastic matrix for the time-reversed chain is defines as follows.: Let $\hat{P}$ be the transition matrix for the time-reversed process, then

$$\begin{aligned}\hat{P}_{i,j} &= P\left(X_m = j | X_{m+1} = i\right) \\ &= \frac{P\left(X_m = j, X_{m+1} = i\right)}{P\left(X_{m+1} = i\right)} \\ &= \frac{P\left(X_m = j\right) P\left(X_{m+1} = i | X_m = j\right)}{P\left(X_{m+1} = i\right)} \\ &= \frac{\pi_j}{\pi_i} P_{j,i} \qquad (2.3)\end{aligned}$$

using Bayes' theorem.

2.2 Markov Jump Process (MJP)

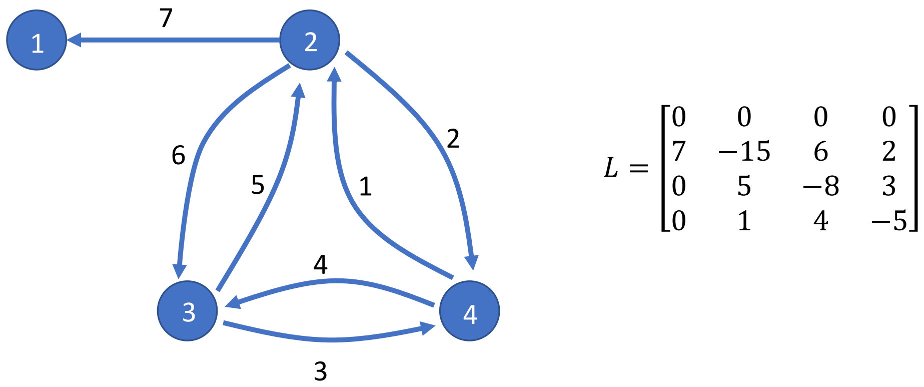
In discrete-time Markov chains, the transitions can occur only at discrete moments of time. More general models allowing the transitions to happen at any moment of time and enabling faster simulations are continuous-time Markov chains also known as Markov Jump Processes (MJPs). The dynamics of a MJP is characterized by the generator matrix L where

$$\begin{cases} L_{i,j} \geq 0, & \forall i, j \in S, i \neq j \\ \sum_{j \in S} L_{i,j} = 0, & \forall i \in S \end{cases} \qquad (2.4)$$

The entries of the generator matrix are interpreted as follows: $L_i = \sum_{j \neq i} L_{i,j}$ is the escape rate from i. The ratio $L_{i,j}/L_i$ is the probability to jump from i to j. For brevity we call $L_{i,j}$ pairwise rates.

We assume L is irreducible, meaning it is possible to get to any state from any other state in the system. It has a unique equilibrium distribution satisfying

$$\pi^T L = 0, \quad \sum_{i \in S} \pi_i = 1. \qquad (2.5)$$



$$L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 7 & -15 & 6 & 2 \\ 0 & 5 & -8 & 3 \\ 0 & 1 & 4 & -5 \end{bmatrix}$$

Figure 2.2: Graph representation of a Markov jump process with 4 nodes. The pairwise rate to move from one state to another is given along each arc. The matrix representation is given by matrix L.

Moreover, independent of the initial distribution, the probability distribution in the system approaches π as time tends to infinity. The generator for the time-reversed MJP is defined by

$$\hat{L}_{i,j} = \frac{\pi_j}{\pi_i} L_{j,i}. \tag{2.6}$$

2.3 Transition Path Theory (TPT)

The discrete transition path theory (TPT) offers a framework to give a quantitative description of the transition process between two given disjoint subsets of states in systems evolving according to MJPs [16].

2.3.1 Motivation

The transition state theory (TST) [30], [32] was introduced in the 1930's to provide a framework for the description of rare events. The TST was derived in the context of analyzing the rate of chemical reactions $R \to P$, where R denotes the reactant state and P the product state. The idea is to pick the optimal dividing surface where the number of crossings from R to P is minimal. However, it is not clear where the optimal surface is. Even if the dividing surface is picked optimally, the number of crossings can be significantly larger than the number of transitions from the reactant state to the product state within any interval of time. This happens due to recrossings. While TST was primarily used to understand qualitatively how chemical reactions take place, some applications require a more quantitative description.

Another relevant approach for describing rare events is the transition-path sampling (TPS) [33]. Instead of counting the number of crosses of some dividing surface as in the TST, the TPS generates an ensemble of reactive trajectories, i.e., those pieces of trajectories that leave the reactant state and enter the product state. Figure 2.3 depicts a few reactive trajectories between two subsets of states representing the reactive state and the product state respectively. Thus, all the relevant information can then be extracted from the ensemble, such as the reaction mechanism and the transition states. However, these reactive trajectories may be very complicated and hard to sample.

An alternative approach to analyze the reactive trajectories that does not involve either of the mentioned issues is the transition path theory (TPT) [15]. The TPT provides a framework for quantifying rare events and enables us to efficiently generate reactive trajectories. The TPT gives precise answers to the following questions:

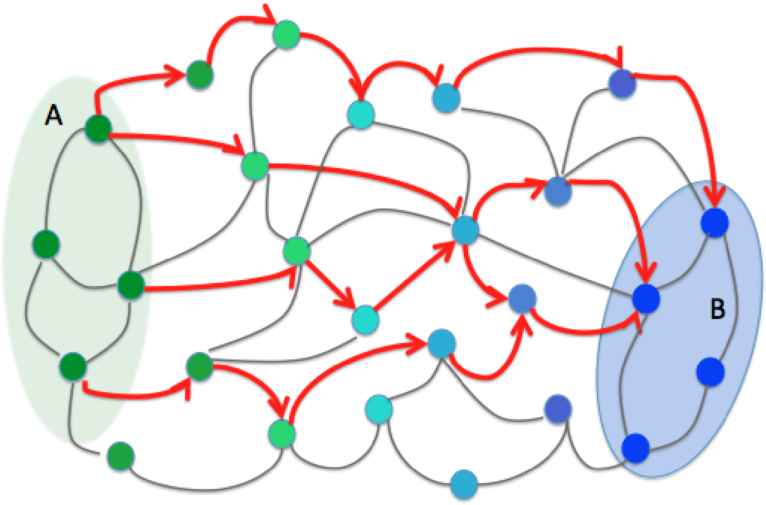


Figure 2.3: Examples of Reactive Trajectories. Given two disjoint subsets denoted A and B, the reactive trajectories (red arrows) are those trajectories that go from A to B without returning to A in-between.

- What is the probability to observe a reactive trajectory at a given state?
- What is the net amount of reactive current going through a given state?
- What is the reaction rate, i.e., the transition rate between two substates, say A (reactant) and B (product)?
- What are the mechanisms of the reactions and how to describe them effectively?

The TPT relates to Bovier and collaborators' [13, 14] works on quantification of transitions in continuous-time and discrete-time Markov chains and built upon the classic potential theory. Contrary to the work of Bovier, TPT does not rely on the assumption the Markov chain is time-reversible.

2.3.2 Basic Concepts of the TPT

We are interested in transitions from one subset $A \subset S$ to another disjoint subset $B \subset S$. The TPT gives a conceptual apparatus for describing statistical properties of the reactive trajectories, i.e., those trajectories that start at A and go to B without returning to A in-between. We define the key functions of the TPT describing the reactive trajectories on the state space S.

The cornerstone functions of the TPT are the forward and backward committors denoted by $q^+ = (q_i^+)_{i\in S}$ and $q^- = (q_i^-)_{i\in S}$ respectively. The answers of the aforementioned questions are given in terms of them. The forward committor is the probability that the process starting at a state $i \in S$ will first reach B rather than A. One can readily check that the forward committor function satisfies the following system of linear algebraic equations:

$$\begin{cases} \sum_{j\in S} L_{i,j} q_j^+ = 0, & i \in S \setminus (A \cup B) \\ q_i^+ = 0, & i \in A \\ q_i^+ = 1, & i \in B \end{cases} \tag{2.7}$$

The backward committor is the probability that the process arriving at state i last came from A rather than B. It satisfies:

$$\begin{cases} \sum_{j\in S} \hat{L}_{i,j} q_j^- = 0, & i \in S \setminus (A \cup B) \\ q_i^- = 1, & i \in A \\ q_i^- = 0, & i \in B \end{cases} \tag{2.8}$$

with $\hat{L}_{i,j}$ being the generator for the time-reversed process. Once the committors are computed, one can express some basic statistical properties of the reactive trajectories.

- The probability distribution of reactive trajectories is given by [16]:

  $$m_i^R = \pi_i q_i^- q_i^+ \tag{2.9}$$

  m_i^R is the probability to find a reactive trajectory at state $i \in S$ at time t. $\sum_i m_i^R$ is equal to the probability for a trajectory to be reactive. Note that this sum: $\sum_{i \in S} m_i^R \leq 1$.

- The probability current of reactive trajectories is defined as the average number of transitions per unit time from state i to j performed by reactive trajectories. It is proven in [16] that it can be expressed as

  $$f_{i,j} = \begin{cases} \pi_i q_i^- L_{i,j} q_j^+, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{2.10}$$

- The effective current gives the net average number of reactive trajectories per time unit making a transition from i to j on their way from A to B. It is defined in [16] as

  $$f_{i,j}^+ = \max\{f_{i,j} - f_{j,i}, 0\} \tag{2.11}$$

- The transition rate from A to B is the average number of transitions per unit time performed by an infinite trajectory:

  $$\nu_{AB} = \lim_{T \to \infty} \frac{N_{AB}(T)}{T}$$

where $N_{AB}(T)$ is the total number of transitions from A to B up to time T. This transition rate is equal to the total reactive current coming out of A which is the same as the total reactive current going into B. It is proven in [16] that:

$$\nu_{AB} = \sum_{i \in A, j \in S} f^{+}_{i,j} = \sum_{i \in A, j \in S} f_{i,j} = \sum_{i \in S, j \in B} f_{i,j} \tag{2.12}$$

It is also proven in [16] that the probability current of reactive trajectories is conserved at each reactive state $i \in (A \cup B)^c$. This theorem is important for deriving our results and is given below.

Theorem 1. *The probability current is conserved at each reactive state i.e.*

$$\sum_{j \in S} (f_{i,j} - f_{j,i}) = 0 \textit{ for all } i \in (A \cup B)^c \tag{2.13}$$

2.3.3 Sampling Techniques for Reactive Trajectories

In 2013, M. Cameron and E. Vanden-Eijnden [17] proposed tools to generate reactive trajectories as well as the special kind of reactive trajectories called no-detour reactive trajectories for time-reversible MJPs. No-detour reactive trajectories for time-reversible networks are those along which the committor function is strictly increasing. Note that, for time-reversible networks, $q^{+}_{i} = 1 - q^{-}_{i}$ and $\hat{L} = L$. The generation of no-detour reactive trajectories relies on the concept of the reactive current given by

$$F_{i,j} := f_{i,j} - f_{j,i}. \tag{2.14}$$

In general, reactive trajectories can be very long, but in the time-reversible case the no-detour reactive trajectories are much shorter. The extension of the concept of the reactive current to the case of time-irreversible Markov chains is not straightforward. In time-reversible MJPs, the effective current accurately describes the reactive current. However in time-irreversible MJPs, the definition of the effective current given in [16] might lead to undesirable consequences such as the reactive current along some edges of a reactive trajectory may exceed the transition rate ν_{AB}. In this dissertation, we resolve this issue and introduce a counterpart of the effective current and no-detour reactive trajectories for time-irreversible Markov chains. Moreover, we propose an algorithm to generate such trajectories and offer its theoretical justification.

2.3.4 Analyzing Markov chains using TPT

One can quantify the transition process in the MJP using the TPT as follows.

1. Solve forward and backward committor equations.

2. Find probability current of reactive trajectories.

3. Find transition rate.

In Chapter 3, we propose a methodology for designing MJPs with desired stationary currents from time-irreversible MJPs and provide their theoretical justification.

Chapter 3: Designing MJP with Desired Stationary Currents

In [17], two modified MJPs were designed for the original time-reversible irreducible MJP. The stationary probability current in the first of them coincided with the probability current of reactive trajectories, and the invariant probability distribution matched the one for reactive trajectories. The second MJP was constructed so that the stationary current in it was equal to the reactive current in the original MJP. These modified MJPs justified the algorithms for generating reactive trajectories and the so-called no-detour reactive trajectories. We remind that the stationary probability current in a MJP is given by

$$J_{i,j} = \pi_i L_{i,j} - \pi_j L_{j,i}.$$

Note that this current is always zero for time-reversible MJPs and nonzero otherwise.

In this chapter, we propose a general framework for designing modified MJPs with desired stationary current and desired invariant distribution and prove a theorem justifying our construction. We introduce a so-called acyclic current that is a counterpart of the reactive current for time-irreversible MJPs. We apply our recipe to time-irreversible networks so that the stationary currents in the modified MJPs are equal to the effective currents and acyclic currents in the original ones.

3.1 Motivation

In many applications, it is of interest to obtain an effective description of the transition process between subsets of states. If the underlying Markov process is time-reversible, the transition process between subsets of interest can effectively be described by the reactive current defined by (2.14). However, the concept of reactive current has not been defined for time-irreversible networks for a reason that we discuss below. Note that Metzner et al. [16], where time-reversibility is not assumed, does not contain the term "reactive current". It introduces only the probability current of reactive trajectories (2.10) and the effective current (2.11). Thus, we propose an extension of the reactive current.

We first consider a simple example that will allow us to highlight some important facts about time-irreversible MJPs and understand how to resolve some issues regarding the definition of the reactive current and reaction pathways. Let us quantify the transitions between the states a and b in the MJP depicted in Fig. 3.1 (top left). We define the time-reversed MJP shown in Fig. 3.1 (top right), and set $A = \{a\}$ and $B = \{b\}$. Then we calculate the forward and backward committors, q^+ and q^-, and compute the probability current of reactive trajectories, $f_{i,j}$. For this MJP, the effective current f^+ coincides with the reactive current. We see that the directed graph $G(S, \{f^+\})$ induced by f^+ contains a cycle, which would never happen for a time-reversible MJP. Moreover, if we follow a single trajectory of this chain starting at A, the effective current along some of its edges may exceed the total effective current emanating from A, i.e., the transition rate. This happens in

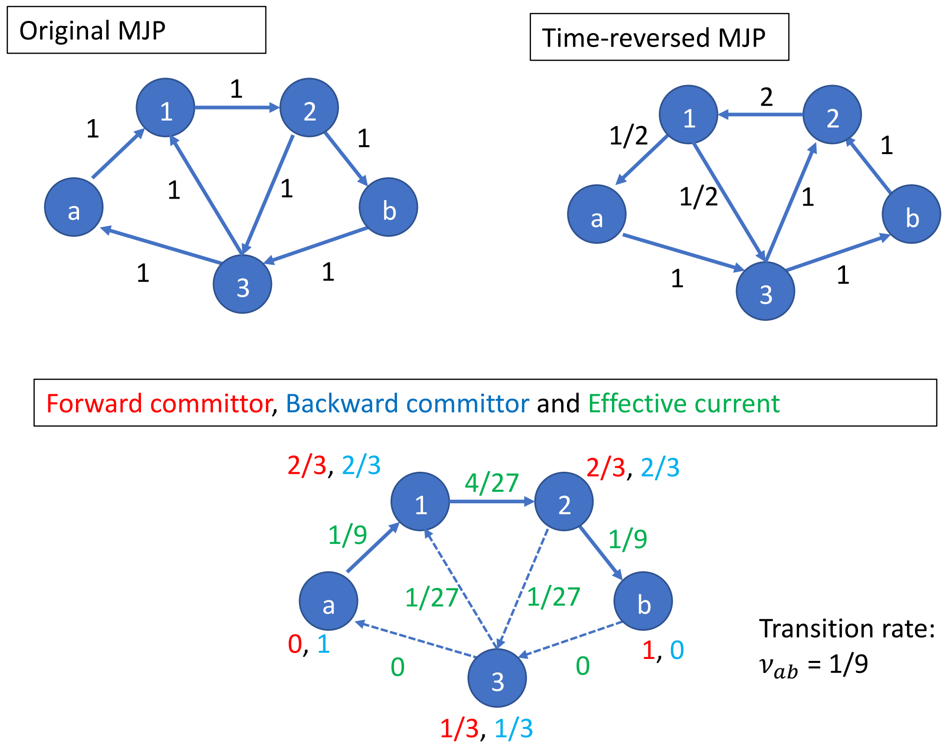


Figure 3.1: An illustrative example. Top left: the original MJP. The black numbers next to the arcs are the $L_{i,j}$'s. Top right: the time-reversed MJP. The black numbers next to the arcs are the $\hat{L}_{i,j}$'s. Bottom: The green numbers next to the arcs are the effective currents. The red and blue numbers next to the states are the forward and backward committor values respectively.

our example: the effective current $f^+_{1,2} = 4/27$ is greater than the total current of 1/9 leaving a and arriving at b (see Fig. 3.1 (bottom)). A similar phenomenon was encountered in the network originating from a budding yeast gene regulatory network (see Chapter 5) and its discovery motivated us to devise a procedure for removing cycles from the graph $G(S, \{f^+\})$. Moreover, we will propose a framework for constructing modified MJPs with desired stationary currents and invariant probability distributions. Propositions 1 and 2 in [17] will be corollaries from our theorem (see Theorem 2 in Section 3.3).

3.2 Setup and Assumptions

We develop a general framework for modifying the original time-irreversible irreducible MJP to make it have a desired stationary current and a desired invariant distribution. Our construction generalized Propositions 1 and 2 in Ref. [17]. Again, we consider a MJP on a finite state space S with infinitesimal generator L. As in [17], we assume that direct jumps from A to B are impossible, i.e.

$$L_{i,j} = 0 \text{ whenever } i \in A \text{ and } j \in B.$$

This allows us to simplify the discussion, however we abandon the other two assumptions from [17]. Let $A, B \subset S$, with $A \cap B = \emptyset$. The set of reactive states will be denoted by $S_R := S \setminus (A \cup B)$. Suppose we have defined a current e satisfying the following properties:

Assumption 1. *Non-negativity:* $e_{i,j} \geq 0$

Assumption 2. *The conservation of current:* $\forall i \in S_R, \sum\limits_{j \in S}(e_{i,j} - e_{j,i}) = 0.$

Assumption 3. *Transition rate:* $\sum\limits_{i \in A}\sum\limits_{j \in S} e_{i,j} = \sum\limits_{i \in S}\sum\limits_{j \in B} e_{i,j} = \nu_{AB}$ *where* ν_{AB} *is the rate from* A *to* B.

It should be noted that both the probability current of reactive trajectories, f, and the effective current, f^+, satisfy assumptions 1 - 3. We will extract the subset of S_R with positive current emanating from them and denote it by R, i.e.,

$$R := \{i \in S_R | \exists j \in S : e_{i,j} > 0\}$$

We will call the states in R the progress states for current e. The rest of the states in S_R will be called the left-out states, and their set will be denoted by $S_0 : S_0 = S_R \setminus R$. As in [17], we combine all the pieces of non-reactive trajectories in the original MJP into an artificial state we call s. We define the process obtained in this way the transition path process originally introduced in the context of SDEs by Lu and Nolen [34]. This definition ensures all trajectories in the reactive process are reactive trajectories.

3.3 Designing Modified MJP

Theorem 2. *Suppose that assumptions 1 - 3 hold. Consider the process on the state space $\tilde{S} = R \cup \{s\}$ defined by the generator M with off-diagonal entries given by*

$$\begin{cases} M_{i,j} = \frac{e_{i,j}}{\mu_i}, & i,j \in R \\ M_{i,s} = \sum\limits_{j \in B} \frac{e_{i,j}}{\mu_i}, & i \in R \\ M_{s,j} = \frac{1}{1-\rho_R} \sum\limits_{i \in A} e_{i,j}, & j \in R \end{cases} \tag{3.1}$$

where $\rho_R = \sum\limits_{i \in R} \mu_i < 1$ and $\mu_i > 0$ for all $i \in R$. Then the desired invariant probability distribution of the transition path process is given by

$$\tilde{\mu}_i = \begin{cases} \mu_i, & i \in R \\ 1 - \rho_R, & i = s \end{cases} \tag{3.2}$$

and the stationary current in the network with state space $\tilde{S}$ and the generator matrix M coincides with the current e in the original network.

Proof. 1. We will first show that the invariant distribution in the modified MJP is given by $\tilde{\mu}$, that is $\sum\limits_{i \in R \cup \{s\}} \tilde{\mu}_i M_{ij} = 0$.

Let $j \in R$. We rewrite the summation as

$$\sum_{i \in R \cup \{s\}} \tilde{\mu}_i M_{ij} = \sum_{i \in R} \tilde{\mu}_i M_{ij} + \tilde{\mu}_s M_{sj}.$$

By removing the jth term from the summation we have

$$\sum_{i \in R \cup \{s\}} \tilde{\mu}_i M_{ij} = \sum_{\substack{i \in R \\ i \neq j}} \tilde{\mu}_i M_{ij} + \tilde{\mu}_j M_{jj} + \tilde{\mu}_s M_{sj}.$$

Using the fact that $M_{jj} = -\sum\limits_{i \neq j} M_{ji}$, we get

$$\sum_{i \in R \cup \{s\}} \tilde{\mu}_i M_{ij} = \sum_{\substack{i \in R \\ i \neq j}} \tilde{\mu}_i M_{ij} - \tilde{\mu}_j \sum_{i \neq j} M_{ji} + \tilde{\mu}_s M_{sj}.$$

Removing the state s from the second summation on the right-hand side gives

$$\sum_{i \in R \cup \{s\}} \tilde{\mu}_i M_{ij} = \sum_{\substack{i \in R \\ i \neq j}} \tilde{\mu}_i M_{ij} - \tilde{\mu}_j \sum_{\substack{i \neq j \\ i \in R}} M_{ji} - \tilde{\mu}_j M_{js} + \tilde{\mu}_s M_{sj}.$$

Next, we factor out $\tilde{\mu}_j$ from the middle two terms and rearrange the order. Thus,

$$\sum_{i \in R \cup \{s\}} \tilde{\mu}_i M_{ij} = \sum_{\substack{i \in R \\ i \neq j}} \tilde{\mu}_i M_{ij} + \tilde{\mu}_s M_{sj} - \tilde{\mu}_j \left(\sum_{\substack{i \neq j \\ i \in R}} M_{ji} + M_{js} \right).$$

We now rewrite the generator M in terms of the current e and invariant distribution μ using 3.1. Thus,

$$\sum_{i \in R \cup \{s\}} \tilde{\mu}_i M_{ij} = \sum_{\substack{i \in R \\ i \neq j}} \mu_i \frac{e_{ij}}{\mu_i} + (1 - \rho_R) \frac{1}{1 - \rho_R} \sum_{i \in A} e_{ij} - \mu_j \left(\sum_{\substack{i \in R \\ i \neq j}} \frac{e_{ji}}{\mu_j} + \sum_{i \in B} \frac{e_{ji}}{\mu_j} \right).$$

By simplifying we obtain

$$\sum_{i\in R\cup\{s\}} \tilde{\mu}_i M_{ij} = \sum_{\substack{i\in R\\ i\neq j}} e_{ij} + \sum_{i\in A} e_{ij} - \sum_{\substack{i\in R\\ i\neq j}} e_{ji} - \sum_{i\in B} e_{ji}.$$

Finally, using assumption 3 for the second and fourth terms and assumption 2 for the first and third terms we have

$$\sum_{i\in R\cup\{s\}} \tilde{\mu}_i M_{ij} = \nu_{AB} - \nu_{AB} = 0.$$

Now let $j = s$. We rewrite the summation as

$$\sum_{i\in R\cup\{s\}} \tilde{\mu}_i M_{ij} = \sum_{i\in R} \tilde{\mu}_i M_{is} + \tilde{\mu}_s M_{ss}.$$

Using the fact that $M_{ss} = -\sum_{i\neq s} M_{si}$, we get

$$\sum_{i\in R\cup\{s\}} \tilde{\mu}_i M_{ij} = \sum_{i\in R} \tilde{\mu}_i M_{is} - \tilde{\mu}_s \sum_{i\in R} M_{si}.$$

Similarly, we rewrite the generator M in terms of the current e and invariant distribution μ using 3.1. Therefore,

$$\sum_{i\in R\cup\{s\}} \tilde{\mu}_i M_{ij} = \sum_{i\in R} \mu_i \sum_{j\in B} \frac{e_{ij}}{\mu_i} - (1-\rho_R) \sum_{i\in R} \frac{1}{1-\rho_R} \sum_{i\in A} e_{ji}.$$

By simplifying we obtain

$$\sum_{i\in R\cup\{s\}} \tilde{\mu}_i M_{ij} = \sum_{i\in R} \sum_{j\in B} e_{ij} - \sum_{i\in R} \sum_{i\in A} e_{ji}.$$

Finally, using assumption 3 we obtain

$$\sum_{i\in R\cup\{s\}} \tilde{\mu}_i M_{ij} = \nu_{AB} - \nu_{AB} = 0.$$

2. Next we show the stationary probability current, $J_{i,j}$ in the MJP with generator matrix M coincides with the current $E_{i,j} = e_{i,j} - e_{j,i}$ for all $i, j \in R$:

$$J_{i,j} = \mu_i M_{i,j} - \mu_j M_{j,i} = e_{i,j} - e_{j,i} = E_{i,j}$$

□

Theorem 2 can be used to generate the reactive trajectories which in turn can be analyzed using statistical tools. In Chapter 4, we develop an algorithm for generating a weighted directed acyclic graph $G(S, \{F^+\})$ where S is the set of vertices inherited from the original MJP, and $F^+ \geq 0$ are the weights for the arcs. The current defined by F^+ possesses properties 1 - 3 and hence we can design a modified MJP with the desired invariant probability distribution.

Next, we show that the transition rate can be computed using an arbitrary cut in the graph $G(S, \{E^+_{ij}\})$ separating the sets A and B. Here we have used the notation $E^+_{ij} := \max\{e_{i,j} - e_{j,i}, 0\}$. Cameron and Vanden-Eijnden [17] defined an AB-cut as any partition or cut in the network $G(S, E)$ such that sets A and B are on different sides of the partition. An AB-cut leads to decomposing S such that $S = S_A \cup S_B$ where $A \subset S_A$, $B \subset S_B$ and $S_A \cap S_B = \emptyset$. As a result, the transition rate, ν_{AB}, can also be expressed in terms of an AB-cut.

Theorem 3. *The transition rate, ν_{AB}, is given by*

$$\nu_{AB} = \sum_{i \in S_A} \sum_{j \in S_B} (e_{ij} - e_{ji}) \tag{3.3}$$

The proof of this statement is as follows:

Proof. For brevity, we will denote $E_{i,j} := e_{i,j} - e_{j,i}$. We will use the fact that for any subset $S' \subset S$,

$$\sum_{i \in S', j \in S'} E_{i,j} = 0. \tag{3.4}$$

Since $S_A = A \cup (S_A \setminus A)$ and $S_B = S \setminus S_A$ we can rewrite the summation as

$$\sum_{i \in S_A} \sum_{j \in S_B} E_{i,j} = \sum_{i \in A \cup (S_A \setminus A)} \sum_{j \in S \setminus S_A} E_{i,j}.$$

By separating the union under the first summation, we obtain

$$\sum_{i \in S_A} \sum_{j \in S_B} E_{i,j} = \sum_{i \in A} \sum_{j \in S \setminus S_A} E_{i,j} + \sum_{i \in S_A \setminus A} \sum_{j \in S \setminus S_A} E_{i,j}.$$

Using the fact that in general $\sum\limits_{i \in A \setminus B} = \sum\limits_{i \in A} - \sum\limits_{i \in B}$, we rewrite the above statement as

$$\sum_{i \in S_A} \sum_{j \in S_B} E_{i,j} = \sum_{i \in A} \sum_{j \in S} E_{i,j} - \sum_{i \in A} \sum_{j \in S_A} E_{i,j} + \sum_{i \in S_A \setminus A} \sum_{j \in S} E_{i,j} - \sum_{i \in S_A} \sum_{j \in S_A} E_{i,j} + \sum_{i \in A} \sum_{j \in S_A} E_{i,j}.$$

We now use the facts that the third summation on the right-hand side above is zero by current conservation and the fourth summation is zero by 3.4. We cancel the second and fifth terms to obtain

$$\sum_{i \in S_A} \sum_{j \in S_B} E_{i,j} = \sum_{i \in A} \sum_{j \in S} E_{i,j} = \nu_{AB}.$$

□

Chapter 4: Generating Acyclic Current

In this chapter, we explore the possibility to extend the notion of the reactive current for time-irreversible MJPs. Unfortunately, as it was shown in chapter 3, the signed effective current is not a good candidate because it can be cyclic and some pathways generated according to it can have larger effective current along some of their edges than the transition rate. To reconcile this issue, we propose a cycle removal algorithm and introduce a so-called acyclic current. The acyclic current will enable us to sample no-detour reactive trajectories introduced in [17]. To find the acyclic current, we use two algorithms from graph theory, namely the Depth-First Search algorithm and Tarjan's algorithm. The details of these algorithms are given in Sections 4.1 and 4.2. Note that we do not call the acyclic current the reactive current because in some cases the acyclic current may be nonunique. We discuss this issue in Section 4.3.1.

4.1 Depth-First Search

Depth-first search (DFS) is a recursive algorithm for traversing or searching tree or graph data structures [35]. Depth-first search starts at an undiscovered vertex and recursively proceeds down a path of adjacent vertices. Once the last node in the path no longer has an adjacent vertex or reaches a discovered vertex, the algorithm then backtracks until it finds an unexplored path, and then explores it. This process continues until all the vertices that are reachable from the original source vertex has been discovered. If any undiscovered vertices remain, then depth-first search selects one of them as a new source, and it repeats the search from that source. The algorithm repeats this entire process until it has discovered every vertex.

```
**Initialization**: Label all vertices in graph $G$ as undiscovered
**Input:** Graph $G$ and a vertex $v$ of $G$
**Output:** All vertices reachable from $v$ labeled as discovered.
**The main body**: DSF$(G, v)$
Label $v$ as discovered
for each vertex $u$ that is adjacent to $v$ do
    if vertex $u$ is not labeled as discovered then
        DSF$(G, u)$
    end
end
```

Algorithm 1: Depth-First Search Algorithm

4.2 Tarjan's Algorithm

Tarjan's algorithm [36] is designed for finding strongly connected components (SCCs) of a directed graph. A strongly connected component of a directed graph $G(V, E)$ is a subgraph $G(V', E')$ with maximal set of vertices V' such that for every pair of vertices u and v in V', there is a directed path from u to v and a directed path from v to u. Tarjan's algorithm implements the depth-first search algorithm (Algorithm 1) to search the graph. Each visited node has two variables associated with it. The first variable, *index*, indicates the order in which the nodes are visited. Once the index is assigned it will never change. The second variable, *lowlink*, represents the smallest index reachable from the current node. Each time a visited node is encountered, the *lowlink* is updated. A stack, S, is used to keep track of the visited nodes. While tracking back from the DFS, if a vertex v's *lowlink* equals its *index* value, it means that none of the vertices v can reach a vertex discovered before v, so all these vertices form a SCC. Thus all vertices up to v need to be popped from the stack. Below is a pseudocode (Algorithm 2) summarizing Tarjan's algorithm.

4.3 Cycle Removal Algorithm

We start with developing an algorithm for generating a weighted directed acyclic graph $G(S, \{f^+\})$ where S is the set of vertices inherited from the original MJP, and $f^+ \geq 0$ is the effective current used for the weights of the arcs. If a MJP is irreversible, the effective current may be cyclic. As a result, the effective current along a single edge along the path from A to B might be larger than the transition rate, ν_{AB}, along some arcs. To ensure that the effective current along a single edge does not exceed the transition rate, we design an algorithm that removes all cycles from the MJP. To find cycles in our graph $G(S, \{f^+\})$, we apply the DFS algorithm. Once we find a cycle, we subtract the minimum current from every edge in the cycle thus breaking this cycle. We continue this process until no more cycles are found. The resulting current is an acyclic current.

```
Input: Directed graph $G(V, E)$ where $V$ is the set of vertices and $E$ is the
  edge list.
Output: Strongly connected component (SCC)
Initialization:$\forall v \in V$ : index[v] := lowlink[v] := 0
count = 0
$S = \emptyset$
The main body: Tarjan(v)
count = count + 1
index[v] := lowlink[v] := count
for each $(v, w) \in E$ do
    if index[w] = 0 then
        Tarjan(w)
        lowlink[v] = min(lowlink[v], lowlink[w])
        else if $w \in S$ then
            lowlink[v] = min(lowlink[v], index[w])
        end
    end
end
if lowlink[v] = index[v] then
    w := S.pop()
    add w to SCC
    while  $index[w] \geq index[v]$ do
        w := S.pop()
        Output the SCC
    end
end
```

Algorithm 2: Tarjan's Algorithm

The computational cost of the cycle removal algorithm depends on the cost of using a depth-first search algorithm. DFS is typically used to traverse an entire graph and takes time $O(|V|+|E|)$ with $|V|$ being the number of vertices in the graph and $|E|$ being the number of edges in the graph. Thus, the worst-case scenario cost of Algorithm 3 is

$$\text{Cost(Algorithm 3)} = O(|E|)\text{Cost(DFS)} = O(|V||E|) + O\left(|E|^2\right). \tag{4.1}$$

A graph on which the cost of Algorithm 3 will be the worst case scenario case is

```
Input: Weighted directed graph $G(S, \{f^+\})$ where S is the set of states and
  $f^+$ is the effective current.
Initialization: Set $\{e\} = \{f^+\}$
The main body:
for $k = 1, 2, ...$ do
    Find cycle, $C_k$, in $G(S, \{e\})$ using DFS algorithm;
    if $C_k$ is nonempty then
        $e_{\min}^{(k)} := \min_{(i \to j) \in C_k} e_{i,j}$;
        for all $(i \to j) \in C_k$ do
            $e_{i,j} := e_{i,j} - e_{\min}^{(k)}$;
        end
        $k = k + 1$;
    else
        Break;
    end
end
$\{F^+\} := \{e\}$;
Output: Weighted directed graph $G(S, \{F^+\})$
```

Algorithm 3: Cycle Removal Algorithm: an algorithm for obtaining an acyclic current from the effective current by means of removing cycles.

shown in Fig. 4.1.

Typically, the cost of Algorithm 3 is much less than Eq. (4.1) since the DFS algorithm terminates as soon as a cycle is found. However, since the number of states in the network under consideration can be large, we are motivated to reduce the cost of this algorithm. To do so, we find the set of strongly connected components (SCCs) of $G(S, f^+)$ using Tarjan's algorithm [36], and then apply Algorithm 3 to each SCC consisting of more than one state separately. Thus, we propose a two-step process for generating an acyclic current.

The cost of Tarjan's algorithm is $O(|V| + |E|)$. Hence, the cost of our two-step

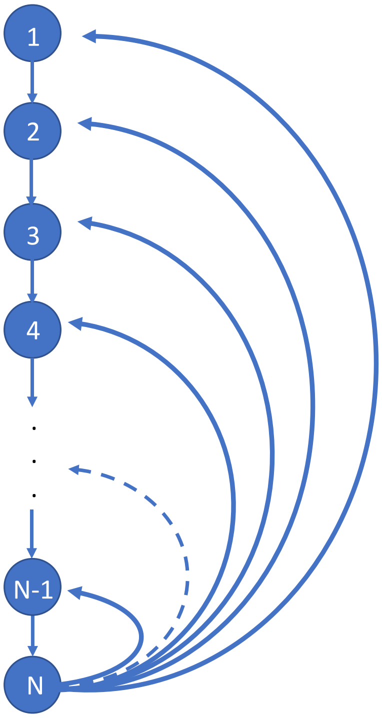


Figure 4.1: Visual example of worst-case scenario cost of Algorithm 3. The given graph has N nodes and $2(N-1)$ edges. We start Algorithm 3 with vertex 1. With one iteration, Algorithm 3 passes through N nodes and N edges before obtaining a cycle. Algorithm 3 will remove a total of $N-1$ cycles and with each iteration we traverse all N vertices and N edges. Thus, the computational cost for this graph will be $(N-1)(N+N) = O(|V||E|) + O(|E|^2)$

```
Initialization Start with a graph $G(S, \{f^+\})$ where $S$ is the set of states
  and $f^+$ is the effective current.
The main body
Step 1: Find all SCCs.
Step 2: Apply cycle removal algorithm to each SCC.
```

Algorithm 4: Two-step process for generating acyclic current: an algorithm for obtaining an acyclic current from the effective current by means of finding the set of SCCs and removing cycles from each of them.

process for finding F^+ does not exceed

$$\text{Cost}\left(\text{Tarjan}\left(G\left(S, \{f^+\}\right)\right)\right) + \sum_{k=1}^{N_{SCC}} N_{f^+}^{(k)} \text{ Cost}\left(\text{DFS}\left(G\left(S^{(k)}, \{f^{+,(k)}\}\right)\right)\right) \tag{4.2}$$

where N_{SCC} is the number of SCCs with more than one state in $G(S, \{f^+\})$, and $G(S^{(k)}, \{f^{+,(k)}\})$ is the k-th SCC with more than one state of $G(S, \{f^+\})$. While either algorithm does give an acyclic current, it may not be unique as we show in Section 4.3.1.

Remark. *This cycle removal algorithm is closely related to the decomposition of flow vectors to the sum of simple paths (see Bertsekas, 1998 [37]). This algorithm decomposes a graph to a set of simple paths by subtracting the minimum flow found in a given simple cycle and terminates when the graph only contains simple paths. A path is said to be simple if it contains no repeated arcs and no repeated nodes. The outputs of the cycle removal algorithm and the simple path decomposition algorithm are different, but their main bodies resemble.*

4.3.1 Non-uniqueness in Cycle Removal Algorithm

Once we have introduced the acyclic current as an output of Algorithm 3, a natural question arises: is the acyclic current unique? Unfortunately, the answer is

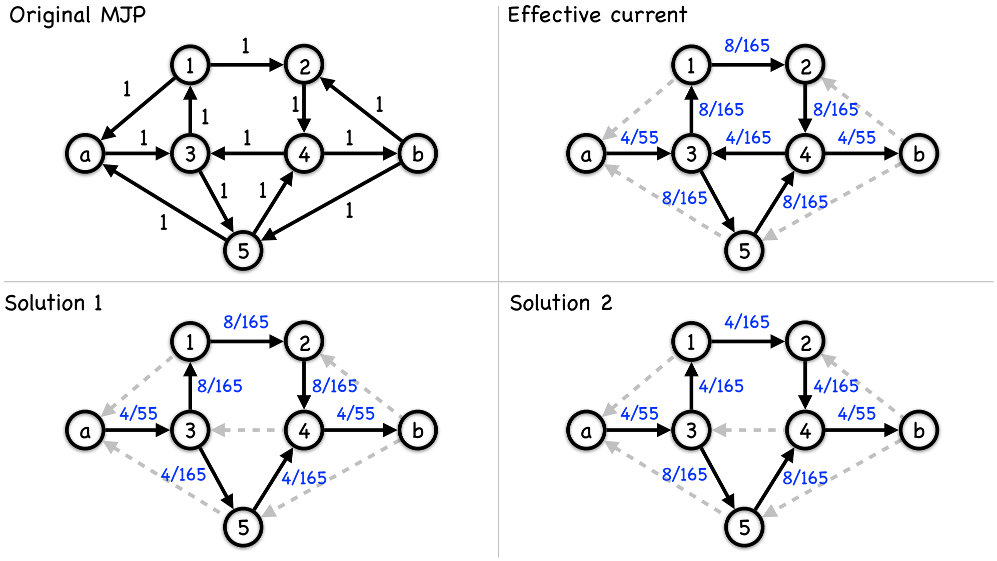


Figure 4.2: An example illustrating nonuniqueness of acyclic currents. Top left: The given MJP. The black numbers next to the arcs are the $L_{i,j}$'s. Top right: The nonzero effective current along each arc is indicated by a blue number. Bottom: The different acyclic currents resulting from the removals of cycles $\{3, 5, 4\}$ and $\{1, 2, 4, 3\}$respectively.

no. The acyclic current obtained by Algorithm 3 or 4 may depend on the order in which the cycles have been removed. This is demonstrated by the example in Fig. 4.2.

Let us consider a MJP with states a, b, and 5 states in between (Fig. 4.2 (top left)). All nonzero off-diagonal entries of the generator matrix L are written next to the corresponding arcs. The invariant probability distribution in this MJP is

$$\pi = \left[\frac{12}{55}, \frac{1}{11}, \frac{9}{55}, \frac{2}{11}, \frac{8}{55}, \frac{7}{55}, \frac{4}{55}\right].$$

The generator matrix $\hat{L}$ of the corresponding time-reversed MJP (defined by Eq. 2.6) is computed. Solving equations 2.7 and 2.8, we find the forward and backward committors:

$$q^+ = \left[0, \frac{1}{3}, \frac{2}{3}, \frac{1}{3}, \frac{2}{3}, \frac{1}{3}, 1\right], \quad q^- = \left[1, \frac{4}{5}, \frac{4}{9}, \frac{4}{5}, \frac{1}{2}, \frac{4}{7}, 0\right]$$

The probability current of reactive trajectories and the effective current shown in Fig. 4.2 (top right) coincide for this example and are cyclic. The two cycles formed in this example are $\{1,2,4,3\}$ and $\{3,5,4\}$. In either case, the minimum current along any arc in either of the cycles is along arc $(4,3)$. Algorithm 3 finds the acyclic current shown in Fig. 4.2 (bottom left) if it eliminates cycle $\{3,5,4\}$. It results in the acyclic current in Fig. 4.2 (bottom right) if it eliminates cycle $\{1,2,4,3\}$. Thus, applying the cycle removal algorithm to this MJP can give two different currents depending on the enumeration of the vertices. While this may be the case, any acyclic current produced by Algorithm 3 still satisfies the conservation of current property as each state $i \in S \setminus (A \cup B)$. However, due to the non-uniqueness, we only refer to this current as an acyclic current and not the reactive current.

4.4 An illustrative example: a discretized Maier-Stein model

We would like to demonstrate the results of our cycle removal methodology on a visual example with a large number of states. To create such an example, we discretize the Maier-Stein SDE [38], a nongradient bistable system, given by:

$$d\mathbf{z} = \mathbf{b}(\mathbf{z})dt + \sqrt{\epsilon}d\mathbf{w}, \quad \mathbf{b}(\mathbf{z}) = \begin{bmatrix} b_x(x,y) \\ b_y(x,y) \end{bmatrix} = \begin{bmatrix} x - x^3 - 10xy^2 \\ -y(1+x^2) \end{bmatrix}, \tag{4.3}$$

where $d\mathbf{w}$ is the standard 2D Brownian motion. We restrict this system to the rectangle $-1.2 \le x \le 1.2$, $-0.6 \le y \le 0.6$ and assign reflective boundary conditions, i.e. the homogeneous Neumann conditions. The generator matrix L is obtained by discretizing the generator for (4.3)

$$\mathcal{L} := b_x(x,y)\frac{\partial}{\partial x} + b_y(x,y)\frac{\partial}{\partial y} + \frac{\epsilon}{2}\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right). \tag{4.4}$$

Next, we obtain an irreversible MJP by discretizing this generator by finite differences on an $N \times N$ mesh. The nonzero off-diagonal entries of L are

$$\begin{aligned}
L_{(i,j),(i+1,j)} &= b_x(x_i,y_j)\frac{1}{2h_x} + \frac{\epsilon}{2h_x^2}, \quad 2 \le i \le N-1,\ 1 \le j \le N,\\
L_{(1,j),(2+1,j)} &= b_x(x_i,y_j)\frac{1}{2h_x} + \frac{\epsilon}{h_x^2}, \quad 1 \le j \le N,\\
L_{(i,j),(i-1,j)} &= -b_x(x_i,y_j)\frac{1}{2h_x} + \frac{\epsilon}{2h_x^2}, \quad 2 \le i \le N-1,\ 1 \le j \le N,\\
L_{(N,j),(N-1,j)} &= -b_x(x_i,y_j)\frac{1}{2h_x} + \frac{\epsilon}{h_x^2}, \quad 1 \le j \le N,\\
L_{(i,j),(i,j+1)} &= b_y(x_i,y_j)\frac{1}{2h_y} + \frac{\epsilon}{2h_y^2}, \quad 1 \le i \le N,\ 2 \le j \le N-1,\\
L_{(i,1),(i,2)} &= b_y(x_i,y_j)\frac{1}{2h_y} + \frac{\epsilon}{h_y^2}, \quad 1 \le i \le N,\\
L_{(i,j),(i,j-1)} &= -b_y(x_i,y_j)\frac{1}{2h_y} + \frac{\epsilon}{2h_y^2}, \quad 1 \le i \le N,\ 2 \le j \le N-1,\\
L_{(i,N),(i,N-1)} &= -b_y(x_i,y_j)\frac{1}{2h_y} + \frac{\epsilon}{h_y^2}, \quad 1 \le i \le N.
\end{aligned} \tag{4.5}$$

We have chosen $N = 70$ and $\epsilon = 0.3$. This combination of parameter values satisfies the requirements that *(i)* ϵ is small enough so that the invariant probability distribution π is bimodal (Fig. 4.3(a)), *(ii)*, for this ϵ, N is large enough so that the off-diagonal entries of L are nonnegative, and *(iii)* N is small enough to make the mesh visual. The subsets A and B are the mesh points lying within the circles of radius 0.2 centered at $(\mp 0.8, 0)$, respectively.

Three reactive trajectories are displayed in Fig. 4.3(b). We see that they tend to wonder around the set of reactive states S_R for a long time prior to reaching

B. The probability density of reactive trajectories is shown by the contour plots. The graph $G(S, \{f^+\})$ induced by the effective current contains four SCCs shown in Fig. 4.3(c). These SCCs were found using Tarjan's algorithm. We have computed an acyclic current F^+ using Algorithm 4. The graph $G(S, \{F^+\})$ is depicted in Fig. 4.3(d), while the intensity of F^+ is displayed in Fig. 4.3(e). Ten samples of no-detour reactive trajectories are shown in Fig. 4.3(f).

Remark. *This example shows that, contrary to the time-reversible case, neither the forward committor increase nor the backward committor decrease along no-detour reactive trajectories. Zoom in Fig. 4.3(d) and observe that (i) directed up blue arcs near $x = y = -0.4$ crossing the level set $q^+ = 0.3$ of the forward committor from larger values to smaller values, and (ii) directed left magenta arcs near $x = 0.5$, $y = \pm 0.6$ crossing the level set $q^- = 0.1$ from smaller values to larger values. Therefore, for time-irreversible MJPs, neither committor serves a good reaction coordinate.*

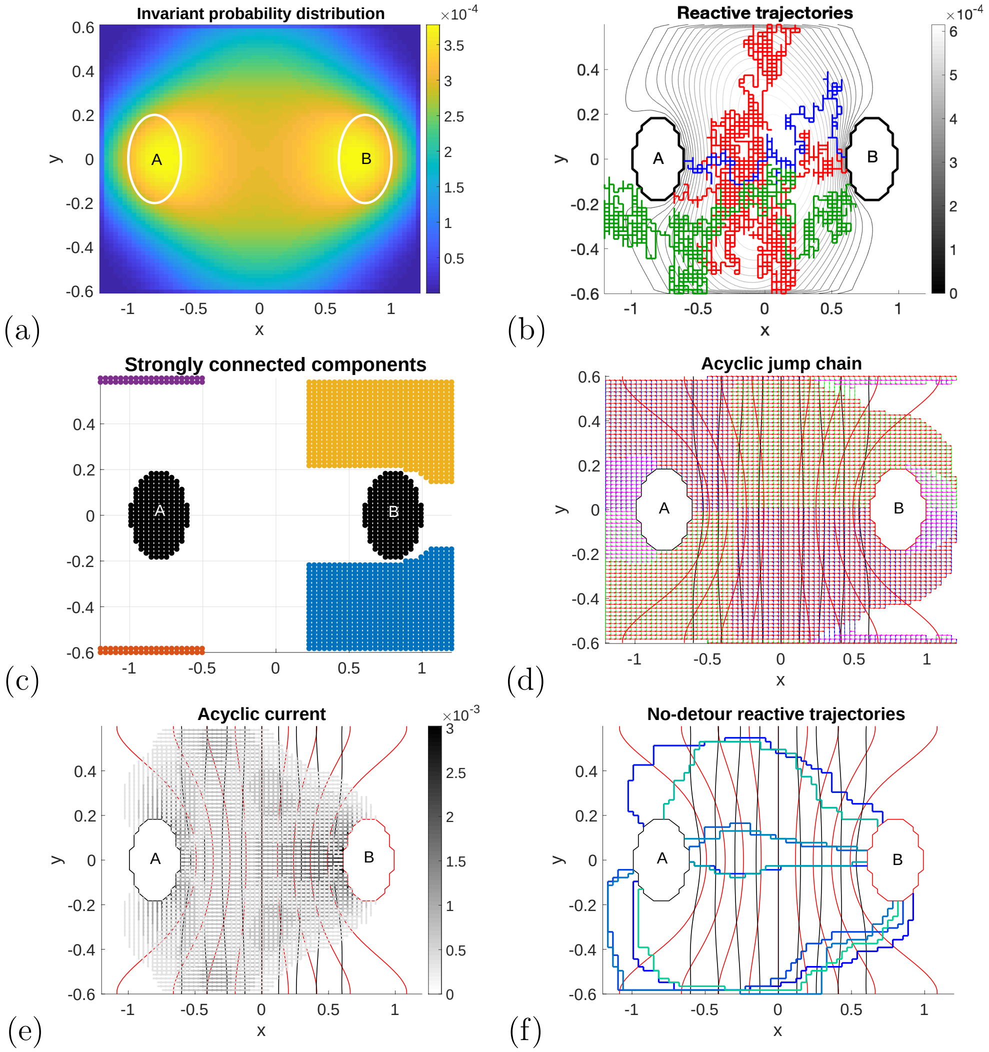


Figure 4.3: (a): The invariant probability distribution π for the Maier-Stein MJP with the generator defined by (4.5). (b): Three reactive trajectories and the level sets of the probability density of reactive trajectories. (c): The four strongly connected components of the graph $G\left(S,\{f^+\}\right)$ are shown by red, blue, purple, and yellow. (d): An acyclic graph $G\left(S,\{F^+\}\right)$. The arcs directed right, left, up, and down are shown by red, magenta, blue, and green arrows respectively. The white areas outside A and B contain the left-out states. The red curves are the level sets of the forward committor, while the black ones are those of the backward committor. (e): The intensity of the acyclic current F^+. (f): Ten no-detour reactive trajectories sampled in the graph $G\left(S,\{F^+\}\right)$.

Chapter 5: Application to Gene Regulatory Network (GRN)

The theoretical and algorithmic developments in Chapters 3 and 4 were motivated by our desire to analyze the dynamics of a stochastic budding yeast gene regulatory network (GRN). Now we will apply this machinery to check which edges of the GRN are essential, and which ones are redundant. In particular, to better understand the effect of stochasticity, we first answer this question for the corresponding deterministic GRN. We start this chapter with presenting a background on the cell cycle and the GRNs.

5.1 Background

5.1.1 Cell Cycle

The cell cycle is a process in which one cell grows leading to the division into two cells. A cell must duplicate all of its components such that the two cells which derive after division each have the information and machinery necessary to repeat the process [25]. The cell cycle consists of four phases. During the first growth phase, known as G1, the cell grows and performs its usual functions. Synthesis or the S phase consists of the DNA in the cell being synthesized and the chromosomes copied.

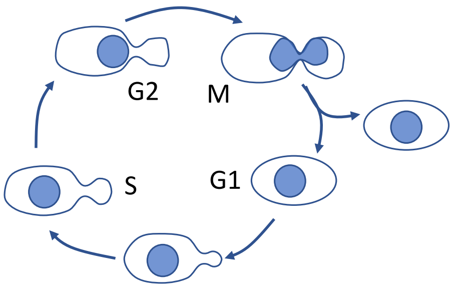


Figure 5.1: Schematic of budding yeast cell cycle.

This is followed by the gap phase, G2, in which the cell prepares for mitosis. Collectively these phases are known as interphase. Finally, during mitosis, or M phase, the chromosomes are separated and the cell divides into two identical cells (see Fig. 5.1). Mitosis can be divided into four subphases: prophase, metaphase, anaphase, and telophase. Chromosomes condense during prophase, align during metaphase, separate during anaphase, and decondense during telophase. After mitosis, each of the new cells enters its own G1 phase and the process repeats.

The cell cycle of budding yeast has been under investigation in many various settings [1,2,25,39,40]. Researchers find budding yeast to be particularly interesting because (1) a great deal is known about the molecular machinery regulating the events of the budding yeast cell cycle, (2) comprehensive mathematical models of the budding yeast cell cycle have been built and studied, (3) the genes and pathways for cell cycle control seem to be very similar in all eukaryotes, including mammals, and (4) the cycle of DNA replication, mitosis, and cell division is crucial to all aspects of biological growth, development, and reproduction [41]. The budding yeast cells are also interesting in that they divide asymmetrically creating a mother and daughter cell after division. Shortly after division, the mother cell will begin a new division process. However, the daughter cell must first grow to a critical size [25].

The cell cycle is mainly controlled by a family of proteins called cyclin-dependent kinases (or CDKs). CDKs interact with proteins called cyclins that are oscillatorily expressed during the cell cycle. The main CDK in yeast is coded by the gene Cdc28. Cdc28 can form a complex with two families of cyclins known as CLNs and CLBs. Once a cell reaches a certain size, a CLN called Cln3 is released and activates Cdc28, initiating all of the processes that lead cell cycle through G1 and into S phase. The rising activity of the Clb1,2/Cdc28 complex induces mitosis. The proper completion of mitosis is facilitated by Cdc20 [25, 40].

There are several checkpoints during the cell cycle of any organism to ensure that the next phase will not occur until the previous one has completed. For a yeast cell, the cell cycle halts at two major checkpoints: (*i*) the G1 checkpoint if DNA damage is detected or the cell has not reached the critical size or (*ii*) the spindle assembly checkpoint if DNA damage is detected, DNA is not replicated completely, or chromosomes are not aligned on the metaphase plate [25, 40]. The models considered here have chosen to include only the G1 checkpoint governed by cell size for simplicity.

5.1.2 Gene Regulatory Network (GRN)

Chen et al. (2000) [25] developed a kinetic model for the cell cycle of budding yeast. This model consisted of 10 nonlinear ODEs, three algebraic equations, and a rule for separating cells at division. There are ∼800 genes involved in the cell cycle

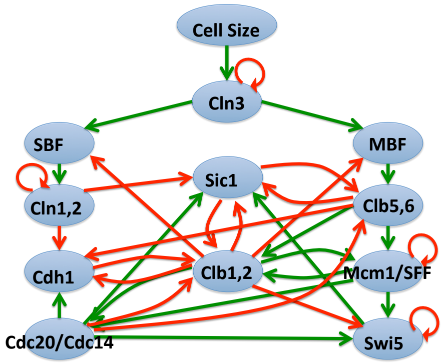


Figure 5.2: Gene regulatory network of budding yeast cell cycle.

of the budding yeast. However, the number of key regulators that are responsible for the control and regulation of this complex process is much smaller. While this model proposed a realistic mechanism for regulating cell division, it involved approximately 50 parameters that needed to be determined to fit experimental observations. In the model, overall cell growth is exponential and the main phases of the budding yeast division cycle are driven by including activities of cyclin-dependent kinases. The 50 parameters used are estimated by trial and error. As a result, [25] only claim the equations and parameter set for the model are sufficient to account for the many properties of cell cycle control. Testing the model on different genotypes of budding yeast, give enough data to provide meaningful confirmation of the model. However, it is not for certain the parameter set is optimal nor do they quantify the robustness in the system.

Li et al. (2004) [1] dramatically simplified the kinetic model proposed by [25] by converting the system of ODEs to a gene regulatory network (GRN). This eliminated the need to estimate the unknown parameters. The GRN consisting of only 11 proteins/ protein complexes was constructed based on literature studies

[39, 41, 42]. A visual of this GRN is shown in Figure 5.2. There are three classes of nodes in the regulatory network:

- Cyclins (Cln1,2, Cln3, Clb1,2, Clb5,6, which bind to the kinase Cdc28);
- Inhibitors and competitors of the cyclin complexes (Sic1, Cdh1, Cdc20, Cdc14);
- Transcription factors (SBF, MBF, Mcm1, SFF, Swi5).

There are three types of edges in the GRN:

- Activation (represented by green arrows);
- Repression (represented by red arrows);
- Self-degradation (represented by self-loops).

There are $n = 11$ nodes in the budding yeast GRN [1]. Let $s \in S$ denote the state of the GRN with S being the set of all possible states of the GRN. Each node i can be in one of two states, active or passive, denoted by $s_i = 1$ or $s_i = 0$ respectively. Thus, s is a vector of 11 components and $|S| = 2^{11} = 2048$. The dynamics of the gene regulatory network is defined by (i) the influence matrix A (5.1) and (ii) the transition rule (5.3). The 11×11 influence matrix $A = (a_{i,j})$ is defined as follows:

$$a_{i,j} = \begin{cases} 1, & \text{if there is green arrow from } j \text{ to } i \\ -1, & \text{if there is red arrow from } j \text{ to } i \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

Table 5.1: The 11 different protein/ protein complexes, node identification and their function.

Protein/ Protein complexes	Node#	Function
Cln3	1	G1-cyclins initiating Start event.
SBF	2	Transcription factor for Cln1,2.
MBF	3	Transcription factor for Clb5,6.
Cln1,2	4	Cyclins involved in budding.
Sic1	5	Stoichiometric inhibitor of Cdc28/Clb2 and Cdc28/Clb5.
Clb5,6	6	B-type cyclins appearing late in G1, involved in DNA synthesis.
Cdh1	7	Activator of the APC; protein involved in Clb2 proteolysis.
Clb1,2	8	B-type cyclin essential for mitosis, present in S/G2/M phase.
Mcm1/SFF	9	Transcription factor for Clb2, Cdc20 and Swi5.
Cdc20/Cdc14	10	Activator of the APC; protein involved in Clb2, Clb5 and proteolysis, and required for exit from mitosis. Phosphatase required for exit of mitosis.
Swi5	11	Transcription factor for Sic1.

For each state $s \in S$, we compute the influence vector

$$v := As \tag{5.2}$$

and define transition of the node state at the next time step $t + 1$ by the following rule.

$$s_i(t+1) = \begin{cases} 1, & \text{if } v_i > 0 \\ 0, & \text{if } v_i < 0 \\ s_i(t), & \text{if } v_i = 0 \end{cases} \tag{5.3}$$

where t is a non-negative integer and time steps are discrete. Throughout this dissertation, this model will be referred to as the deterministic model.

The temporal evolution of the protein states, presented in Table 5.2, follows the cell-cycle sequence. Highlighted in each row are the necessary protein/ protein complexes that must be activated in order to be in that particular phase as described

Table 5.2: Protein state over time determined by the deterministic model reproduced from Li et al. [1]. Colors denote the different phases of the biological pathway. *Row* indicates the time step, *Phase* indicates the cell-cycle phase, and the remaining columns represent the protein state at the given time.

Row	Cln3	SBF	MBF	Cln1,2	Sic1	Clb5,6	Cdh1	Clb1,2	Mcm1	Cdc20	Swi5	Phase
									SFF	Cdc14		
1	1	0	0	0	1	0	1	0	0	0	0	START
2	0	1	1	0	1	0	1	0	0	0	0	G1
3	0	1	1	1	1	0	1	0	0	0	0	G1
4	0	1	1	1	0	0	0	0	0	0	0	G1
5	0	1	1	1	0	1	0	0	0	0	0	S
6	0	1	1	1	0	1	0	1	1	0	0	G2
7	0	0	0	1	0	1	0	1	1	1	0	M
8	0	0	0	0	0	0	0	1	1	1	1	M
9	0	0	0	0	1	0	0	1	1	1	1	M
10	0	0	0	0	1	0	0	0	1	1	1	M
11	0	0	0	0	1	0	1	0	0	1	1	M
12	0	0	0	0	1	0	1	0	0	0	1	G1
13	0	0	0	0	1	0	1	0	0	0	0	G1*

in Section 5.1.1. The cell cycle starts with a signal from "cell size" that excites the stationary growth state G1 resulting in the state called "START". The states of the 11 nodes in the GRN at the START state are listed in row 1 of Table 5.2. The cell cycle transforms back to the stationary G1 state, denoted as G1* in row 13 of Table 5.2, through a sequence of states determined by the deterministic model. We will refer to this temporal evolution as the biological pathway. The phases of the biological pathway are denoted with different colors: purple, the start of the cell cycle; blue, the other G1 states; green, the S state; yellow, the G2 state; and red, the M states.

The main result of [1] is that with the use of a simple deterministic model, the GRN representing the budding yeast cell cycle is stable and robust. The biological stationary state G1* is the attractor with the largest basin size. This property is largely preserved with respect to small perturbations to the dynamic trajectories of all 1,764 protein states that flow into the stationary state G1*. Particularly, when either removing edges, adding edges or switching edges from activating to inhibiting (or vice versa), the relative changes of the basin size of the biggest attractor are small (Fig. 4 in [1]). This was also compared to ones obtained from the ensemble of random networks. While Li et. al. looked at the effect of edge deletion to the basin size of the largest attractor, they left unaddressed the exact changes to the biological pathway caused by these perturbations.

5.1.3 Dynamical Network

For any GRN with n nodes, the dynamical network is the graph with 2^n vertices representing all possible states, and edges representing transitions between the states. Since the evolution of the network over time has the Markov property, edges in the dynamical network define a transition probability of a Markov chain. The dynamical network gives us a way to quantify most likely transitions between cell states which is the key idea behind TPT.

In order to understand the dynamical network of the deterministic GRN we have reproduced the results from [1]. Since we are looking at the deterministic model of this GRN, each cell state in our dynamical network has a unique outgoing arc, i.e. there is a unique possible transition from each state. Hence, the dynamical network is a directed graph whose connected components have at most one loop. Its exploration by the DFS algorithm shows that this graph is forest-like consisting of 7 subtrees.

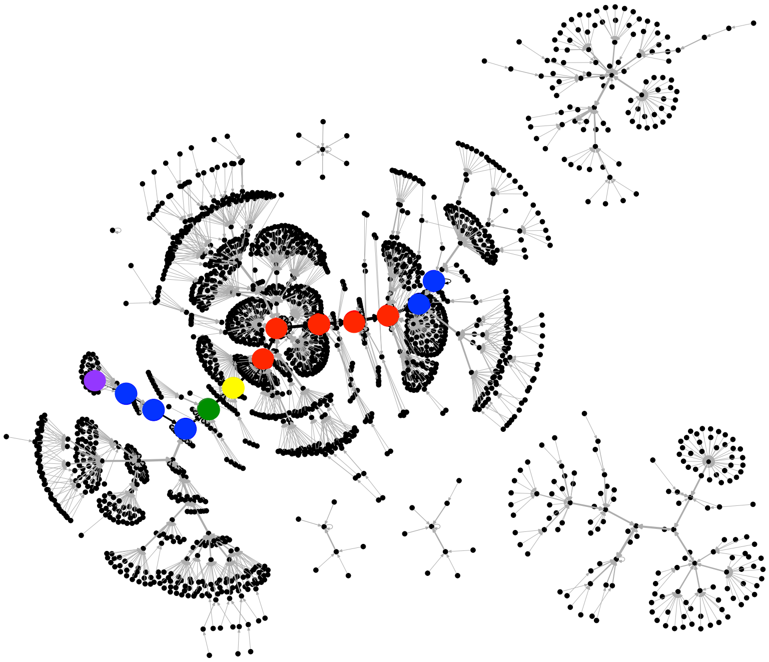


Figure 5.3: The dynamical network for the deterministic GRN that matches the network in Fig. 2 in [1]. The pathway corresponding to the cell cycle from the START state to G1* are shown with large dots. Their colors correspond to different phases of the cell cycle: purple, START; blue, other G1 states; green, S state; yellow, G2 state; and red, M states.

The results of the DFS algorithm are shown in Fig. 5.3. Each cell state is represented by a black node, with the arrows between them indicating dynamic flows from one state to another. The phases of the biological pathway are denoted with different colors: purple, START state; blue, the other G1 states on the biological pathway; green, the S state; yellow, the G2 state; red, the M states.

The discrete-time dynamics of the gene regulatory network defined by the deterministic rules in [1] has seven attractors (stationary states) of the network. Hence, all of the initial states eventually flow into one of the seven stationary states shown in Table 5.3. Dynamic trajectories starting from 1764 out of the 2048 possible states, end up at one particular attractor, G1*. The cell's stationary state being

Table 5.3: Fixed points and basin size determined by deterministic model. Each fixed point is found on a given row. *Sink #* is the row index, *Size* is the size of the basin of attraction, and the remaining columns represent the protein state of the fixed point. This table coincides with Table 1 from [1] as it should.

Sink #	Cln3	SBF	MBF	Cln1,2	Sic1	Clb5,6	Cdh1	Clb1,2	Mcm1	Cdc20	Swi5	Size
									SFF	Cdc14		
1	0	0	0	0	0	0	0	0	0	0	0	7
2	0	1	0	1	0	0	0	0	0	0	0	151
3	0	0	0	0	1	0	0	0	0	0	0	9
4	0	0	1	0	1	0	0	0	0	0	0	7
5	0	0	0	0	0	0	1	0	0	0	0	1
6	0	0	0	0	1	0	1	0	0	0	0	1764
7	0	0	1	0	1	0	1	0	0	0	0	109

the most common attractor of the network insures the stability of the cell state is guaranteed. While this model accurately depicts the cell cycle, it does not capture the intrinsic randomness of this biological process. Thus, a stochastic model was introduced.

5.1.4 Stochastic model of GRN

Zhang et al. (2006) [2] modified the deterministic model [1] by making the transition rules stochastic. We will refer to this model as the stochastic model or stochastic GRN. The stochastic model builds upon the deterministic model by taking into account certain degrees of unpredictability or randomness that may happen due to the environment and allows for the model to self-organize. Stochasticity plays a fundamental role in biological processes. At the intracellular level for example, randomness can be caused by low copy numbers of chemical reactants and an inhomogeneous distribution of the chemical reactants inside the cell [43]. This randomness occurring in biological processes justifies the use of a stochastic approach to investigate them.

As in the deterministic model, the time steps here are logic steps that depend on the current state of each node, s_i, rather than depict actual times. We compute the influence vector v as defined by Eq.(5.2). Under these assumptions, the transition probability is given by:

$$\mathbb{P}\{s_1(t+1), ..., s_{11}(t+1) | s_1(t), ..., s_{11}(t)\} = \prod_{i=1}^{11} \mathbb{P}\{s_i(t+1) | s_1(t), ..., s_{11}(t)\} \quad (5.4)$$

where

if $v_i \neq 0$

$$\mathbb{P}\{s_i(t+1) = 1 | s_1(t), ..., s_{11}(t)\} = \frac{e^{\beta v_i}}{e^{\beta v_i} + e^{-\beta v_i}} \quad (5.5)$$

$$\mathbb{P}\{s_i(t+1) = 0 | s_1(t), ..., s_{11}(t)\} = \frac{e^{-\beta v_i}}{e^{\beta v_i} + e^{-\beta v_i}} \quad (5.6)$$

if $v_i = 0$

$$\mathbb{P}\{s_i(t+1) = s_i(t) | s_1(t), ..., s_{11}(t)\} = \frac{1}{1 + e^{-\alpha}} \quad (5.7)$$

As opposed to the deterministic model, in this model if the protein i has a self-degradation loop, $a_{ii} = -0.1$ as in [2]. The positive number β in the system is an inverse temperature-like parameter which accounts for random factors coming from environment. The positive number α is used to characterize the stochasticity in the system when the input to a node is zero. This parameter is important in controlling the likelihood for a protein to maintain its state when there is no input to it from surrounding nodes. Note that, when $\beta, \alpha \to \infty$, these transition rules converge to the deterministic rules in [1].

The main result of [2] is that in addition to the dynamical and structural stability of the GRN found by [1], the GRN is also stable against stochastic fluctuations for a wide range of "temperatures" characterized by the parameter β. For large β values, which corresponds to low "temperatures", the stationary state G1* is the most probable state of the system. For small β values or high "temperatures", the system behaves randomly and cannot carry out the main biological function. This transition temperature is found at around $\beta \approx 1.03$. While they found values for β for the model to be stable, they did not look at how perturbations in the GRN, particularly through edge deletion, effects the main biological function of the system.

5.2 A "Mutation Analysis" of the Deterministic GRN

5.2.1 Identifying Redundant Edges

We've developed the following algorithm in order to identify redundant edges in the GRN and test the robustness of the model. We distort the GRN shown in Fig. 5.2 by removing one edge and adjust the influence matrix A defined by Eq.(5.1). We apply the transition rules of Eq.(5.3) to obtain the dynamical network and explore it using the DFS. We keep track of the number and sizes of the subtrees of the dynamical graph and check if there is a pathway corresponding to the cell cycle going from START to G1*. If such a pathway exists, we compare it to the one in the original dynamical network and mark changes in it if any. We then repeat this process for all edges of the regulatory network in order to determine which edges can be removed without destroying the cell cycle.

As shown in Table 5.4, we have established that 11 of the 34 edges of the GRN in Fig. 5.2 can be removed without destroying the cell cycle. The removal of

```
Initialization  Start with the influence matrix $A$ for the GRN defined by
  (5.1). Each nonzero entry in $A$ corresponds to an edge in the GRN.
Result: List of pathways from any node in the dynamical network to a
        stationary state.
The main body
for each edge in the GRN do
    Step 1: Make the corresponding entry a 0 in the influence matrix $A$.
    Step 2: Update the transition rules defined by (5.3).
    Step 3: Run DSF algorithm to find all pathways from "START" to
    G1* carrying at least 20% of the normalized current.
end
```

Algorithm 5: Edge Removal Algorithm for Deterministic GRN.

these edges will have little to no effect on the main cycle described in Table 5.2 and may only affect the number of attractors and number of states in their basins. One reason for the little to no effect is due to the fact that the roles of certain proteins tend to overlap. Particularly, ones in the families of cyclins (CLNs and CLBs). Any one of the pairs of cyclins (Cln1,2, Clb1,2, Clb5,6) can do the essential jobs of the other two if the cell is large enough. For example, Clb1,2 can trigger DNA synthesis in the absence of Clb5,6 [25]. This explains why the removal of edges which have no effect all involve at least one member of the family of cyclins. Removal of edge Mcm1/SFF → Clb1,2 and edge Mcm1/SFF → Cdc20/Cdc14 both involve reducing the number of steps in the mitotic phase. Hence the removal of these edges does not destroy the cell cycle but in fact decreases the number of time steps in the cell cycle. The removal of the remaining edges in Table 5.4 all have a slight change in one or two of the rows but all still have the necessary proteins active for each of the cell cycle phases. Thus, the removal of any of these edges does not effect the overall biological pathway.

Table 5.4: The edges of the gene regulatory network that can be removed without destroying the cell cycle modeled by the deterministic model. N_{sinks} represents the number of attractors and N_{G1^*} represents the number of states converging to the stationary state G1*. The effect on the cell cycle corresponds to differences in protein state over time shown in Table 5.2 and s refers to the cell state (i.e. $s_i = 1$ or 0 if protein i is active or inactive). "Row" corresponds to the row in Table 5.2 and we show how a row is modified according to the edge removal.

	Removed edge	**Effect on the main cycle**	N_{sinks}	N_{G1^*}
1	Clb5,6 → Sic1	No effect	7	1747
2	Clb5,6 → Cdh1	No effect	8	1738
3	Clb5,6 → Mcm1/SFF	Row 6: $s = [01110101000]$	7	1734
4	Cdh1 → Clb1,2	No effect	7	1819
5	Clb1,2 → Sic1	No effect	7	1761
6	Clb1,2 → Cdh1	Row 9: $s = [00001011111]$ Row 10: $s = [00001010111]$	7	1760
7	Clb1,2 → Mcm1/SFF	Row 9: $s = [00001001011]$ Row 10: $s = [00001000010]$ Skips Row 11	7	1635
8	Clb1,2 → Cdc20/Cdc14	No effect	7	1814
9	Clb1,2 → Swi5	Row 7: $s = [00010101111]$	7	1764
10	Mcm1/SFF → Clb1,2	Skips Row 9	8	1567
11	Mcm1/SFF → Cdc20/Cdc14	Skips Row 11	7	1650

Table 5.5: The edges of the gene regulatory network that when removed do not include all phases of the cell cycle when modeled by the deterministic model. N_{sinks} represents the number of attractors and N_{G1^*} represents the number of states converging to the stationary state G1*. The effect on the cell cycle corresponds to differences in protein state over time shown in Table 5.2 and s refers to the cell state (i.e. $s_i = 1$ or 0 if protein i is active or inactive). "Row" corresponds to the row in Table 5.2 and we show how a row is modified according to the edge removal.

	Removed edge	**Effect on the main cycle**	N_{sinks}	N_{G1^*}
1	Cln1,2 → Cdh1	Between Row 3 and Row 8: $s = [01110010000]$ $s = [01110110000]$ $s = [01110100100]$ $s = [01110101111]$ $s = [00010101111]$	9	1755
2	Sic1 → Clb5,6	Between Row 2 and Row 8: $s = [01111110000]$ $s = [01110100100]$ $s = [01110101111]$ $s = [00010101111]$	8	1836
3	Clb5,6 → Clb1,2	Between Row 5 and Row 8: $s = [01110100100]$ $s = [01110101111]$ $s = [00010101111]$	8	1585

Table 5.5 shows the 3 edges that when removed lead to the skip of the gap phase G2. While the cell states when modeled by the deterministic model still evolve from START to G1*, the states do not go through all phases of the cell cycle as described in Section 5.1.1. In the removal of each of these edges, the cell goes from G1, to S phase, to M phase and finally to stationary G1* but never goes through G2. Specifically, the protein Cdc20/Cdc14 becomes active sooner than expected which causes the cell to go directly from synthesis to mitosis. All other proteins do become active in the correct sequence.

Removal of any of the other 20 edges results in the absence of the main cycle. This means that starting from any state s, the pathway including the START state does not reach G1*. For example, the removal of the edge Cln1,2 $\rightarrow$ Cln1,2 in particular results in a biological pathway consisting of G1 states, S state, G2 state, and stopping at M state. This would indicate that the cell grows, stops in mitosis and never reaches G1* in order to restart the cell cycle. Since the inactive Cln1,2 protein is essential for completion of mitosis and has no neighboring proteins to repress its expression level, it forces the cell to stay in mitosis.

5.3 A "Mutation Analysis" of the Stochastic GRN

5.3.1 Exploring Dynamical Network

The stochastic transition rules (Eqs.(5.4)-(5.5)) create a dense transition matrix P. This results in a complete dynamical network with the degree of each vertex being 2048 and exploring the graph using DFS would take too long to search in its

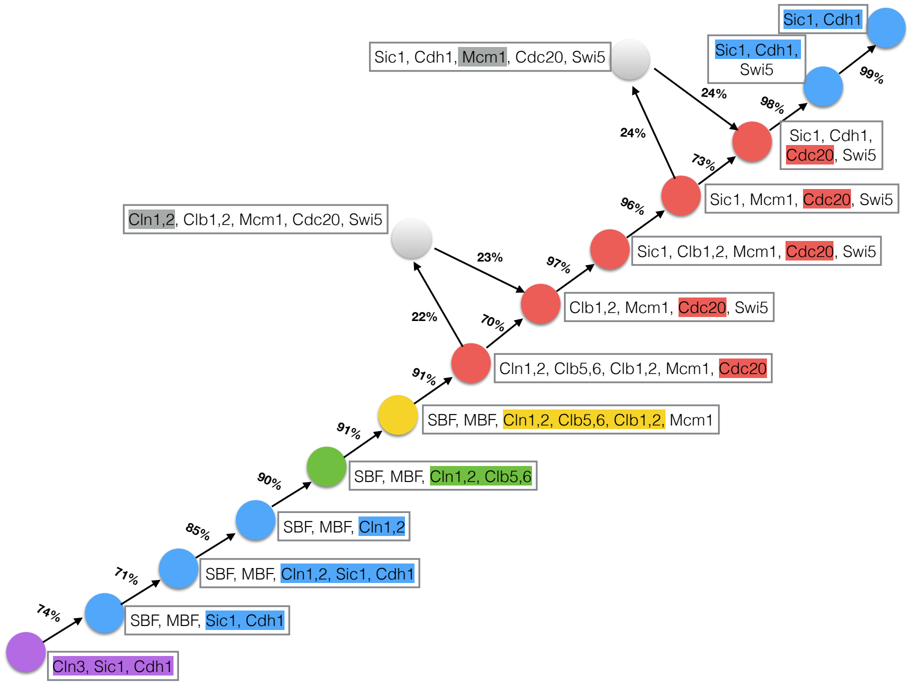

Sic1, Cdh1
Sic1, Cdh1, Swi5
99%
Sic1, Cdh1, Mcm1, Cdc20, Swi5
24%
98%
24%
Sic1, Cdh1, Cdc20, Swi5
73%
Sic1, Mcm1, Cdc20, Swi5
96%
Cln1,2, Clb1,2, Mcm1, Cdc20, Swi5
Sic1, Clb1,2, Mcm1, Cdc20, Swi5
23%
97%
Clb1,2, Mcm1, Cdc20, Swi5
22%
70%
Cln1,2, Clb5,6, Clb1,2, Mcm1, Cdc20
91%
SBF, MBF, Cln1,2, Clb5,6, Clb1,2, Mcm1
91%
SBF, MBF, Cln1,2, Clb5,6
90%
SBF, MBF, Cln1,2
85%
SBF, MBF, Cln1,2, Sic1, Cdh1
71%
SBF, MBF, Sic1, Cdh1
74%
Cln3, Sic1, Cdh1

Figure 5.4: Acyclic current through cell cycle for $\alpha = 5$, $\beta = 6$ using stochastic model. Using Algorithm 3, 284,460 cycles were removed. As in Table 5.2, the phases of the biological pathway are denoted with different colors: purple, the start of the cell cycle; blue, the other G1 states; green, the S state; yellow, the G2 state; and red, the M states.

entirety. We want to extract only the most likely transitions to occur that give all phases of the cell cycle. Thus, to analyze the dynamical network, we use the transition path theory (TPT) (Section 2.3) to single out two specific sets of nodes and analyze the statistical properties of the reactive trajectories by which transitions between these sets occur.

We first preprocess the dynamical network to reduce the computational efforts. While the transition matrix of the dynamical network is dense, most of the pairwise rates have extremely low probabilities. This is because even after introducing stochasticity to the model, transition between certain cell states are still very unlikely to occur. We eliminate all probabilities less than 10^{-14} which leads to a

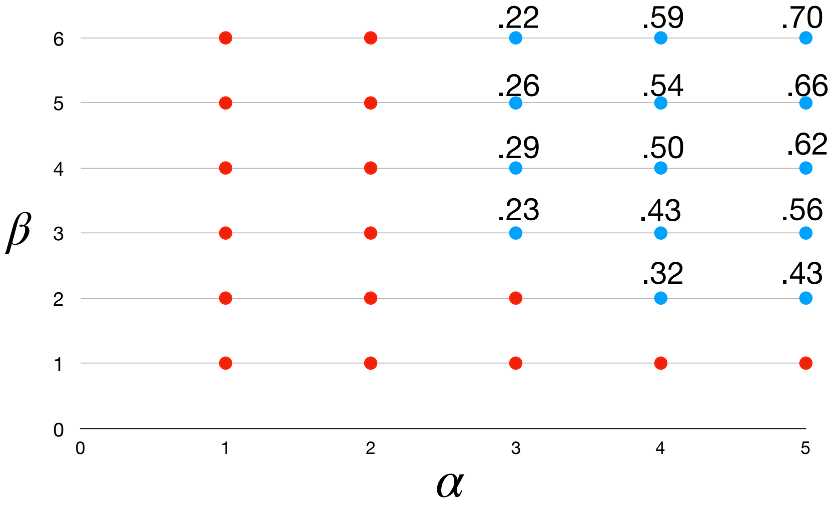


Figure 5.5: The minimax current in pathways from START to G1* for various α and β using stochastic model. Blue dots: pathway carrying a minimum of 20% of the current has been produced. Red dots: no pathways carrying a minimum of 20% was produced. The numbers near the blue dots indicate the minimax current for that particular α and β.

better conditioning of the generator matrix. This decreases the number of edges in our dynamical network from 4,194,304 to 655,195.

We compute the necessary quantities in order to use TPT to analyze the network. Let the set A consist of the single START state and let the set B consist of the single G1* state. We compute the forward and backward committors, $q^+ = (q_i^+)_{i \in S}$ and $q^- = (q_i^-)_{i \in S}$. Using the committor functions we compute the probability current of the reactive trajectories, $f_{i,j}$, and the effective current, $f_{i,j}^+$. The effective current is cyclic and thus we use the cycle removal algorithm (Algorithm 3) to compute an acyclic current. Tarjan's algorithm showed that the network $G(S, \{f^+\})$ consists of only 3 SCCs, namely: excited G1 state; stationary G1* state; and all remaining 2046 states. Hence, we cannot reduce the cost of Algorithm 3 by preprocessing its input with Tarjan's Algorithm.

Finally, we use DFS on the graph $G(S, \{e^+\})$, where e_{ij} is an acyclic current carried from i to j, in order to find pathways only from START to G1*. We normalize the current by dividing each entry by the total current leaving A. This ensures that the total current leaving A sums to 1. We refer to this as the normalized current. Since the graph $G(S, \{e^+\})$ has many pathways from START to G1*, we chose a threshold of 20% to extract only those pathways that carry significant amount of the normalized current. Using a threshold less than 20% gave many pathways and made it difficult to account for differences between the pathways.

Figure 5.4 shows the pathway of cell states that carries at least 20% of the normalized acyclic current for parameters $\alpha = 5$ and $\beta = 6$ in Eq.(5.5). Parameter values $\alpha = 5$ and $\beta = 6$ were chosen in order to compare our results to that of Fig. 3 in [2]. We removed 284,460 cycles from the graph $G(S, \{f^+\})$ using Algorithm 3 to obtain an acyclic current. Figure 5.4 shows that the pathway in which most of the current is carried is the biological pathway of the cell cycle shown in Table 5.2.

Figure 5.5 gives a table of the minimax current for various α and β values. For a given α and β, we find the minimax current by finding the minimum current in a particular pathway, then finding the maximum min current from all pathways. For larger α and β, the minimax current is also larger. This is as expected since again, as α and $\beta \to \infty$, the stochastic model recovers the deterministic model and most of the current will be carried along the main biological pathway (Table 5.2).

5.3.2 Addressing the Issue of Nonuniqueness of the Acyclic Current.

Section 4.3.1 illustrates an example where Algorithm 3 does not give a unique solution for the acyclic current. To reiterate, the acyclic current obtained by Algorithm 3 depends on the order in which the cycles are removed. In this section, we investigate the issue of nonuniqueness by applying Algorithm 3 to our stochastic dynamical network where we randomly change the order in which the cycles are removed.

Parameter values $\alpha = 5$ and $\beta = 6$ were chosen in order to compare with Fig. 5.4. These values were used primarily by Zhang et al. [2] as they are large enough so that the stochastic model behaves closely to the deterministic model while still adding some stochasticity. In order to remove cycles in a different order,

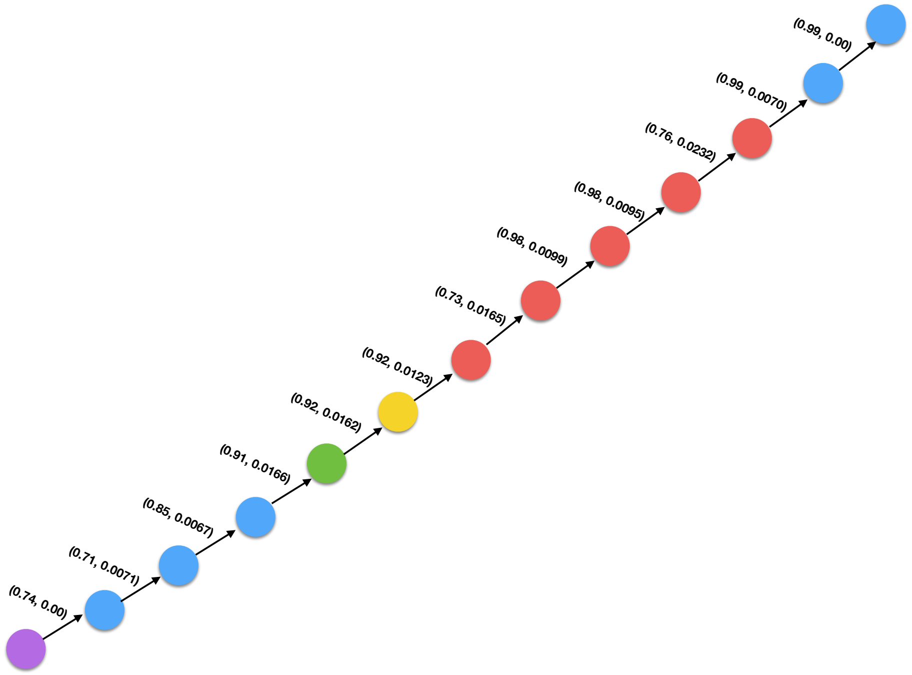


Figure 5.6: Biological pathway found after 20 iterations of removing cycles in different order. The values on each edge are the mean and standard deviation of the acyclic current respectively.

we randomly permute the indexes of the states of the dynamical network. Using a permutation matrix Q, we compute

$$f^+_{\text{permuted}} = Q^T f^+ Q.$$

We then run our cycle removal algorithm on f^+_{permuted}, obtain our original enumeration using

$$f^+ = Q f^+_{\text{permuted}} Q^T,$$

and run depth first search to find pathways. This was repeated 10 times.

Figure 5.6 illustrates the result of this nonuniqueness analysis. For each random permutation of rows and columns, the main biological pathway was found. Next to each edge in the main biological pathway is the mean and standard deviation of currents through each edge over all permutations. The largest standard

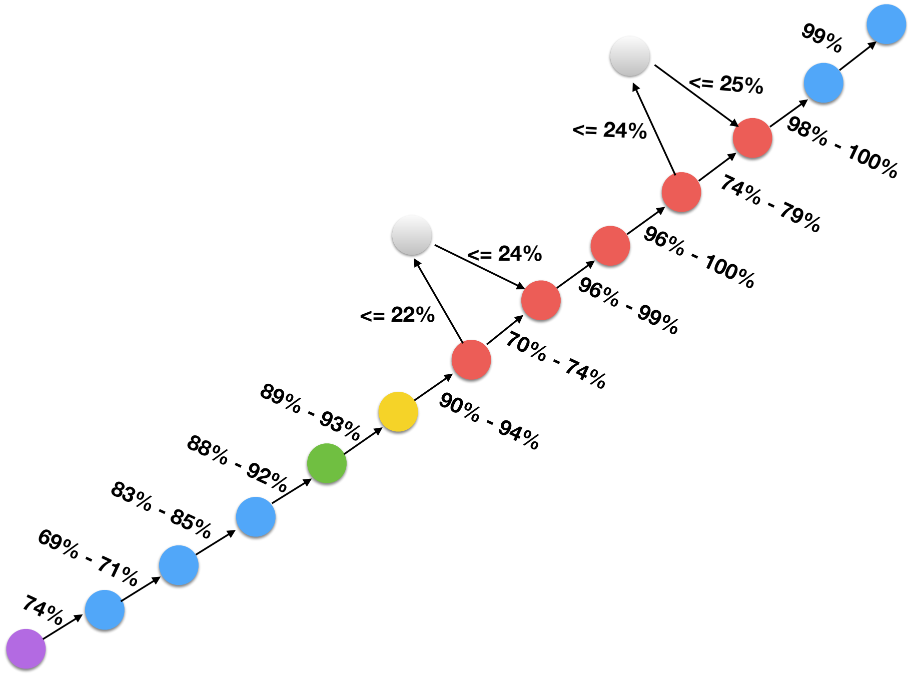


Figure 5.7: All pathways from START to G1* found after 20 iterations of removing cycles in different orders. The values on each edge are the range of acyclic current values found (minimum value - maximum value) after applying Algorithm 3 on the permuted network.

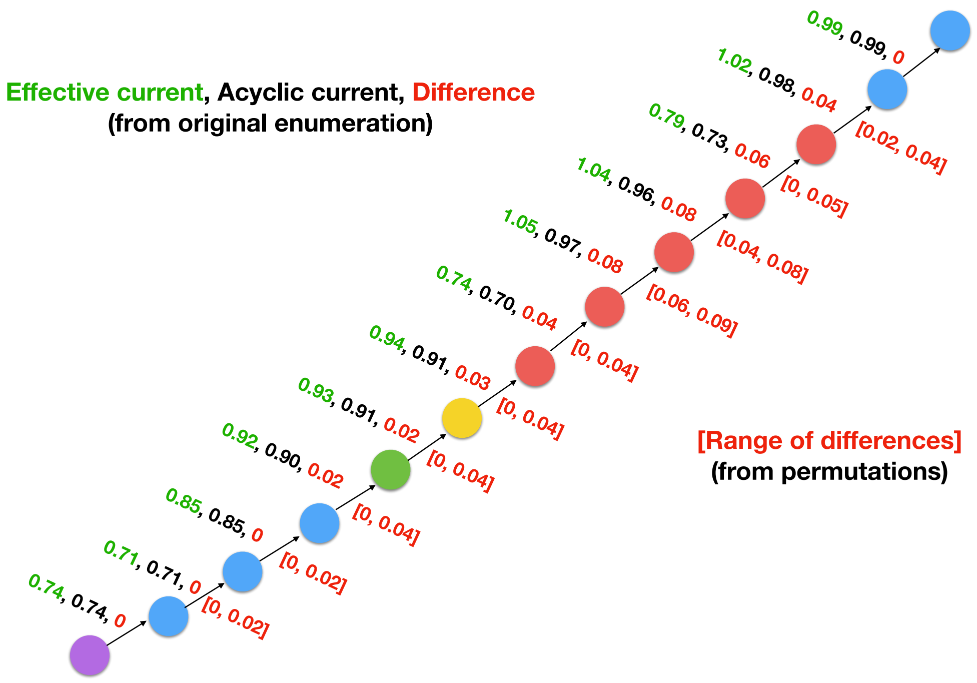


Figure 5.8: All pathways from START to G1* found after 20 iterations of removing cycles in different orders. The values on left each edge are the effective current (green), acyclic current (black), and the difference (red) found from the original enumeration. On the right of each edge is the range of differences between the effective current and the acyclic current from the 20 iterations (red).

deviation value being 0.0232. All pathways from START to G1* found after 10 iterations of randomly permuting the indexes of the states is shown in Figure 5.7. Each edge shows the range of acyclic current values that were found from the different permutations. While the order the cycles are removed changes the acyclic current, in any permutation the main biological pathway was still found. None of the acyclic currents found deviated significantly from the acyclic current found in the original enumeration (Fig. 5.4).

5.3.3 Identifying Redundant Edges

The algorithm developed for identifying redundant edges in the stochastic GRN is similar to Algorithm 5 for the deterministic model except for the TPT tools and the computation of the acyclic current that are used to analyze GRNs with missing edges instead of solely the DFS algorithm. We distort the GRN shown in Fig. 5.2 by removing one single edge and adjust the influence matrix A defined by Eq.(5.1). We apply the stochastic transition rules (Eqs.(5.4)-(5.5)) to obtain the transition matrix. We preprocess the stochastic matrix by zeroing out all entries that are less than 10^{-14} and compute the effective current. Finally, we use our cycle removal algorithm to obtain the acyclic current and explore the resulting graph $G(S, \{e^{+}\})$. We keep track of pathways from START to G1* that carry a significant amount of current (at least 20% of the normalized current). If such a pathway exists, we compare it to the one in the original dynamical network and mark changes in it if any. We then repeat this process for all 34 edges of the regulatory network in order to determine which edges can be removed without destroying the cell cycle.

```
Input: Cut-off for the transition probability in $P$
Initialization Start with the influence matrix $A$ for the GRN defined by
  (5.1). Each nonzero entry in $A$ corresponds to an edge in the GRN.
Output: List of pathways from START to G1* carrying a minimum of
  20% of the normalized current.
The main body
for each edge in the GRN do
    Step 1: Make the corresponding entry a 0 in the influence matrix $A$.
    Step 2: Update the stochastic transition rules defined by (5.4)-(5.7).
    Obtain matrix $P$.
    Step 3: Remove all entries in $P$ less than cut-off.
    Step 4: Compute effective current, $f^+_{i,j}$, defined by (2.11).
    Step 5: Run cycle removal algorithm (Algorithm 3) to obtain acyclic
    current.
    Step 6: Run DFS algorithm to find all pathways carrying at least 20%
    of the normalized acyclic current.
end
```

Algorithm 6: Edge Removal Algorithm for Stochastic GRN

As shown in Table 5.6, we have established that 26 of the 34 edges in the GRN can be removed from the network without destroying the cell cycle described by the stochastic GRN. This means the removal of these edges will have little to no effect on the biological pathway described in Table 5.2. 6 of the remaining 8 edges can be removed and a pathway from A to B can be achieved but does not follow the biological pathway (Table 5.7). The removal of only 2 of the edges, Cln3 → SBF and SBF → Cln1,2, results in no path found carrying at least 20% of the normalized acyclic current. Thus, even when stochasticity is added to the model, no meaningful pathway is produced in the removal of these edges. These results suggest that these edges are essential to the survival of the cell cycle.

Table 5.6: The edges of the gene regulatory network that can be removed without destroying the cell cycle modeled by the stochastic model [2]. All pathways carrying a minimum of 20% of the current was found after each removal in the GRN. Parameter values $\alpha = 5$ and $\beta = 6$ were used. The dominant pathway (pathway carrying most current) in each case was compared to the biological pathway described in Table 5.2. The effect on the main cycle corresponds to differences in protein state over time show in Table 5.2 and s refers to the cell state (i.e. $s_i = 1$ or 0 if protein i is active or inactive). "Row" corresponds to the row in Table 5.2 and we show how a row is modified according to the edge removal.

	Removed edge	**Effect on the main cycle**
1	Cln3 → Cln3	Row 2: $s = [11101010000]$ Row 3: $s = [11111010000]$ Row 4: $s = [11110000000]$ Row 5: $s = [11110100000]$ Row 6: $s = [11110101100]$ Between Row 6 and Row 7 $s = [11110101110]$ $s = [11110101111]$ $s = [01110101111]$ $s = [00010101111]$
2	Cln3 → MBF	Row 2: $s = [01001010000]$ Row 3: $s = [01011010000]$ Row 4: $s = [01010000000]$ Row 5: $s = [01010100000]$ Row 6: $s = [01010101100]$
3	Cln1,2 → Cln1,2	Between Row 7 and Row 8: $s = [00010001111]$
4	Sic1 → Clb1,2	Between Row 9 and Row 11 $s = [00001011111]$ $s = [00001010111]$
5	Clb5,6 → Sic1	No effect
6	Clb5,6 → Cdh1	No effect
7	Clb5,6 → Mcm1/SFF	Row 6: $s = [01110101000]$
8	Cdh1 → Clb1,2	No effect
9	Clb1,2 → SBF	Between Row 6 and Row 8 $s = [01010101110]$ $s = [01010001111]$ $s = [00010001111]$
10	Clb1,2 → MBF	Between Row 6 and Row 8

		$s = [00110101110]$ $s = [00100101111]$ $s = [00000101111]$
11	Clb1,2 → Sic1	No effect
12	Clb1,2 → Cdh1	Row 9: $s = [00001011111]$ Row 10: $s = [00001010111]$
13	Clb1,2 → Mcm1/SFF	Between Row 8 and Row 12 $s = [00001001011]$ $s = [00001000010]$
14	Clb1,2 → Cdc20/Cdc14	No effect
15	Clb1,2 → Swi5	Row 7: $s = [00010101111]$
16	Mcm1/SFF → Clb1,2	Skips Row 9
17	Mcm1/SFF → Mcm1/SFF	Between Row 7 and Row 8: $s = [00010001111]$
18	Mcm1/SFF → Cdc20/Cdc14	Skips Row 11
19	Mcm1/SFF → Swi5	Row 8: $s = [00000001110]$ Row 9: $s = [00001001110]$ Row 10: $s = [00001000110]$
20	Cdc20/Cdc14 → Sic1	Row 9: $s = [00000011111]$ Row 10: $s = [00000010111]$
21	Cdc20/Cdc14 → Clb5,6	Between Row 6 and Row 9 $s = [01000101110]$ $s = [00000101111]$ $s = [00001101111]$
22	Cdc20/Cdc14 → Cdh1	Between Row 7 and Row 8 $s = [00010001111]$ Between Row 10 and Row 13 $s = [00001000011]$ $s = [00001000001]$ $s = [00001000000]$
23	Cdc20/Cdc14 → Clb1,2	Between Row 9 and Row 11 $s = [00001011111]$ $s = [00001010111]$
24	Cdc20/Cdc14 → Cdc20/Cdc14	Between Row 7 and Row 8: $s = [00010001111]$
25	Cdc20/Cdc14 → Swi5	Row 8: $s = [00000001110]$ Row 9: $s = [00001001110]$ Row 10: $s = [00001000110]$ Skips Row 12
26	Swi5 → Swi5	Between Row 7 and Row 8: $s = [00010001111]$

Table 5.7: The edges of the gene regulatory network that when removed do not include all phases of the cell cycle when modeled by the stochastic model. All pathways carrying a minimum of 20% of the current was found after each removal in the GRN. Parameter values $\alpha = 5$ and $\beta = 6$ were used. The dominant pathway (pathway carrying most current) in each case was compared to the biological pathway described in Table 5.2. The effect on the main cycle corresponds to differences in protein state over time shown in Table 5.2 and s refers to the cell state (i.e. $s_i = 1$ or 0 if protein i is active or inactive). "Row" corresponds to the row in Table 5.2 and we show how a row is modified according to the edge removal.

	Removed edge	**Effect on the main cycle**
1	MBF → Clb5,6	Between Row 4 and Row 8: $s = [01110001000]$ $s = [00010001110]$
2	Cln1,2 → Sic1	Between Row 3 and Row 8: $s = [01111000000]$ $s = [01111100000]$ $s = [01110100100]$ $s = [01110101111]$ $s = [00010101111]$
3	Cln1,2 → Cdh1	Between Row 3 and Row 8: $s = [01110010000]$ $s = [01110110000]$ $s = [01110100100]$ $s = [01110101111]$ $s = [00010101111]$
4	Sic1 → Clb5,6	Between Row 2 and Row 8: $s = [01111110000]$ $s = [01110100100]$ $s = [01110101111]$ $s = [00010101111]$
5	Clb5,6 → Clb1,2	Between Row 5 and Row 8: $s = [01110100100]$ $s = [01110101111]$ $s = [00010101111]$
6	Swi5 → Sic1	Skips Row 2 - Row 12

5.4 Comparison of the Deterministic and Stochastic Models

Figure 5.9 shows a closer comparison of the results of the edge removal algorithms for the deterministic and stochastic models. Edges colored green/red correspond to ones with green/red arrows in the GRN shown in Fig. 5.2 respectively.

We see that 5 of the edges can be individually removed from the GRN without having any effect on the main cycle described by Table 5.2 when modeled by both the deterministic and stochastic model. Namely these edges are Clb5,6 → Sic1, Clb5,6 → Cdh1, Cdh1 → Clb1,2, Clb1,2 → Sic1, and Clb1,2 → Cdc20/Cdc14.

There are 11 edges that can be removed from the GRN when modeled by the deterministic model having little to no effect on the cell cycle and 3 edges that when removed do not go through all phases of the cell cycle as described in section 5.1.1. These same 11 edges can also be removed from the GRN when modeled by the stochastic model having little to no effect on the cell cycle and the same 3 edges can be removed when modeled by the stochastic model resulting in skipping the gap G2 phase. These 3 edges are again Cln1,2 → Cdh1, Sic1 → Clb5,6, and Clb5,6 → Clb1,2. In both models, when removing any of these 3 edges, the protein complex Cdc20/Cdc14 activates sooner than expected resulting in the G2 phase being skipped.

There are additional 15 edges that can be removed from the stochastic model with little to no effect on the cell cycle and additional 3 edges that can be removed but do not go through all the cell cycle phases. These additional 3 edges are MBF → Clb5,6, Cln1,2 → Sic1, and Swi5 → Sic1 (Rows 1,2 and 6 from Table 5.7). In the removal of MBF → Clb5,6, the protein Clb5,6 never activates causing the cell to never reach synthesis or the G2 phase. Synthesis being a crucial step in the cell cycle, the removal of this edge completely destroys the process. The removal of Cln1,2 → Sic1 results in the G2 phase being skipped due to the premature activation of Cdc20/Cdc14. Finally, the removal of Swi5 → Sic1 results in a jump directly from the excited G1 state to the stationary G1* state. This removes all states in between including those involved in synthesis and mitosis, both being crucial phases in the cell cycle. Thus, while we still find a path from START to G1*, the removal of this edge also destroys the cell cycle.

Our analysis shows that the stochastic model is much more robust than the deterministic one since it allows for the individual removal of more than double the number of edges than the deterministic model allows. We use the term robust in the sense that when applying Algorithm 3 to the model, the biological pathway can be extracted under a wide range of edges being individually removed from the GRN.

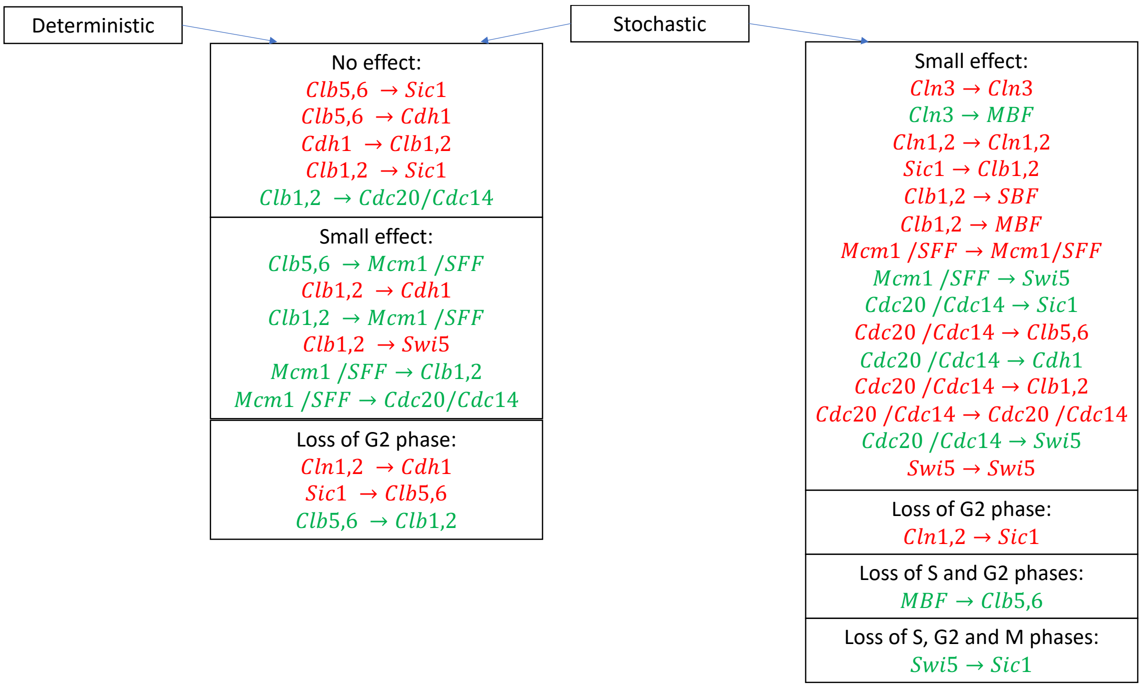


Figure 5.9: Comparison of edge removal algorithms for deterministic and stochastic models. The edges are categorized by their removal having either no effect on the cell cycle, a small effect on the cell cycle or the loss of some particular phases of the cell cycle. Edges are colored according to being an activating or inhibiting edge as described in Section 5.1.2.

Chapter 6: Conclusion

6.1 Summary of Methodological Advances

We have developed a set of analytical and computational tools based on transition path theory (TPT) to analyze flows in time-irreversible Markov chains. Our developments are equally applicable to both discrete-time and continuous-time Markov chains. Since the concept of reactive current has not been defined for time-irreversible networks, we proposed an extension of this. We propose a general recipe for designing modified Markov chains with desired stationary current and desired invariant distribution. With this, we are able to apply our recipe to time-irreversible networks so that the stationary currents in the modified Markov chains are equal to the effective currents and acyclic currents in the original ones. We call the currents generated in this manner the acyclic currents which can be used as a counterpart of the reactive current for time-irreversible Markov chains. To generate these acyclic currents in practice, we developed a cycle removal algorithm (Algorithm 3). As a result, we are able to apply our tools based on TPT to quantify the transitions from one user defined subset of states to another.

As a visual example, we applied Algorithm 3 to the Maier-Stein SDE. With this example, we were able to reduce the computational efforts of our cycle removal algorithm (Algorithm 3) by applying the algorithm to strongly connected components separately. This example also shows that in time-irreversible Markov chains, the forward (or backward) committor values does not have to strictly increase (or decrease) along no-detour reactive trajectories which is contrary to the time-reversible case.

6.2 Summary of Application to GRN

We investigate the dynamics of the GRN using two algorithms: (1) depth-first search algorithm used for the deterministic model and (2) TPT and the cycle removal algorithm used for the stochastic model. TPT and the cycle removal algorithm allow us to find acyclic currents where only the most likely transitions that occur between cell states are found. We then use these algorithms to test the robustness of the GRN by identifying redundant edges in the models. In order to identify redundant edges in models we apply the following steps. We first distort the GRN by removing one edge and adjusting the influence matrix. We then apply the appropriate transition rules for each model to obtain the dynamical networks. We explore these dynamical networks using algorithms (1) and (2) described above. Finally, we check if there is a pathway corresponding to the cell cycle going from the excited growth phase to the stationary growth phase. If such a pathway exists, we compare it to the one in the original dynamical network and mark changes in it if any. We repeat this process for all edges of the regulatory network in order to determine which edges can be removed without a major effect on the cell cycle. Our methodology can be used as a deterministic computational tool for analysis of the dynamical network.

We conclude the removal of 11 out of 34 edges make no significant effect on the cell cycle modeled by the deterministic GRN. Thus, if a mutation occurs in protein A that does not allow for protein B to be turned on, the cell cycle will still proceed as expected. In some cases, there may only be minor modifications to the size of the basin of attraction or main cycle pathway. There are 3 edges that when removed only result in the loss of the gap phase G2. The remaining 20 edges are very essential to the gene regulatory network. Therefore, removing any of the remaining 20 edges causes the cell cycle to be destroyed and cell states converge to states not on the biological pathway. We also conclude that 26 out of the possible 34 edges can be removed without notable effects on the cell cycle modeled by the stochastic model. There are 4 edges that result in the loss of the gap phase, 2 edges that result in the loss of synthesis and/or mitosis, and removing either of the remaining 2 edges produce no pathway. Overall, this shows that the stochastic model is notably more robust than the deterministic one.

The GRN for the budding yeast cell cycle is a robust network. From a biological point of view, it is important for us to perform mutation analysis on GRNs since it would be beneficial to be able to specifically determine which mutations allow for the survival of the biological process of interest and which mutations aid in their destruction. Future research would include testing the robustness and investigating the dynamics of more complex GRNs.

6.3 Future work

One question that arises is, can this methodology be used if we are unable to store the entire dynamical network for a GRN into computer memory? With an N-node GRN, the dynamical network consist of 2^N nodes. Hence, given a GRN with say 50 nodes, we are unable to store the entire network with $2^{50} \approx 10^{15}$ nodes and apply our methodology. However, even when the network is large and complex, we would like our tools to still be efficient in analyzing these networks. Through our analysis, we have observed that even though the dynamical network for some GRNs become huge, the vast majority of states are biologically meaningless. Thus, if we can incorporate in our algorithm a way to extract meaningful states, say through Monte Carlo sampling, then we can apply our methodology to larger and more complex networks. We would like to explore this idea through more examples involving larger networks [9, 44].

Gene regulatory networks are becoming an increasingly-popular tool for the modeling and analysis of biological processes [45–47]. One goal is to use our methodology to answer questions arising in more complicated GRNs. One network of interest is the GRN model representing the segment polarity genes [48, 49]. These genes are a group of genes involved in embryonic pattern formation in the fruit fly Drosophila Melanogaster. Homologs of the segment polarity genes have been identified in vertebrates, including humans, which suggests strong evolutionary conservation. This GRN is more complex in that these genes refine and maintain their expression through the network of intra- and intercellular regulatory interactions and consists of 16 nodes. Moreover, some models of the GRN introduce a time-delay in the activation or inactivation of nodes [50]. It would be of interest to us to see how these time-delays effect the dynamic of the network and overall robustness.

Finally, our analytical and computational tools can also be applied to time-irreversible Markov chains arising in other applications. One application of interest comes from investigating the aggregation process of Lennard-Jones atoms. In 2017, Forman and Cameron [51] proposed expected initial and pre-attachment distributions to analyze the aggregation/deformation in the LJ_{6-14} network, where LJ_N is the N-atom Lennard-Jones cluster and the network representing its energy landscape (this was later extended to the LJ_{6-15} network). With this they answered the question: If the aggregation process starts at the bicapped tetrahedron local minimum of LJ_6, formed as a result of the attachment of an additional atom to the only minimum of LJ_5, what configurations are most likely to be observed in each LJ_N as the aggregation process proceeds to LJ_{14}. This LJ_{6-15} network is not only time-irreversible but also reducible. However, allowing for the attachment and detachment of particles gives rise to the irreducible network we desire. Thus, we would like to explore the use of our methodology to answer similar questions for this extended network.

Bibliography

[1] Fangting Li, Tao Long, Ying Lu, Qi Ouyang, and Chao Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences*, 101(14):4781–4786, 2004.

[2] Yuping Zhang, Minping Qian, Qi Ouyang, Minghua Deng, Fangting Li, and Chao Tang. Stochastic model of yeast cell-cycle network. *Physica D: Nonlinear Phenomena*, 219(1):35–39, 2006.

[3] Albert-László Barabási et al. *Network science*. Cambridge university press, 2016.

[4] Wataru Souma, Yoshi Fujiwara, and Hideaki Aoyama. Complex networks and economics. *Physica A: Statistical Mechanics and its Applications*, 324(1-2):396–401, 2003.

[5] David Wales et al. *Energy landscapes: Applications to clusters, biomolecules and glasses*. Cambridge University Press, 2003.

[6] Stanley Wasserman, Katherine Faust, et al. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[7] Peter J Carrington, John Scott, and Stanley Wasserman. *Models and methods in social network analysis*, volume 28. Cambridge university press, 2005.

[8] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[9] Ilya Shmulevich, Ilya Gluhovsky, Ronaldo F Hashimoto, Edward R Dougherty, and Wei Zhang. Steady-state analysis of genetic regulatory networks modelled by probabilistic boolean networks. *Comparative and functional genomics*, 4(6):601–608, 2003.

[10] Xiao Fan Wang and Guanrong Chen. Complex networks: small-world, scale-free and beyond. *IEEE circuits and systems magazine*, 3(1):6–20, 2003.

[11] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[12] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao-Bin Hu. Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1706–1712, 2009.

[13] Anton Bovier, Michael Eckhoff, Véronique Gayrard, and Markus Klein. Metastability and low lying spectra in reversible markov chains. *Communications in mathematical physics*, 228(2):219–255, 2002.

[14] Anton Bovier, Michael Eckhoff, Véronique Gayrard, and Markus Klein. Metastability in reversible diffusion processes i: Sharp asymptotics for capacities and exit times. *Journal of the European Mathematical Society*, 6(4):399–424, 2004.

[15] Weinan E. and Eric Vanden-Eijnden. Towards a theory of transition paths. *Journal of statistical physics*, 123(3):503, 2006.

[16] Philipp Metzner, Christof Schütte, and Eric Vanden-Eijnden. Transition path theory for Markov jump processes. *Multiscale Modeling & Simulation*, 7(3):1192–1219, 2009.

[17] Maria Cameron and Eric Vanden-Eijnden. Flows in complex networks: theory, algorithms, and application to Lennard-Jones cluster rearrangement. *Journal of Statistical Physics*, 156(3):427–454, 2014.

[18] Michael Manhart and Alexandre V Morozov. Statistical physics of evolutionary trajectories on fitness landscapes. In *First-Passage Phenomena and Their Applications*, pages 416–446. World Scientific, 2014.

[19] Michael Manhart and Alexandre V Morozov. Path-based approach to random walks on networks characterizes how proteins evolve new functions. *Physical review letters*, 111(8):088102, 2013.

[20] Eric H Davidson and Douglas H Erwin. Gene regulatory networks and the evolution of animal body plans. *Science*, 311(5762):796–800, 2006.

[21] Andre Ribeiro, Rui Zhu, and Stuart A Kauffman. A general modeling strategy for gene regulatory networks with stochastic dynamics. *Journal of Computational Biology*, 13(9):1630–1639, 2006.

[22] Selene L Fernandez-Valverde, Felipe Aguilera, and René Alexander Ramos-Díaz. Inference of developmental gene regulatory networks beyond classical model systems: new approaches in the post-genomic era. *Integrative and comparative biology*, 58(4):640–653, 2018.

[23] Frank Emmert-Streib, Matthias Dehmer, and Benjamin Haibe-Kains. Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Frontiers in cell and developmental biology*, 2:38, 2014.

[24] Debashis Barik, David A Ball, Jean Peccoud, and John J Tyson. A stochastic model of the yeast cell cycle reveals roles for feedback regulation in limiting cellular variability. *PLoS computational biology*, 12(12), 2016.

[25] Katherine C Chen, Attila Csikasz-Nagy, Bela Gyorffy, John Val, Bela Novak, and John J Tyson. Kinetic analysis of a molecular model of the budding yeast cell cycle. *Molecular biology of the cell*, 11(1):369–391, 2000.

[26] Likun Zheng, Meng Chen, and Qing Nie. External noise control in inherently stochastic biological systems. *Journal of mathematical physics*, 53(11):115616, 2012.

[27] Michael B Elowitz, Arnold J Levine, Eric D Siggia, and Peter S Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186, 2002.

[28] Peter Hänggi. Stochastic resonance in biology how noise can enhance detection of weak signals and help improve biological information processing. *ChemPhysChem*, 3(3):285–290, 2002.

[29] Jonathan M Raser and Erin K O'Shea. Noise in gene expression: origins, consequences, and control. *Science*, 309(5743):2010–2013, 2005.

[30] Henry Eyring. The activated complex in chemical reactions. *The Journal of Chemical Physics*, 3(2):107–115, 1935.

[31] James R Norris. *Markov chains*. Number 2. Cambridge university press, 1998.

[32] Donald G Truhlar, Bruce C Garrett, and Stephen J Klippenstein. Current status of transition-state theory. *The Journal of physical chemistry*, 100(31):12771–12800, 1996.

[33] Christoph Dellago, Peter Bolhuis, and Phillip L Geissler. Transition path sampling. *Advances in chemical physics*, 123:1–78, 2002.

[34] Jianfeng Lu and James Nolen. Reactive trajectories and the transition path process. *Probability Theory and Related Fields*, 161(1-2):195–244, 2015.

[35] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[36] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[37] Dimitri P Bertsekas. *Network optimization: continuous and discrete models.* Athena Scientific Belmont, 1998.

[38] Robert S Maier and Daniel L Stein. A scaling theory of bifurcations in the symmetric weak-noise escape problem. *Journal of statistical physics*, 83(3-4):291–357, 1996.

[39] Frederick R Cross, Vincent Archambault, Mary Miller, and Martha Klovstad. Testing a mathematical model of the yeast cell cycle. *Molecular biology of the cell*, 13(1):52–70, 2002.

[40] Katherine C Chen, Laurence Calzone, Attila Csikasz-Nagy, Frederick R Cross, Bela Novak, and John J Tyson. Integrative analysis of cell cycle control in budding yeast. *Molecular biology of the cell*, 15(8):3841–3862, 2004.

[41] John J Tyson, Bela Novak, Kathy Chen, and John Val. Checkpoints in the cell cycle from a modeler's perspective. In *Progress in cell cycle research*, pages 1–8. Springer, 1995.

[42] Michael D Mendenhall and Amy E Hodge. Regulation of cdc28 cyclin-dependent protein kinase activity during the cell cycle of the yeast saccharomyces cerevisiae. *Microbiol. Mol. Biol. Rev.*, 62(4):1191–1243, 1998.

[43] Alan U Sabino, Miguel FS Vasconcelos, Misaki Yamada Sittoni, Willian W Lautenschlager, Alexandre S Queiroga, Mauro CC Morais, and Alexandre F Ramos. Lessons and perspectives for applications of stochastic models in biological and cancer research. *Clinics*, 73, 2018.

[44] Shane Squires, Andrew Pomerance, Michelle Girvan, and Edward Ott. Stability of boolean networks: The joint effects of topology and update rules. *Physical Review E*, 90(2):022814, 2014.

[45] Fernando M Delgado and Francisco Gómez-Vela. Computational methods for gene regulatory networks reconstruction and analysis: A review. *Artificial intelligence in medicine*, 95:133–145, 2019.

[46] Gilles Bernot, Jean-Paul Comet, Adrien Richard, Madalena Chaves, Jean-Luc Gouzé, and Frédéric Dayan. Modeling and analysis of gene regulatory networks. In *Modeling in Computational Biology and Biomedicine*, pages 47–80. Springer, 2013.

[47] Ezio Bartocci and Pietro Lió. Computational modeling, formal analysis, and tools for systems biology. *PLoS computational biology*, 12(1), 2016.

[48] Réka Albert and Hans G Othmer. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster. *Journal of theoretical biology*, 223(1):1–18, 2003.

[49] George Von Dassow, Eli Meir, Edwin M Munro, and Garrett M Odell. The segment polarity network is a robust developmental module. *Nature*, 406(6792):188–192, 2000.

[50] Madalena Chaves, Reka Albert, and Eduardo D Sontag. Robustness and fragility of boolean models for genetic regulatory networks. *Journal of theoretical biology*, 235(3):431–449, 2005.

[51] Yakir Forman and Maria Cameron. Modeling aggregation processes of lennard-jones particles via stochastic networks. *Journal of Statistical Physics*, 168(2):408–433, 2017.