COMPUTATIONS OF DSGRN

by

LUN ZHANG

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Mathematics

Written under the direction of

Konstantin Mischaikow

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2020

ABSTRACT OF THE DISSERTATION

Computations of DSGRN

By LUN ZHANG

Dissertation Director:

Konstantin Mischaikow

The study of this thesis is mainly motivated by the computational problems that arised from paper [7] where authors designed a new dynamic system model called Dynamic Signatures Generated by Regulatory Networks (DSGRN) to simulate gene regulatory networks. An essential property of DSGRN model is that the phase transition graph is invariant over a family of subvariety which is defined by a set of simple polynomials over parameter space. One of the work in DSGRN is to study the family of subvariety over parameter space and two basic problem arises: realizability and topology. As to reailizability, we are concerned with constructing the whole family of subvariety. For topology, we want to compute the homology group of a set of subvariety for a given phenotype. This thesis has two main contributions. First, we develop a new algorithm that greatly extends the computational ability of DSGRN to wider class of regulatory network. Second, we design a computational framework for the homology computation of subvariety over the parameter space.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor, Professor Konstantin Mischaikow, for his support and guidance during my Ph.D. study. He inspired me with many interesting problems, advised me on research and academic writings. His insights, knowledge, patience, and motivation helped me at every step. He also provided me with great opportunities to get in touch with the math community. I could not have imagined having a better advisor for my Ph.D. study. I would like to thank Shane Kepley and Marcio Gameiro, for giving valuable suggestions on my work. I would like to thank my thesis committee members, Professors Micheal Saks, Fioralba Cakoni, and Tomas Gedeon for their support and encouragement on the writing of my dissertation.

Many thanks to my fellow math students: Xin Fu, Jin Gong, Carlos Ronchi, Kelly Spendlove, Hongjie Yu, Yuxuan Yang, Yukun Yao, Songhao Zhu, . We discussed many math problems and ideas together.

Last but not the least, I would like to thank my parents for their unconditional love and support

Dedication

To my parents.

Table of Contents

Chapter 1

Introduction

This introduction provides a brief description of how the topics of this thesis arises in the context of mathematical modeling of problems from systems biology and the outline of the thesis.

Biologists often describe regulatory networks in terms of annotated directed graphs, such as that shown in Figure 1.1 where the labeling of the edges, $n \to m$ or $n \dashv m$ indicates whether node n activates or represses node m. Our goal is to describe the type of dynamics that can be expressed by the regulatory network.

We first have a look of two examples of regulatory network, see Figure 1.2, and how to use them to analyze the gene dynamics qualitatively. The two networks in Figure 1.2 are the self-oscillatory and the toggle switch network.

The toggle switch model was introduced in [30] to study the bistability in e-coli, and the self-oscillatory is a subnetwork in the p53 regulatory network [2]. They are both directed graphs, but there is an important difference between regulatory network and directed graph. For a regulatory network there are two different types of edges, activation $\to$ and repression $\dashv$ which assume biological meaning.

For example, as to $1 \to 2$, it means a large amount of protein produced by gene 1 will activate the expression of gene 2. On the other hand $2 \dashv 1$ means a large amount of protein produced by gene 2 will repress the expression of gene 1. As to the toggle switch we have $1 \dashv 2$ and $2 \dashv 1$, which means a large amount of protein produced by each of the two genes will repress the expression of other gene. Even a simple naive analysis of these two regulatory networks suggests that they exhibit very different dynamics.

For the self-oscillatory network, a large amount of protein produced by gene 1 will activate the expression of gene 2. When the amount of the protein produced by gene

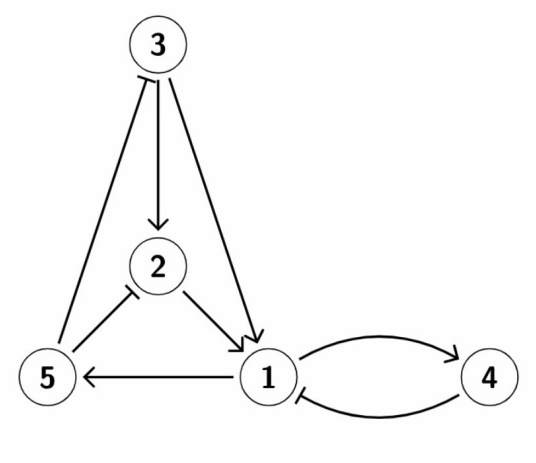
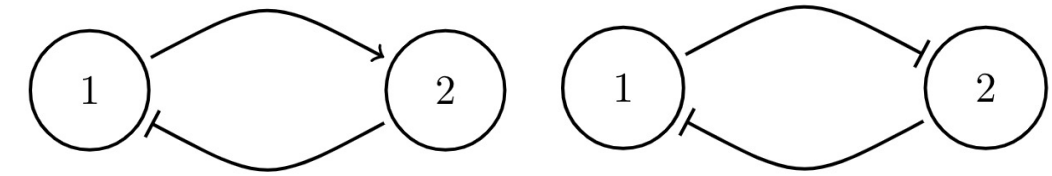


Figure 1.1: Example of a regulatory network.



(a) (b)

Figure 1.2: (a) Self-oscillatory regulatory network (b) Toggle switch network

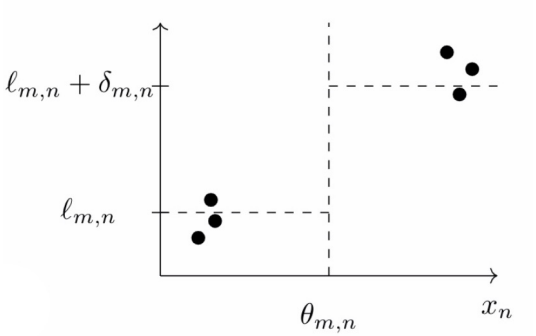


Figure 1.3: Model for edge $n \to m$ where $\ell_{m,n}$ indicates low level of growth rate of x_m induced by x_n and $\ell_{m,n} + \delta_{m,n}$ indicates high level of growth rate of x_m induced by x_n. $\theta_{m,n}$ provides information about the value of x_n that lies between low and high values inducing low and high expression levels.

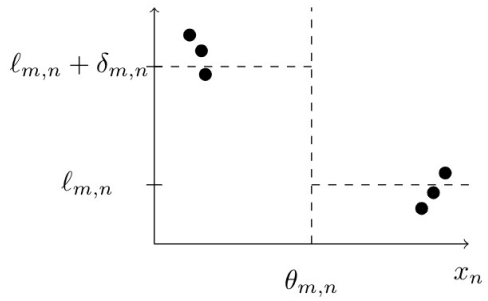


Figure 1.4: Model for edge $n \dashv m$ where $\ell_{m,n}$ indicates low level of growth rate of x_m induced by x_n and $\ell_{m,n} + \delta_{m,n}$ indicates high level of growth rate of x_m induced by x_n. $\theta_{m,n}$ provides information about the value of x_n that lies between low and high values inducing low and high expression levels.

2 increases to some level, it starts repressing the expression of gene 1. Consequently, when the amount of protein produced by gene 1 is reduced to lower level, the expression of gene 2 will also be slowed down which will finally cause the amount of protein produced by gene 1 back to high level. From above qualitative analysis, the self-oscillatory regulatory network encodes cyclic dynamics. For toggle switch, if the initial amount of protein produced by gene 1 is large, it will repress the expression of gene 2. Then when the amount of protein produced by gene 2 is low, it does not have repression effect over the expression of gene 1, in the end we will end at a state such that the amount of protein produced by gene 1 is large and protein produced by gene 2 is small. In another word, toggle switch will encode fixed point dynamics.

The above discussion has shown the power of a regulatory network to analyze the dynamics of gene interactions qualitatively. In order to move further researchers need to develop quantitative model over regulatory network.

A natural choice is to use ordinary differential equations, i.e. we associate a dynamical system to the regulatory network

$$\dot{x} = F(x, \lambda)$$

where λ represents some biological parameters, such as decay rates, expression levels, or thresholds. However, the biggest trouble is we do not know what F is. Also the dimension of parameter λ can be high for more complicated models like p53 network

model Figure 1.1 . At this point it is too complicated to understand the dynamics by looking at the picture.

Achieving the goal to describe the dynamics for a regulatory network requires that we impose a mathematical interpretation on the regulatory network that is compatible with its use as a biological model. With this in mind, we assign to node n a state variable, $x_n > 0$, that corresponds with the quantity of a protein, mRNA, or a signaling molecule. Precise nonlinear expressions for the interactions of the variables are *not* assumed to be known, but we do assume that the sign of the rate of change of x_m is determined by

$$-\gamma_m x_m + \Lambda_m(x) \tag{1.1}$$

where γ_m indicates the decay rate and Λ_m is a parameter dependent function that characterizes the rate of growth of x_m. Note that Λ_m is a function of x_n if and only if there exists an edge from n to m in the regulatory network.

Since the biological model provides minimal information about the effect of x_n on x_m we want to choose a mathematical expression with a minimal set of assumptions. The rates of growth of x_m due to x_n are labeled as $0 < \ell_{m,n} < \ell_{m,n} + \delta_{m,n}$.

Figure 1.3 corresponds to an edge $n \to m$ and $\theta_{m,n}$ indicates that the rate of increase $\ell_{m,n}$ must occur at some lesser value of x_n and the rate of increase $\ell_{m,n} + \delta_{m,n}$ must occur at some greater value of x_n. An arrow of the form $n \dashv m$ leads to the opposite relation 1.4.

This introduces three positive parameters, $\ell_{m,n}$, $\delta_{m,n}$, and $\theta_{m,n}$, for each edge in the regulatory network. Note that this is the minimal number of parameters that allows one to quantify the assumption that x_n activates x_m (or equivalently that x_n represses x_m). We encode this information with the following functions

$$\lambda^+_{m,n}(x_n) = \begin{cases} \ell_{m,n} & \text{if } x_n < \theta_{m,n} \\ \ell_{m,n} + \delta_{m,n} & \text{if } x_n > \theta_{m,n} \end{cases} \quad \text{and} \quad \lambda^-_{m,n}(x_n) = \begin{cases} \ell_{m,n} + \delta_{m,n} & \text{if } x_n < \theta_{m,n} \\ \ell_{m,n} & \text{if } x_n > \theta_{m,n}. \end{cases}$$

We do *not* assume that the values of $\ell_{m,n}, \delta_{m,n}$, or $\theta_{m,n}$ are known. This is intentional

as many of these parameters do not have an easy biological interpretation and/or correspond to physical constants which are difficult or impossible to accurately measure. Thus, the goal is not to determine the dynamics at any choice of parameters, but to determine the range and robustness of the qualitative dynamics exhibited by a network.

A regulatory network such as that of Figure 1.1 does not indicate how multiple inputs to a particular node should be processed. An approach that is used is to assume a simple algebraic relationship involving sums and products of the $\lambda^{\pm}$. As an example, a reasonable choice for x_1 of Figure 1.1 is

$$\Lambda_1(x_2, x_3, x_4) = \left(\lambda^+(x_2) + \lambda^+(x_3)\right) \lambda^-(x_4). \tag{1.2}$$

Observe that each $\lambda^{\pm}$ takes only two values and therefore, generically Equation (1.2) takes 8 distinct values which are

$$\begin{aligned}
p_0 &= (\ell_1 + \ell_2)\ell_3 & p_4 &= (\ell_1 + \ell_2 + \delta_1)\ell_3 \\
p_1 &= (\ell_1 + \ell_2)(\ell_3 + \delta_3) & p_5 &= (\ell_1 + \ell_2 + \delta_1)(\ell_3 + \delta_3) \\
p_2 &= (\ell_1 + \ell_2 + \delta_2)\ell_3 & p_6 &= (\ell_1 + \ell_2 + \delta_1 + \delta_2)\ell_3 \\
p_3 &= (\ell_1 + \ell_2 + \delta_2)(\ell_3 + \delta_3) & p_7 &= (\ell_1 + \ell_2 + \delta_1 + \delta_2)(\ell_3 + \delta_3).
\end{aligned}$$

As is suggested in the caption of Figures 1.3, 1.4 we do not interpret the values of $\lambda^{\pm}$, or Λ as literal expressions of the nonlinear interactions, but rather, that the associated parameter values are landmarks of whatever of the "true" nonlinear function is. This has several consequences.

1. We cannot expect Equation (1.1) to provide precise information about the growth rate of x_m. Therefore we restrict our attention to asking whether the level of x_m is increasing or decreasing. However, we wish to answer this question over all the possible parameter values γ, θ, ℓ and δ.

2. The only values of x_m at which the dynamics of x_n change are of the form $\theta_{m,*}$. The associated hyperplanes $x_j = \theta_{k,j}$ decompose phase space $[0, \infty)^N$, where N is the number of nodes in the network, into N-dimensional rectangular regions

called *domains*.

3. We can determine all possible signs of (1.1) associated with (1.2) by studying all possible total orders that can be realized by $\{p_0, \ldots, p_7\}$. To be more precise, define $\mathcal{P} = \{p_0, \ldots, p_7\}$.

   For a given $\sigma \in S_8$ which is symmetric group of order 8, define its realizable set

$$\Xi_\sigma := \left\{\xi \in \mathsf{R}^6_+ : p_{\sigma(k)}(\xi) < p_{\sigma(k+1)}(\xi) \text{ for all } 0 \leq k \leq 6\right\}. \tag{1.3}$$

   and also define a subset of S_8 such that

$$\mathcal{T}(\mathcal{P}, (0,\infty)^6) := \{\sigma \in S_8 : \Xi_\sigma \text{ is nonempty}\}.$$

   Then if we can determine $\mathcal{T}\left(\{p_0, \ldots, p_7\}, (0,\infty)^6\right)$ for (1.2) we can determine all possible signs of (1.1) associated with (1.2) by cataloguing the relative values of $\gamma_1\theta_{j,1},\ j = 4, 5$ with respect to $\{p_0, \ldots, p_7\}$. For example, if we can find a ξ such that $p_0(\xi) < p_1(\xi) < \cdots < p_7(\xi), j = 0, \ldots, 6$, then we can make the $-\gamma_1\theta_{j,1} + p_i(\xi) < 0, -\gamma_1\theta_{j,1} + p_{i+1}(\xi) > 0$ by choosing appropriate values of $\gamma_1, \theta_{j,1}$. For more details, we refer to chapter 2.

We have developed software, Dynamic Signatures Generated by Regulatory Networks (DSGRN), that given the information of the form provided by consequence 3 is capable of efficiently building a database of the global dynamics of a regulatory network over all of parameter space [7]. In [7] we are restricted to considering networks with nodes that have at most three in edges and at most three out edges. In the thesis, we can extend the networks with nodes that have no more than 4 input edges and 5 output edges and with some restriction we can allow 5-6 inputs edges.

1.1 Outline of thesis

In chapter 2, we introduce the algebraic constraints linear extension problem (AC-LEP) which is coming from the study of sign for equations (1.1). In order to solve AC-LEP we first focus on its linear case which we call linear constraints linear extension

problem (LC-LEP) and develop an efficient algorithm to solve LC-LEP. Then we use a linearization method to transform AC-LEP to LC-LEP and solve the AC-LEP by using the LC-LEP algorithm under some restrictions. At the end, we combine the linearization method with cylindrical algebraic decomposition [4] and sampling method to solve the general AC-LEP. The result in this chapter allowed us to greatly expand the capabilities of DSGRN [7]. In particular, it can now handle the algebraic combinations of 4 to 6 input edges as is indicated in Section 2.4.

In chapter 3, we introduce the parameter space homology computation problem in DSGRN model. We first define the sets which are of interest for their homology. Those sets do not have good topological structure, i.e. CW complex structure, so we find a series of CW complex to approximate those sets in homology sense. We first show the approximation is good enough, i.e. the singular homology of those sets are isomorphic to the cellular homoloy of the approximating CW complex if some simple numerical conditions are satisfied. Then we provide the an efficient computation framework to construct those CW complex and compute their homology. At last, we show the applications to regulatory networks.

1.2 An example

We use a simple example to illustrate all the points introduced previously. Consider the toggle switch dynamics, Figure 1.2 (b), then the sign of the rate of change of x_1 and x_2 are determined by the sign of

$$-\gamma_1 x_1 + \Lambda_1(x_2) \tag{1.4}$$

$$-\gamma_2 x_2 + \Lambda_2(x_1) \tag{1.5}$$

and a reasonable choice for $\Lambda_1(x_1)$ and $\Lambda_2(x_2)$ is

$$\Lambda_1(x_2) = \lambda^-_{1,2}(x_2), \Lambda_2(x_1) = \lambda^-_{2,1}(x_1)$$

From the previous discussion, in order to study the sign of rate of change around

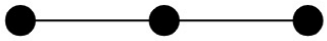


Figure 1.5: Factor graph for node 1 parameter space in toggle switch. The left node represents parameter region $\{\gamma_1\theta_{2,1} < l_{1,2}\}$, the middle node represents parameter region $\{l_{1,2} < \gamma_1\theta_{2,1} < l_{1,2} + \delta_{1,2}\}$, the right node represents parameter region $\{l_{1,2} + \delta_{1,2} < \gamma_1\theta_{2,1}\}$. The edges in between means in order to change from one parameter region to another one, we only need to change one inequality.

the switching thresholds $\theta_{1,2}, \theta_{2,1}$, we only need to study the all the total order that can be realized by $\mathcal{P}_1 = \{p_0 = l_{1,2}, p_1 = l_{1,2} + \delta_{1,2}\}$, $\mathcal{P}_2 = \{p_0 = l_{2,1}, p_1 = l_{2,1} + \delta_{2,1}\}$ for node 1,2 respectively. As $\mathcal{P}_1$ and $\mathcal{P}_2$ are similar, we only need to study one of them. We choose to compute set

$$\mathcal{T}(\mathcal{P}_1, (0,\infty)^2)$$

It is easy to see $\mathcal{T}(\mathcal{P}_1, (0,\infty)^2) = \{(0,1)\}$ as $l_{1,2} < l_{1,2} + \delta_{1,2}$ when $\delta_{1,2} > 0$. Even though $\mathcal{T}(\mathcal{P}, (0,\infty)^2)$ is easy to compute in our case, in general or for more complicated Λ, this problem is difficult and it is the main topic of chapter 2. Once we have determined $\mathcal{T}(\mathcal{P}_1, (0,\infty)^2) = \{(0,1)\}$, we can determine all possible signs of (1.4) associated with $\mathcal{P}_1$ by cataloguing the relative values of $\gamma_1\theta_{2,1}$ with respect to $\{p_0, p_1\}$. The parameter regions corresponding to different signs of (1.4) are $\{\{\gamma_1\theta_{2,1} < l_{1,2}\}, \{l_{1,2} < \gamma_1\theta_{2,1} < l_{1,2} + \delta_{1,2}\}, \{l_{1,2} + \delta_{1,2} < \gamma_1\theta_{2,1}\}\}$. Symmetrically for node 2 we have $\{\{\gamma_2\theta_{1,2} < l_{2,1}\}, \{l_{2,1} < \gamma_2\theta_{1,2} < l_{2,1} + \delta_{2,1}\}, \{l_{2,1} + \delta_{2,1} < \gamma_2\theta_{1,2}\}\}$. Thus, the three parameter regions for each node can be visualized like Figure 1.5 which we call factor graph.

This implies for each factor graph $PG(i)$, $i = 1, 2$, the associated parameter space $(\gamma, \theta, \ell, \delta) \in (0,\infty)^4$ is subdivided into the regions where the explicit constraints are as follows:

$$PG(1) = \{\{\gamma_1\theta_{2,1} < l_{1,2}\}, \{l_{1,2} < \gamma_1\theta_{2,1} < l_{1,2} + \delta_{1,2}\}, \{l_{1,2} + \delta_{1,2} < \gamma_1\theta_{2,1}\}\}$$
$$PG(2) = \{\{\gamma_2\theta_{1,2} < l_{2,1}\}, \{l_{2,1} < \gamma_2\theta_{1,2} < l_{2,1} + \delta_{2,1}\}, \{l_{2,1} + \delta_{2,1} < \gamma_2\theta_{1,2}\}\}.$$

The parameter graph for the toggle switch is the product of the factor graphs, i.e.

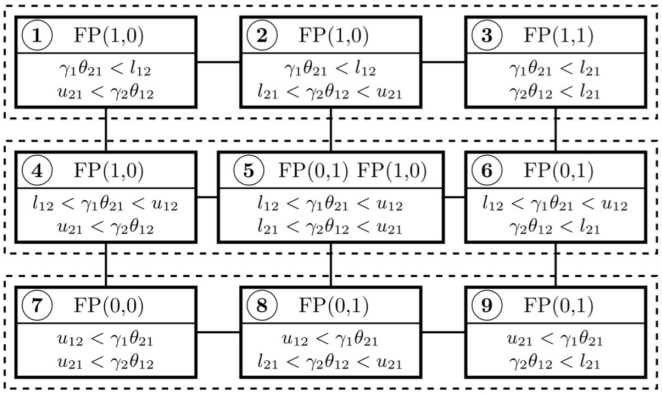


Figure 1.6: Parameter graph for toggle switch parameter space. The upper row for each node shows the relative position of fixed point with respect to thresholds $\theta_{1,2}, \theta_{2,1}$ and the second row shows the defining inequalities of the parameter regions.

.

$PG = PG(1) \times PG(2)$, and the subdivision of the full parameter space $(0,\infty)^8$ is given by product of the subdivisions associate with each factor graph. Figure 1.6 indicates PG along with the associated constraints.

By calling DSGRN we can get the graph Figure 1.6 which shows the correspondence between parameter regions and regulatory network dynamics. Observe that a subset of nodes S of the parameter graph corresponds to a subset of parameter space $(0,\infty)^8$. In applications the choice of S is determined by dynamics. For example, as indicated in Figure 1.6 in the toggle switch the dynamics associated with the central node is that of bistability while the dynamics of the eight surrounding nodes is that of monostability. Two natural questions are:

1. What is the topology of parameter space that exhibits bistability?

2. What is the topology of parameter space that exhibits monostability?

We claim that regions associated to single nodes in the parameter graph are contractible and that the homology for the second region is the same as that of a circle.

To proceed further some notation is helpful. Given a node s of a parameter graph PG, let $\widetilde{\Upsilon}(\{s\})$ denote the associated subset of the parameter space $(0,\infty)^D$ and a node for parameter graph $PG(i)$. Given a node s_i of a factor graph $PG(i)$, let $\widetilde{\Upsilon}_i(s_i)$ denote the associated subset of the parameter space $(0,\infty)^{D_i}$. A *naive* statement of the goal of

this section is as follows. Let S be a subset of nodes of a parameter graph PG, compute the homology of

$$\widetilde{\Upsilon}(S) = \left(\bigcup_{s \in S} \mathrm{cl}(\widetilde{\Upsilon}(\{s\})) \right) \cap (0, \infty)^D.$$

where s denote the parameter node and

There is a fundamental problems with $\widetilde{\Upsilon}(S)$ as defined above. To compute $\mathrm{Hom}(\widetilde{\Upsilon}(S))$ we require that $\widetilde{\Upsilon}(S)$ be a finite cell-complex that can be realized as a CW complex This is clearly not the case as $\widetilde{\Upsilon}(S)$ is neither compact nor bounded. This problem is not easily rectified since $\widetilde{\Upsilon}(S)$ is meant to represent biologically relevant parameters. In particular, since the edges of the regulatory network are meant to indicate activation or repression it is essential that $\delta > 0$ and $\theta > 0$. To resolve this problem for each node s in a factor graph we define CW complexes $\widetilde{\mathcal{X}}_\epsilon(\{s\})$ which are a good approximation to $\widetilde{\Upsilon}(\{s\})$.

To construct the CW complex, observing the factor graph $PG(i)$, we first find the parameter space has a natural direct product structure such that $\{\gamma_1, \theta_{2,1}, l_{1,2}, \delta_{1,2}\} \times \{\gamma_2, \theta_{1,2}, l_{2,1}, \delta_{2,1}\}$ while the change rate of x_1 only depends on the parameters of the first component and x_2 only depends on the second. Then we can construct CW complex on each component and take their product as the CW complex over the whole parameter space. In toggle switch network, the two nodes are at symmetric position, so we only need to construct the CW complex for one of them and the other is the same.

We focus the parameter space for node 1, i.e. $\{\gamma_1, \theta_{2,1}, l_{1,2}, \delta_{1,2}\}$. We claim the bounded set $Z = \{\gamma_1 = 1, \theta_{2,1} + l_{1,2} + \delta_{1,2} = 1, \theta_{2,1} > 0, l_{1,2} > 0, \delta_{1,2} > 0\}$ has the property such that $\widetilde{\Upsilon}_1(S_1)$ is homotopic equivalent to $\widetilde{\Upsilon}_1(S_1) \cap Z$ for any S_1 in $PG(1)$. Then we consider closed and bounded set $\widetilde{\mathcal{X}}_{1,\epsilon}(S_1) = \widetilde{\Upsilon}_1(S_1) \cap Z \cap [\epsilon, \infty)^4$ for $\epsilon = 0.001$. From definition, $\widetilde{\mathcal{X}}_{1,\epsilon}(S_1)$ has natural cell complex structure and is approximating set $\widetilde{\Upsilon}_1(S_1)$ in homology sense. For details and general result we refer reader to chapter 3. A geometric realization of this cell complex $\widetilde{\mathcal{X}}_{1,\epsilon}(S_1)$ is given in Figure 1.7 and the

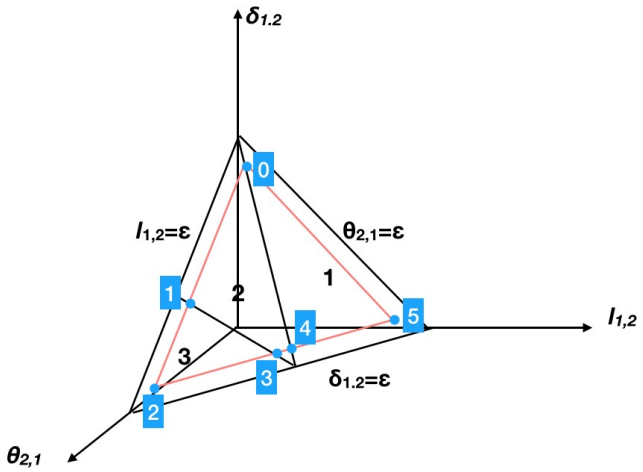


Figure 1.7: CW complex for node 1 parameter space in toggle switch. The regions $1, 2, 3$ is the approximation to the parameter regions $\{\gamma_1\theta_{2,1} < l_{1,2}\}$, $\{l_{1,2} < \gamma_1\theta_{2,1} < l_{1,2} + \delta_{1,2}\}$, $\{l_{1,2} + \delta_{1,2} < \gamma_1\theta_{2,1}\}$ respectively

.

cell-complex represented by their vertices is given by

$$\mathcal{X}^2 = \{\xi_1^2 = \{0, 4, 5\}, \xi_2^2 = \{0, 1, 3, 4\}, \xi_3^2 = \{1, 2, 3\}\}$$

$$\mathcal{X}^1 = \{\xi_1^1 = \{0, 5\}, \xi_2^1 = \{4, 5\}, \xi_3^1 = \{0, 4\}, \xi_4^1 = \{0, 1\}, \xi_5^1 = \{1, 3\}, \xi_6^1 = \{3, 4\},$$
$$\xi_7^1 = \{1, 2\}, \xi_8^1 = \{2, 3\}\}$$

$$\mathcal{X}^0 = \{\xi_1^0 = \{0\}, \xi_2^0 = \{1\}, \xi_3^0 = \{2\}, \xi_4^0 = \{3\}, \xi_5^0 = \{4\}, \xi_6^0 = \{5\}\}$$

where the incidence number κ is given by $\kappa(\xi_j^i, \xi_l^k) = 1$ if $i = k + 1$ and $\xi_l^k \subset \xi_j^i$ else 0. Note the number field we used in homology computation is always Z_2.

Once we have constructed the CW complex for the whole parameter space denote as $\widetilde{\mathcal{X}}_\epsilon(S)$, we can apply standard tools, like SageMath, to compute the homology group for some given regions.

In the case of the toggle switch, the DSGRN database, Figure 1.6, indicates that the only parameter node at which the dynamics indicates bistability is 5 and its corresponding homology group $\mathrm{Hom}(\widetilde{\mathcal{X}}_\epsilon(S))$ is isomorphic to the homology group of a point. For all nodes complementary to 5 the DSGRN database reports monostability. Let S

denote this set of nodes, we determine that

$$\mathrm{Hom}_k(\widetilde{\mathcal{X}}_\epsilon(S)) \cong \begin{cases} \mathsf{Z}_2 & \text{if } k = 0, 1 \\ 0 & \text{otherwise.} \end{cases}$$

In chapter 3, we show the singluar homology $\mathrm{Hom}(\widetilde{\Upsilon}(S))$ is isomorphic to cellular homology $\mathrm{Hom}(\widetilde{\mathcal{X}}_\epsilon(S)$ for any S in parameter graph.

Chapter 2

Computing linear extensions for Boolean lattices with algebraic constraints

In this chapter we consider the classical problem of computing linear extensions of a given poset which is well known to be a difficult problem. However, in our setting the elements of the poset are multi-variate polynomials, and only a small "admissible" subset of these linear extensions, determined implicitly by the evaluation map, are of interest. This seemingly novel problem arises in the study of global dynamics of gene regulatory networks in which case the poset is a Boolean lattice. We provide an algorithm for solving this problem using linear programming for arbitrary partial orders of linear polynomials. This algorithm exploits this additional algebraic structure inherited from the polynomials to efficiently compute the admissible linear extensions. The biologically relevant problem involves multi-linear polynomials and we provide a construction for embedding it into an instance of the linear problem.

2.1 Introduction

Consider a set of real polynomials $\mathcal{P}$, defined on a domain $\Xi \subset \mathsf{R}^d$, equipped with a partial order $\prec$. We are interested in identifying linear extensions (total orders compatible with $\prec$) that are satisfied by $\mathcal{P}$ under evaluation at a point in Ξ.

To be more precise consider a semi-algebraic set $\Xi \subset \mathsf{R}^d$, called the *evaluation domain*, and a collection of polynomials $\mathcal{P} := \{p_0, \ldots, p_K\} \subset \mathsf{R}[x_1, \ldots, x_d]$. Let $\prec$ denote a partial order on $\mathcal{P}$ such that if $p \prec q$, then

$$p(\xi) < q(\xi) \quad \text{for all } \xi \in \Xi. \tag{2.1}$$

Let S_{K+1} denote the set of permutations on $K+1$ symbols. We identify linear extensions of $\mathcal{P}$ with a subset of S_{K+1} as follows. Given $\sigma \in S_{K+1}$, let $\prec_\sigma$ denote the linear order

$$p_{\sigma(0)} \prec_\sigma p_{\sigma(1)} \prec_\sigma \cdots \prec_\sigma p_{\sigma(K)}.$$

We define the *realizable set* associated to σ by

$$\Xi_\sigma := \left\{\xi \in \Xi : p_{\sigma(k)}(\xi) < p_{\sigma(k+1)}(\xi) \text{ for all } 0 \leq k \leq K-1\right\}. \tag{2.2}$$

Observe that if $\Xi_\sigma \neq \emptyset$, then $\prec_\sigma$ is a linear extension of $\prec$. The *algebraically constrained linear extension problem* (AC-LEP) defined by $(\mathcal{P}, \prec, \Xi)$ is to determine

$$\mathcal{T}(\mathcal{P}, \prec, \Xi) := \{\sigma \in S_{K+1} : \Xi_\sigma \text{ is nonempty}\}.$$

Notice that in the formulation of the AC-LEP we have identified the partial order $\prec_\sigma$ with the associated element in S_{K+1}. We will use this identification throughout the remainder of the chapter. We say that each $\sigma \in \mathcal{T}(\mathcal{P}, \prec, \Xi)$ is an *admissible* linear extension.

As is discussed in Chapter 1 our motivation for study the AC-LEP comes from modeling the dynamics of regulatory networks in biology and in particular characterizing relevant subsets of parameter space. For the moment we attempt to put this problem into a broader mathematical context as the problem itself, as well as our solutions for some special cases, have elements of both classical real algebraic geometry and order theory.

Quantifier elimination and real algebraic geometry

Observe that if $\Xi = \mathsf{R}^d$, $\mathcal{P}$ is an arbitrary collection of polynomials, then $\sigma \in S_{K+1}$ is admissible if and only if there exists $\xi \in \mathsf{R}^d$ such that $p_{\sigma(k)}(\xi) - p_{\sigma(k+1)}(\xi) < 0$ for all $0 \leq k \leq K$. These inequalities define a semi-algebraic set and therefore, taking $\prec$ to be the trivial partial order (i.e. $\mathcal{P}$ is a single anti-chain), this instance of AC-LEP is equivalent to the classical problem of decidability for real semi-algebraic sets.

The previous example illustrates a major challenge in solving the AC-LEP. The first general algorithm for solving the quantifier elimination/decidability problem for polynomials in R^d with feasible running time was the cylindrical algebraic decomposition (CAD) given by Collins [13] in 1975. The CAD algorithm works by subdividing Ξ into subsets on which the polynomials are sign invariant. Given such a decomposition, decidability is reduced to simply evaluating each polynomial at a sample point located in each subset and checking if it satisfies the necessary inequalities. Unfortunately, the computational complexity of the algorithm grows like

$$\mathcal{O}\left((2D)^{2^d-1}(K+1)^{2^d-1}2^{2^{d-1}-1}\right) \quad \text{where} \quad D = \max\{\deg p : p \in \mathcal{P}\}.$$

This worst case running time is known to be sharp even for classes of "nice" polynomials e.g. linear [12], and moreover, the worst case is also typical [5]. As a result, the question of whether or not $\sigma \in \mathcal{T}(\mathcal{P}, \prec, \Xi)$, even for a single $\sigma \in S_{K+1}$, is often intractable for problems of practical interest.

In addition, this algorithm does not provide partial information. It either runs to completion, in which case it is guaranteed to provide an answer, or it provides no information. Furthermore, we note that if additional algebraic constraints are added e.g. we assume $\Xi_0 \subset \Xi \subset \mathsf{R}^d$ is a strict semi-algebraic subset, then the CAD algorithm can handle this by simply appending the polynomial constraints which define Ξ_0 to the set of polynomials. However, this dramatically increases the complexity of the CAD algorithm, despite the fact that the number of admissible linear extensions can only decrease.

Some improved algorithms have been proposed which aim to reduce the complexity of specific aspects of the problem or for special classes of polynomials (e.g. [22, 10, 11]). These improvements often provide dramatic algorithmic speedups for checking whether $\sigma \in \mathcal{T}(\mathcal{P}, \prec, \Xi)$ for a single linear extension. However, these algorithms have the same worse case running time as the general CAD algorithm and understanding which classes of polynomials benefit is still a very active area of research. Therefore, these improved algorithms alone are not sufficient to handle instances of AC-LEP since we are interested

in determining which of the $(K+1)!$ possible semi-algebraic sets are nonempty. An efficient algorithm which does not produce a decomposition of Ξ into sign invariant subsets would still need to be called $(K+1)!$ times. However, we will make use of these improved algorithms as a post-processing step which we discuss further in Section 2.3.

Computing linear extensions of Boolean lattices

Let us momentarily ignore the algebraic structure in the AC-LEP by forgetting that $\mathcal{P}$ is a collection of polynomials. Hence, we focus only on the poset structure $(\mathcal{P}, \prec)$, and consider the problem of computing all linear extensions. The related problem of counting all linear extensions of a partial order is a well studied problem in order theory. Its importance is due in large part to its connection with the complexity of sorting elements in a list. If one considers a list of $(K+1)$ distinct values which have been partially sorted by making pairwise comparisons on a subset of its elements, then these comparisons induce a partial order. Therefore, the linearly ordered values of the fully sorted list are given by one of the possible linear extensions for the partial order. As a result, the complexity of completely sorting a list is intimately connected to counting linear extensions for posets.

Observe that computing the set of all linear extensions of $(\mathcal{P}, \prec)$ is not easier than counting them which is known to be $\#P$-complete [8]. In particular, a polynomial time algorithm for computing all possible linear extensions for arbitrary posets would imply that $P = NP$ by Toda's theorem [31]. Moreover, we are interested not only in counting linear extensions, but explicitly computing them. Therefore, we are also concerned with how fast the number of admissible linear extensions grows.

For reasons we discuss in Section 2.3, we are specifically interested in the case that $(\mathcal{P}, \prec)$ is a Boolean lattice. Specifically, for fixed $n \in \mathsf{N}$, define $S_n := \{1, \ldots, n\}$ and let 2^{S_n} denote its power set. The *standard* n-dimensional Boolean lattice is the poset, $(2^{S_n}, \prec_B)$, where $\prec_B$ is the partial order defined by inclusion. We say a poset, $(\mathcal{P}, \prec)$, is an n-dimensional Boolean lattice if $(\mathcal{P}, \prec)$ is order isomorphic to the standard n-dimensional Boolean lattice and we write $\prec_B$ in place of $\prec$.

Estimating the number of linear extensions for Boolean lattices was first considered

in [25] which established a nontrivial upper bound. Later, Brightwell and Tetali [9] proved the following result that essentially settles the question for all practical considerations. If $Q(n)$ denotes the number of linear extensions of an n-dimensional Boolean lattice, then

$$\frac{\log Q(n)}{2^n} = \log \binom{n}{\lfloor n/2 \rfloor} - \frac{3}{2}\log e + \mathcal{O}\left(\frac{\ln n}{n}\right). \tag{2.3}$$

The estimate in Equation (2.3) illustrates a major challenge in solving the instances of AC-LEP of interest in this chapter. Consider an instance of AC-LEP given by $(\mathcal{P}, \prec_B, \Xi)$ where $(\mathcal{P}, \prec_B)$ is an n-dimensional Boolean lattice and Ξ is any evaluation domain.

Suppose we had a black box for efficiently computing all linear extensions of a Boolean lattice denoted by $L \subset S_{K+1}$. Furthermore, assume we also had a "CAD"-like algorithm which could efficiently check if $\Xi_\sigma \neq \emptyset$. Then, one would need to call this algorithm only $\#L$-many times as opposed to $(K+1)!$ as we argued above. However, the growth estimate in Equation (2.3) implies that the number of calls to this algorithm would still grow exponentially.

This work

In this work we present efficient algorithms for solving two specific instances of the AC-LEP. The first is the *linearly constrained linear extension problem* (LC-LEP), in which $\mathcal{P}$ is a set of linear polynomials, Ξ is a polytope, and $\prec$ is an arbitrary partial order. We present an efficient algorithm for solving the LC-LEP in Section 2.2. The second instance of the AC-LEP, which we call the *parameter space decomposition* (PSD) problem and describe now (see Definition 2.3.4 for a precise definition), is motivated by an application from systems biology described in Chapter 1

Definition 2.1.1. *For* $n \in \mathbb{N}$, *an* interaction function *of order* n *is a polynomial in* n *variables,* $z = (z_1, \ldots, z_n)$, *of the form*

$$f(z) = \prod_{j=1}^{q} f_j(z) \tag{2.4}$$

where each factor has the form

$$f_j(z) = \sum_{i \in I_j} z_i$$

and the indexing sets $\{I_j : 1 \leq j \leq q\}$ *form a partition for* $\{1, \ldots, n\}$. *We define the* interaction type *of* f *to be* $\mathbf{n} := (n_1, \ldots, n_q)$ *where* n_j *denotes the number of elements in* I_j.

Remark 2.1.1. We leave it to the reader to check that the order of the indexing sets I_j does not matter for any of the analysis in this chapter. Therefore, for convenience in reporting results (see Section 2.4) we will always assume that

$$n_1 \geq n_2 \geq \cdots \geq n_q.$$

To define an instance of the PSD problem, fix an interaction function f of order n and let $\mathcal{P}$ be the collection of polynomials in the $2n$ positive real variables, $\{\ell_i, \delta_i : 1 \leq i \leq n\}$, obtained by evaluating $f(z)$ with each $z_i \in \{\ell_i, \ell_i + \delta_i\}$. Taking all possible combinations of z_i for $1 \leq i \leq n$ produces the polynomials for the PSD problem,

$$\mathcal{P} = \{p_0, \ldots, p_{2^n-1}\} \subset \mathsf{R}[\ell_1, \ldots, \ell_n, \delta_1, \ldots, \delta_n]. \tag{2.5}$$

In Section 2.3, we will define an indexing map between the 2^n elements of $\mathcal{P}$ and the standard n-dimensional Boolean lattice. Let $\prec_B$ denote the Boolean lattice partial order with respect to this index map, and set $\Xi = (0, \infty)^{2n}$. The *PSD problem* is the instance of the AC-LEP defined by $(\mathcal{P}, \prec_B, \Xi)$. In Section 2.3, we prove that $(\mathcal{P}, \prec_B, \Xi)$ satisfies Equation (2.1). However, we present some examples before continuing.

Example 2.1.2. *The simplest nonlinear PSD problem arises from the interaction function*

$$f(z) = (z_1 + z_2)z_3$$

which has interaction type, $\mathbf{n} = (2, 1)$*. The PSD polynomials for this interaction function are given by*

$$
\begin{aligned}
p_0 &= (\ell_1 + \ell_2)\ell_3 & p_4 &= (\ell_1 + \ell_2 + \delta_1)\ell_3 \\
p_1 &= (\ell_1 + \ell_2)(\ell_3 + \delta_3) & p_5 &= (\ell_1 + \ell_2 + \delta_1)(\ell_3 + \delta_3) \\
p_2 &= (\ell_1 + \ell_2 + \delta_2)\ell_3 & p_6 &= (\ell_1 + \ell_2 + \delta_1 + \delta_2)\ell_3 \\
p_3 &= (\ell_1 + \ell_2 + \delta_2)(\ell_3 + \delta_3) & p_7 &= (\ell_1 + \ell_2 + \delta_1 + \delta_2)(\ell_3 + \delta_3).
\end{aligned}
$$

The PSD evaluation domain is $\Xi = (0, \infty)^6$ *and the partial order,* $\prec_B$*, is imposed on* $\mathcal{P}$ *by identifying* p_i *with the vertex of a unit cube whose coordinates are* $(i)_2 \in \mathsf{F}_2^3$ *where* $(i)_2$ *is the binary expansion of* i*. The solution to this PSD problem is the set of admissible linear extensions of* $(\mathcal{P}, \prec_B)$*, such that* $\sigma \in \mathcal{T}\left(\mathcal{P}, \prec_B, (0, \infty)^6\right)$ *if and only if* $\Xi_\sigma \neq \emptyset$*. We note that there are* $8! = 40,320$ *linear orders on* $\mathcal{P}$*. However, only* 48 *of these are linear extensions of* $\prec_B$*, and of these, only the following* 20 *linear extensions are admissible.*

$(0,1,2,3,4,5,6,7)$	$(0,1,2,4,6,3,5,7)$	$(0,1,4,5,2,3,6,7)$	$(0,2,1,4,6,3,5,7)$
$(0,1,2,3,4,6,5,7)$	$(0,1,4,2,5,3,6,7)$	$(0,1,4,5,2,6,3,7)$	$(0,2,4,1,6,3,5,7)$
$(0,1,2,4,3,5,6,7)$	$(0,1,4,2,5,6,3,7)$	$(0,2,1,3,4,6,5,7)$	$(0,2,4,6,1,3,5,7)$
$(0,1,2,4,3,6,5,7)$	$(0,1,4,2,6,5,3,7)$	$(0,2,1,4,3,6,5,7)$	$(0,4,1,2,5,6,3,7)$
$(0,4,2,6,1,5,3,7)$	$(0,4,1,5,2,6,3,7)$	$(0,4,2,1,6,5,3,7)$	$(0,4,1,2,6,5,3,7)$

The 28 *\missing" linear extensions are those which do not satisfy certain algebraic constraints which are imposed by the polynomial structure. For example, observe that for xed* $\xi \in (0, \infty)^6$*, if* $p_3(\xi) < p_6(\xi)$*, then* $p_1(\xi) < p_4(\xi)$ *must also hold.*

Unlike the partial order which constrains all possible linear extensions, this order relation is conditional. Indeed, there exist choices of ξ *such that* $p_3(\xi) > p_6(\xi)$ *in which case there is no requirement imposed on the order of* $p_1(\xi), p_4(\xi)$*, and in fact, there are admissible linear extensions which satisfy both choices, e.g. the rst two orders in column four. As another example, observe that* $p_5(\xi) < p_6(\xi)$*, if and only if* $p_1(\xi) <$

$p_2(\xi)$.

This algebraic constraint is bi-conditional, however, it also can not be represented in the partial order since both choices occur in at least one admissible order.

To emphasize the role of the interaction function in determining the algebraic constraints, we consider a similar PSD problem that is also an instance of LC-LEP.

Example 2.1.3. *Consider the interaction type,* $\mathbf{n} = (3)$ *with corresponding interaction function*

$$f = z_1 + z_2 + z_3.$$

As in Example 2.1.2, we obtain 8 *PSD polynomials given explicitly by*

$$\begin{aligned} p_0 &= \ell_1 + \ell_2 + \ell_3 & p_4 &= \ell_1 + \ell_2 + \ell_3 + \delta_1 \\ p_1 &= \ell_1 + \ell_2 + \ell_3 + \delta_3 & p_5 &= \ell_1 + \ell_2 + \ell_3 + \delta_1 + \delta_3 \\ p_2 &= \ell_1 + \ell_2 + \ell_3 + \delta_2 & p_6 &= \ell_1 + \ell_2 + \ell_3 + \delta_1 + \delta_2 \\ p_3 &= \ell_1 + \ell_2 + \ell_3 + \delta_2 + \delta_3 & p_7 &= \ell_1 + \ell_2 + \ell_3 + \delta_1 + \delta_2 + \delta_3 \end{aligned}$$

The evaluation domain and partial order are identical to the PSD problem in Example 2.1.2. Nevertheless, only the following 12 *linear extensions are admissible*

$$\begin{array}{cccc} (0,1,2,3,4,5,6,7) & (0,1,4,5,2,3,6,7) & (0,2,4,1,6,3,5,7) & (0,4,1,5,2,6,3,7) \\ (0,1,2,4,3,5,6,7) & (0,2,1,3,4,6,5,7) & (0,2,4,6,1,3,5,7) & (0,4,2,1,6,5,3,7) \\ (0,1,4,2,5,3,6,7) & (0,2,1,4,3,6,5,7) & (0,4,1,2,5,6,3,7) & (0,4,2,6,1,5,3,7) \end{array}$$

Similarly, the missing 36 linear extensions in this example fail to satisfy some algebraic constraints. In both cases, the set of admissible linear extensions is a fraction of the set of all linear extensions of the Boolean lattice. In other words, the algebraic structure implies that the admissible linear extensions are a sparse subset of all linear extensions. The algorithm in this chapter exploits the algebraic and order theoretic aspects of the PSD problem to overcome the computational complexity limitations which plague both problems in general. Furthermore, we prove that this algorithm finds all

possible linear extensions. For both examples we obtained the (12 and 20 respectively) admissible solutions *without* first computing the linear extensions of $(\mathcal{P}, \prec_B)$ and then checking which are admissible.

Related work

As is indicated above our original motivation for this chapter arises from problems in systems biology for which explicit complete solutions to the PSD problem are required. As such the majority of this introduction has focused on the question of efficacy of computation. However, there is another direction in which the work of this chapter overlaps with other efforts. In particular, observe that the case where the interaction function is linear, i.e. has interaction type $\mathbf{n} = (1, \ldots, 1)$, solving the AC-LEP is equivalent to identifying all the cells of a hyperplane arrangement. This latter problem has been the subject of considerable study (see [27] for an introduction) and in particular, Maclagan [21] provides the number of solutions for the linear PSD problem for $n = 1, \ldots, 7$. Our computations (see Table 2.1) lead to the same numbers, as expected.

After accounting for symmetry in the number of linear PSD solutions for interaction types $(1, 1, 1, 1)$, $(1, 1, 1, 1, 1)$, and $(1, 1, 1, 1, 1, 1)$, reported in column two of Table 2.1, we obtain

$$\frac{336}{4!} = 14 := a_4, \quad \frac{61920}{5!} = 516 := a_5, \quad \frac{89414640}{6!} = 124187 := a_6$$

which align with sequence A009997 in the OEIS [23]. From [17], we know this sequence represents the number of comparative probability orderings on all subsets of n elements that can arise by assigning a probability distribution to the individual elements. The equivalence of comparative probability orderings and solutions to the linear PSD problem follows directly from the definition of comparative probability.

Organization of chapter

The remainder of this chapter is organized as follows. In Section 2.1 we briefly describe how the PSD problem arises naturally in the study of global dynamics for gene regulatory networks. In Section 2.2, we present an efficient algorithm for solving instances of the LC-LEP. In Section 2.3, we show that the LC-LEP is related to the PSD problem in the following way. If $\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ is an instance of the PSD problem, then we construct an associated instance of LC-LEP, denoted by $(\mathcal{P}', \prec_B, \Xi')$, which satisfies the inclusion

$$\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) \subseteq \mathcal{T}\left(\mathcal{P}', \prec_B, \Xi'\right). \tag{2.6}$$

We refer to this instance of LC-LEP as the *linearized* PSD problem associated to $\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$. We exploit this construction and the algorithm for solving the LC-LEP presented in Section 2.2, to provide a means of efficiently computing a collection of candidates that contains the solution to the PSD problem. We prove that in some cases the inclusion in Equation (2.6) is actually an equality. More generally, this inclusion is strict, but the candidate set is a sparse subset of the collection of all linear extensions of $(\mathcal{P}, \prec_B)$. In this case we describe algorithms for removing the non-admissible solutions.

Finally, in Section 2.4 we present the results for all PSD solutions with order up to four. Additionally, we have some results for PSDs or order five and size. For the remaining cases and PSDs of higher order the computations become too large.

2.2 Solving the LC-LEP

In this Section, we provide an efficient algorithm to solve the LC-LEP defined in Section 2.1. Note that if $q \in \mathsf{R}[x_1, \ldots, x_d]$ is a linear polynomial, then evaluation of q defines a linear functional on R^d. Thus, there exists a unique vector $\mathbf{u}_q \in \mathsf{R}^d$, that we call the *representation vector* for q, satisfying

$$q(\xi) = \mathbf{u}_q \cdot \xi \qquad \text{for all} \quad \xi \in \mathsf{R}^d.$$

Equivalently, $\mathbf{u}_q$ is the vector of coefficients of q. Since the evaluation domain for the LC-LEP is a polytope, there exists a collection of linear polynomials $\mathcal{Q}_\Xi$, such that

$$\Xi = \left\{ \xi \in \mathsf{R}^d : \xi \cdot \mathbf{u}_q > 0 \text{ for all } \quad q \in \mathcal{Q}_\Xi \right\}. \tag{2.7}$$

We assume that (2.7) is satisfied for the remainder of this section.

To foreshadow our approach recall that by definition $\sigma \in \mathcal{T}(\mathcal{P}, \prec, \Xi)$ if and only if $\Xi_\sigma \neq \emptyset$. Our approach to determining the latter is to recast it in the language of linear algebra on cones in R^d. From this perspective, the problem is equivalent to rigorously solving a linear programming problem and the efficacy of our algorithm is based on the fact that this can be done efficiently. With this goal in mind, we begin with a few remarks concerning cones and ordered vector spaces.

2.2.1 Cones

Definition 2.2.1. *A subset $C \subset \mathsf{R}^d$ is a* cone *if $v \in C$ and $\theta \in [0, \infty)$ implies that $\theta v \in C$. The cone C is* pointed *if it is closed, convex, and satisfies*

$$C \cap -C = C \cap \{-v : v \in C\} = 0. \tag{2.8}$$

Observe that (2.8) implies that a pointed cone does not contain any lines. A vector $v \in \mathsf{R}^d$ is a *conic combination* of $\{v_1, \ldots, v_k\} \subset \mathsf{R}^d$ if $v = \theta_1 v_2 + \cdots + \theta_k v_k$ where $\theta_1, \ldots, \theta_k \geq 0$. Suppose $V = \{v_1, \ldots, v_k\} \subset \mathsf{R}^d$. The *conic hull* of V is given by

$$\text{cone}(V) := \left\{ \sum_{i=1}^{k} \theta_i v_i : 0 \leq \theta_i, \ i = 1, \ldots, k \right\}.$$

The following result is left to the reader to check.

Proposition 2.2.2. *Given $V = \{v_1, \ldots, v_k\} \subset \mathsf{R}^d$, $\text{cone}(V)$ is the smallest closed convex cone that contains V.*

We make use of the following propositions.

Proposition 2.2.3. *Suppose $V = \{v_0, \ldots, v_m\} \subset \mathsf{R}^d$ is a collection of nonzero vectors such that* $\mathrm{cone}(V)$ *is a pointed cone. Then, there exists some $v' \in \mathsf{R}^d$ such that $v' \cdot v_i > 0$ for all $0 \leq i \leq m$.*

Proof. Observe that $-v_m \notin \mathrm{cone}(\{v_0, \ldots, v_{m-1}\}) \subset \mathrm{cone}(V)$ since $\mathrm{cone}(V)$ is pointed. Hence $\{-v_m\}$ and $\mathrm{cone}(\{v_0, \ldots, v_{m-1}\})$ are disjoint, convex, closed subsets of R^d.

Therefore, by the hyperplane separation theorem [6], there exists $v' \in \mathsf{R}^d$ such that $v' \cdot -v_m < 0$ and $v' \cdot v > 0$ for any $v \in \mathrm{cone}(\{v_0, \ldots, v_{m-1}\})$. □

Proposition 2.2.4. *Suppose $V = \{v_0, \ldots, v_m\} \subset \mathsf{R}^d$ is a collection of nonzero vectors such that* $\mathrm{cone}(V)$ *is a pointed cone. If $-v \notin \mathrm{cone}(V)$, then* $\mathrm{cone}(V \cup \{v\})$ *is pointed.*

Proof. Suppose that $\mathrm{cone}(V \cup \{v\})$ is not pointed. Then, there exists $w \neq 0$ such that $w, -w \in \mathrm{cone}(V \cup \{v\})$ or equivalently

$$w = \sum_{i=0}^{m} \alpha_i v_i + \alpha v \quad \text{and} \quad -w = \sum_{i=0}^{m} \beta_i v_i + \beta v$$

where $\alpha_i, \beta_i, \alpha, \beta$ are all nonnegative. Note that if $\alpha = \beta = 0$, then $\pm w \in \mathrm{cone}(V)$, which contradicts the assumption that V is pointed. The sum of the two equations above is

$$-(\alpha + \beta)v = \sum_{i=0}^{m} (\alpha_i + \beta_i) v_i.$$

This implies that $-v \in \mathrm{cone}(V)$, contradicting the assumption that $\mathrm{cone}(V)$ is pointed. □

The previous propositions illustrate the importance of solving the *cone inclusion* problem: given a vector $v \in \mathsf{R}^d$, and finite set of vectors $V \subset \mathsf{R}^d$, determine whether or not $v \in \mathrm{cone}(V)$. Algorithm 1, stated below, solves this problem. Observe that checking if $v \in \mathrm{cone}(V)$ is equivalent to solving the following linear programming feasibility

problem.

$$\begin{aligned} &\text{Does there exist} \quad \alpha \\ &\text{such that} \quad \mathbf{V}\alpha = v \\ &\text{and} \quad \alpha \geq 0? \end{aligned} \tag{2.9}$$

where $\mathbf{V}$ is the column matrix of V.

Linear programming is a powerful tool that is widely used in convex optimization, and as a result, there are many available solvers/algorithms for solving the linear programming feasibility problem [32]. The results can be made rigorous by performing computations using interval arithmetic [28] or rational linear programming [3] in the case that $\mathbf{V}$ is rational. Observe that the PSD problem defined in Section 2.1 satisfies this constraint. As is made clear in Section 2.4, we use different solvers depending on the machine employed to do the computations. In any case, we take for granted the existence of a rigorous solver for the feasibility problem in Equation 2.9 as a "black box" which we call `LPSolver` which is employed in the following algorithm.

Algorithm 1: Cone inclusion

Input: $v, V = \{v_1, \ldots, v_m\}$, `LPSolver`

Output: **True**, **False**

Result: Return **True** if $v \in \text{cone}(V)$ otherwise **False**

```
1 Function InCone(v, V, LPSolver):
2 |   Return LPSolver(v, V)
3 End Function
```

The next algorithm uses Proposition 2.2.4 (see line 4) and Algorithm 1 to determine

if a set of vectors defines a pointed cone.

Algorithm 2: Cone pointedness

Input: $V = \{v_1, \ldots, v_m\}$

Output: **True**, **False**

Result: Return **True** if $\text{cone}(V)$ is pointed otherwise **False**

```
Function CheckCone(V):
    V' = {v_1}
    for i = 2 ... m do
        if InCone(−v_i, V') then
            Return False
        else
            V' = V' ∪ {v_i}
        end
    end
    Return True
End Function
```

We now show that the LC-LEP can be reformulated as a problem of identifying whether some specific subsets of vectors generate pointed cones.

Definition 2.2.5. *Given an instance of the LC-LEP,* $(\mathcal{P}, \prec, \Xi)$*, we define the* base cone *as* $\text{cone}(V_0) := \text{cone}(V_\Xi \cup V_\prec)$ *where* V_Ξ *and* $V_\prec$ *are defined as follows. Set*

$$V_\Xi := \{\mathbf{u}_q : q \in \mathcal{Q}_\Xi\}$$

where $\mathcal{Q}_\Xi$ *are the representation vectors as defined in Equation* (2.7)*. Applying Algorithm 2 to* V_Ξ *(and the fact that we assume* $\Xi \neq \emptyset$*) shows that* $\text{cone}(V_\Xi)$ *is pointed. Define*

$$V_\prec := \{\mathbf{u} : \mathbf{u} \text{ is the representing vector of } p - q \text{ where } q \prec p \text{ and } p, q \in \mathcal{P}\}.$$

Observe that if $(\mathcal{P}, \prec)$ *satisfies Equation* (2.1)*, then the representation vector for* $p - q$ *is an element of* $V_\prec$ *by definition. Therefore, by Proposition 2.2.3,* $V_\prec$ *is pointed.*

The motivation behind our definition of V_0 is that it characterizes the algebraic constraints in the LC-LEP in terms of linear algebra that can be efficiently checked. The next proposition shows that the same idea works for linear extensions.

Given $\sigma \in S_{K+1}$, we define

$$V_\sigma := V_0 \cup \left\{ \mathbf{u}_{p_{\sigma(i+1)}} - \mathbf{u}_{p_{\sigma(i)}} : p_{\sigma(i)} \in \mathcal{P},\ i = 0, \ldots, K-1 \right\}. \tag{2.10}$$

Proposition 2.2.6. *For any $\sigma \in S_{K+1}$, $\Xi_\sigma \neq \emptyset$ if and only if* $\mathrm{cone}(V_\sigma)$ *is a pointed cone.*

Proof. First we assume $\Xi_\sigma \neq \emptyset$ and that $\xi \in \Xi_\sigma$. On one hand, if $\mathrm{cone}(V_\sigma)$ is not pointed, then there are nonzero vectors $-v, v \in \mathrm{cone}(V_\sigma)$. However, the definition of Ξ_σ implies that $-v \cdot \xi > 0$ and $v \cdot \xi > 0$, which is a contradiction.

On the other hand, if $\mathrm{cone}(V_\sigma)$ is pointed, then by Proposition 2.2.3 there exists a $\xi_0 \in \mathsf{R}^d$ such that $\xi_0 \cdot v > 0$, for all $v \in V_\sigma$. Thus, the definition of V_σ implies that $\xi_0 \in \Xi_\sigma$ and hence $\Xi_\sigma \neq \emptyset$. □

We emphasize that the importance of Proposition 2.2.6 is the implied equivalence

$$\mathcal{T}(\mathcal{P}, \prec, \Xi) = \{\sigma : \Xi_\sigma \neq \emptyset\} = \{\sigma : \mathrm{cone}(V_\sigma) \text{ is pointed}\}.$$

2.2.2 An algorithm for identifying $\mathcal{T}(\mathcal{P}, \prec, \Xi)$

In this section we present an algorithm for solving an arbitrary instance of the LC-LEP.

Algorithm 3: LC-LEP solver

Input: $\sigma_{\text{part}} = [\,], \mathcal{P}, V = V_0, \texttt{Ret} = \{\}$

Output: $\mathcal{T}(\mathcal{P}, \prec, \Xi)$

Result: `Ret`: collection of all linearly realizable total order under restriction of V

```
Function OrderingGenerator(σ_part, P, V, Ret):
    if σ_part == [ ] and CheckCone(V) is not True then
        Return
    end
    l + 1 = length of σ_part
    if l == K then
        add σ_part to Ret
        Return
    end
    for i = 0 .. K do
        if i ∉ σ_part then
            u' = u_{p_i} − u_{p_{σ_part(l)}}
            if not InCone(−u', V) then
                OrderingGenerator(σ_part + [i], P, V ∪ {v'}, Ret)
            end
        end
    end
End Function
```

In the Algorithm 3, for convenience, we take $\mathbf{u}_{p_{\sigma_{\text{part}}(-1)}} = 0$. To prove the correctness of the algorithm it is useful to denote the return of Algorithm 3 given input $(\mathcal{P}, \prec, \Xi)$ as $\mathcal{T}_{alg}(\mathcal{P}, \prec, \Xi)$.

Definition 2.2.7. *For xed $(\mathcal{P}, \prec, \Xi)$, $\sigma \in S_{K+1}$ and for $k = 1, \cdots, K$, de ne*

$$V_{\sigma,k} = \{\mathbf{u}_{p_{\sigma(i)}} - \mathbf{u}_{p_{\sigma(i-1)}}\}_{i=1,\ldots,k} \cup V_0,$$

where V_0 is the base cone for $(\mathcal{P}, \prec, \Xi)$ as in De nition 2.2.5, and $\mathbf{u}_{p_{\sigma(j)}}, j = 0, \ldots, K$ is the representation vector of $p_j \in \mathcal{P}$. For convenience, we de ne $V_{\sigma,0} = V_0$ and we observe that $V_{\sigma,K} = V_\sigma$ from Equation (2.10).

Theorem 2.2.1. *Algorithm 3 solves the LC-LEP.*

Proof. Given $(\mathcal{P}, \prec, \Xi)$, we need to show that $\mathcal{T}(\mathcal{P}, \prec, \Xi) = \mathcal{T}_{alg}(\mathcal{P}, \prec, \Xi)$. We may assume that cone(V_0) is pointed since if not, then both $\mathcal{T}_{alg}(\mathcal{P}, \prec, \Xi)$ and $\mathcal{T}(\mathcal{P}, \prec, \Xi)$ are empty.

We first show that $\mathcal{T}_{alg}(\mathcal{P}, \prec, \Xi) \subset \mathcal{T}(\mathcal{P}, \prec, \Xi)$, i.e. for any $\sigma \in \mathcal{T}_{alg}(\mathcal{P}, \prec, \Xi)$ we show that the set $\Xi_\sigma \neq \emptyset$. As indicated above we assume cone(V_0) = cone$(V_{\sigma,0})$ is pointed. For Algorithm 3, lines 2-4 returns the empty set if cone(V_0) is not pointed. Otherwise, it passes to lines 5-9 which checks if σ_{part} is a total order over $\{0, \ldots, K\}$. If so, it is added to the return variable, `Ret`. If σ_{part} is not a total order, then lines 10-17 extend it to a total order by recursively constructing $V_{\sigma,i}$ from $V_{\sigma,i-1}$ for $1 \leq i \leq K$.

Therefore, it suffices to show that $V_{\sigma,k}$ are all pointed for $k = 1, \ldots, K$.

Fix $k \in \{1, \ldots, K\}$. In lines 11-12, we find a candidate $i \in \{0, \ldots, K\}$ which is not in the image of σ_{part}, and define $V_{\sigma,k+1} = V_{\sigma,k} \cup \{\mathbf{u}'\}$ where $\mathbf{u}' = \mathbf{u}_{p_i} - \mathbf{u}_{p_{\sigma(k)}}$. In line 13, we verify that $-\mathbf{u}' \notin V = V_{\sigma,k-1}$ and it follows from Proposition 2.2.4, that cone$(V_{\sigma,k})$ is pointed. For each σ appended to `Ret`, we have cone$(V_{\sigma,k})$ is pointed for $k = 0, \ldots, K$. In particular, cone$(V_{\sigma,K})$ = cone(V_σ) is pointed, and from Proposition 2.2.6, we have $\Xi_\sigma \neq \emptyset$.

We now prove that $\mathcal{T}(\mathcal{P}, \prec, \Xi) \subset \mathcal{T}_{alg}(\mathcal{P}, \prec, \Xi)$. Assume that $\sigma \in \mathcal{T}(\mathcal{P}, \prec, \Xi)$. By definition this means that $\Xi_\sigma \neq \emptyset$ and from Proposition 2.2.6, V_σ is pointed. For each $k = 1, \ldots, K$, we have $V_{\sigma,k} \subset V_\sigma$, and thus $V_{\sigma,k}$ is pointed. As $V_{\sigma,k}$ is pointed, we know $-(\mathbf{u}_{p_{\sigma(k)}} - \mathbf{u}_{p_{\sigma(k-1)}}) \notin V_{\sigma,k-1}$ for $k = 1, \ldots, K$. Therefore, line 13 in Algorithm 3 will not fail to extend σ at each step in the recursion and after K recursive extensions, σ

will be appended to Ret and thus, $\sigma \in \mathcal{T}_{alg}(\mathcal{P}, \prec, \Xi)$. □

2.3 Solving the general PSD problem

In this section we present a solution for the PSD problem described in Section 2.1. The solution is based on the observation that the PSD problem has a natural Boolean lattice structure. Thus, for the linear PSD problem, the LC-LEP solver described in Section 2.2 provides a solution. For nonlinear PSD problems, we construct a map that "embeds" it into an instance of LC-LEP (of higher dimension) in the sense that the inclusion in Equation (2.6) holds. We prove a sufficient condition for which this inclusion is equality and describe a method for disqualifying spurious solutions when it is strict.

2.3.1 The PSD as an instance of AC-LEP

Throughout this section $\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ denotes a PSD problem for a fixed interaction function f of order type $\mathbf{n} \in \mathsf{N}^q$ as defined in Equation (2.5) where $\prec_B$ is the Boolean lattice partial order. Our first goal is to show that $\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ satisfies Equation (2.1), and in particular, that every $\sigma \in \mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ is a linear extension of a Boolean lattice. We start by defining appropriate indices for the elements of $\mathcal{P}$.

Definition 2.3.1. *Suppose* $\mathbf{n} \in \mathsf{N}^q$ *is the interaction type for* $f \in \mathsf{R}[z_1, \ldots, z_n]$. *As in De nition 2.1.1 let* $\{I_1, \ldots, I_q\}$ *denote the indexing sets for each summand of* f. *Setting* $I := \bigcup_{j=1}^{q} I_j$ *we denote a typical element of* I *by* $I_j(k)$ *which we de ne as the* k^{th} *largest element of* I_j. *Let* $E := \{\alpha : \{1, \ldots, n\} \to \{0,1\}\}$ *be the set of all Boolean functions de ned on* I. *The* Boolean indexing map, *denoted by* $B : E \to \left\{0, \ldots, 2^{n-1}\right\}$, *is de ned by the formula*

$$B(\alpha) := \sum_{j=1}^{q} \sum_{k=0}^{n_j - 1} \alpha(I_j(k)) 2^{\kappa_{j,k}} \qquad \kappa_{j,k} = k + \sum_{j'=1}^{j-1} n_{j'}.$$

We will also consider, $\alpha \in E$, as a vector of Boolean functions defined as follows. Let E_j denote the set of Boolean functions defined on I_j. Then, elements of E can be

represented as vectors of the form

$$\alpha = (\alpha_1, \ldots, \alpha_q) \quad \text{where } \alpha_j := \alpha\Big|_{I_j} \in E_j \quad \text{for } 1 \leq j \leq q.$$

Note that under this identification, E has the equivalent representation as $E = E_1 \times \cdots \times E_q$.

Definition 2.3.2. *Suppose* $\mathbf{n} \in \mathsf{N}^q$ *is the interaction type for an interaction function,* $f \in \mathsf{R}[z_1, \ldots, z_n]$ *as in De nition 2.1.1, and* E *denotes the corresponding Boolean indices. For* $\alpha \in E$*, de ne* $p_\alpha \in \mathcal{P} \subset \mathsf{R}[\ell_1, \ldots, \ell_n, \delta_1, \ldots, \delta_n]$*, by the formula*

$$p_\alpha := \prod_{j=1}^{q} \left(\sum_{k \in I_j} \ell_k + \alpha(k)\delta_k \right). \tag{2.11}$$

When convenient, we use a linear indexing scheme for elements of $\mathcal{P}$ which we define via the Boolean indexing map by identifying $p_i := p_\alpha$ where $\alpha = B^{-1}(i)$. To avoid confusion, we exclusively use Greek subscripts when referring to elements of $\mathcal{P}$ by their Boolean indices, and Latin subscripts when referring to elements of $\mathcal{P}$ by their linear indices. We leave it to the reader to check that the linearly indexed polynomials in Examples 2.1.2 and 2.1.3 are in agreement with that of Definition 2.3.2 via this identification.

Definition 2.3.3. *Let* $\alpha, \beta \in E$ *be a pair of Boolean indices corresponding to* $\mathbf{n} \in \mathsf{N}^q$*. An ordered pair* (α, β) *satis es the* one bit condition *if* $\alpha(I_j(k)) \leq \beta(I_j(k))$*, for all* $1 \leq j \leq q$ *and* $0 \leq k \leq n_j - 1$*, with equality for all but* exactly *one* (j, k) *pair.*

Remark 2.3.1. Observe that if (α, β) satisfy the one bit condition and (j_0, k_0) is the unique pair for which α and β take different values, then $\alpha(I_{j_0}(k)) = 0$ and $\beta(I_{j_0}(k)) = 1$.

Remark 2.3.2. The one bit condition induces a poset structure on E by setting $\alpha \prec \beta$ for each (α, β) satisfying the one bit condition, and extending the relation transitively.

Definition 2.3.4. *Suppose* $\mathbf{n} \in \mathsf{N}^q$ *is the interaction type for an interaction function,* $f \in \mathsf{R}[z_1, \ldots, z_n]$ *as in De nition 2.1.1, and* E *denotes the corresponding Boolean*

indices. Let $\mathcal{P}$ be the set of polynomials indexed as in Definition 2.3.2. The PSD problem *is defined by the triple, $(\mathcal{P}, \prec, (0,\infty)^{2n})$ where $\prec$ is given by Remark 2.3.2.*

The next proposition proves that $(\mathcal{P}, \prec, (0,\infty)^{2n})$ satisfies Equation (2.1), and furthermore that $(\mathcal{P}, \prec)$ is a Boolean partial order which justifies expressing the PSD problem as $\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$.

Proposition 2.3.5. *Consider a PSD problem $(\mathcal{P}, \prec, (0,\infty)^{2n})$. Then,*

1. *$(\mathcal{P}, \prec)$ is a Boolean lattice.*
2. *For any $\alpha, \beta \in E$, if $\alpha \prec \beta$, then*

$$p_\alpha(\xi) < p_\beta(\xi) \quad \textit{for all} \quad \xi \in (0,\infty)^{2n}.$$

Proof. To prove the first claim, let $S_n = \{1, \ldots, n\}$ and let $(2^{S_n}, \prec_B)$ denote the standard Boolean lattice. Define a map, $\varphi : E \to 2^{S_n}$, by the formula

$$\varphi(\alpha) = \{j \in S_n : \alpha(j) = 1\},$$

and we note that φ is a bijection since E is defined to be the collection of all Boolean maps defined on S_n. Furthermore, for any $\alpha, \beta \in E$, we have by Definition 2.3.3 that $\alpha \prec \beta$ if and only if

$$\{j \in S_n : \alpha(j) = 1\} \subset \{j \in S_n : \beta(j) = 1\}$$

implying that φ is an order isomorphism.

To establish the second claim, we must show that if $\alpha \prec \beta$, then $p_\alpha(\xi) < p_\beta(\xi)$ holds for all $\xi \in (0,\infty)^{2n}$. Note that by transitivity, it suffices to prove this holds for (α, β) satisfying the one bit condition. In this case we have

$$\beta(I_j(k)) - \alpha(I_j(k)) = \begin{cases} 1 & \text{if } j = j_0 \text{ and } k = k_0 \\ 0 & \text{otherwise.} \end{cases}$$

for some $j_0 \in \{1, \ldots, q\}$, $k_0 \in \{0, \ldots, n_{j_0-1}\}$. If $\xi = (\ell_1, \ldots, \ell_n, \delta_1, \ldots, \delta_n) \in \Xi$, then from Equation (2.11) we have

$$\begin{aligned} p_\beta(\xi) &= \left(\left(\sum_{k \in I_{j_0}} \ell_k + \alpha(I_j(k))\delta_k \right) + \delta_{k_0} \right) \prod_{j \neq j_0} \sum_{k \in I_j} \ell_k + \alpha(I_j(k))\delta_k \\ &= p_\alpha(\xi) + \delta_{k_0} \prod_{j \neq j_0} \sum_{k \in I_j} \ell_k + \alpha(I_j(k))\delta_k \\ &> p_\alpha(\xi) \end{aligned}$$

as required. □

With Proposition 2.3.5 in mind, we return to writing $\prec_B$ in place of $\prec$ for the PSD problem where $\prec_B$ is the partial order of a Boolean lattice inherited by $\mathcal{P}$ from the one bit condition.

2.3.2 The linear PSD problem

We consider two cases of the PSD problem: the interaction type $\mathbf{n} \in \mathsf{N}^q$ for the function $f \in \mathsf{R}[z_1, \ldots, z_n]$ has the form $\mathbf{n} = (1, 1, \ldots, 1)$ or $\mathbf{n} = (n)$. In the first case, f is linear (see Example 2.1.3) and the PSD problem is an instance of LC-LEP. In the second case, $\log f$ is linear so after a simple change of variables, we obtain an instance of LC-LEP with equivalent solutions since log is monotone, hence order preservingł. We focus on the first case, leaving it to the reader to check that the second case is same modulo the evaluation domain (R^{2n} versus $(0, \infty)^{2n}$). Following the algorithm described in Section 2.2, we encode the partial order defined by $\prec_B$ as a set of linear constraints defined by a base cone which we must show is pointed. We begin by denoting the set of representation vectors for $\mathcal{P}$ as

$$\mathcal{V} := \left\{ \mathbf{u}_{p_\alpha} \in \{0,1\}^{2n} : \mathbf{u}_{p_\alpha} \text{ is the representation vector of } p_\alpha,\ \alpha \in E \right\}.$$

We define the set,

$$V_{\prec_B} := \left\{ \mathbf{u}_{p_\beta} - \mathbf{u}_{p_\alpha} : \mathbf{u}_{p_\alpha}, \mathbf{u}_{p_\beta} \in \mathcal{V},\ (\alpha, \beta) \text{ satisfies the one bit condition} \right\}. \quad (2.12)$$

which encodes the $\prec_B$ partial order into the representation vectors. These vectors will be the generators of the base cone for the algorithm in Section 2.2. Thus, we must show that cone($V_{\prec_B}$) generates a pointed cone.

Lemma 2.3.6. *Let $C_0 := \text{cone}(V_{\prec_B})$ denote the cone generated by $V_{\prec_B}$, then C_0 is pointed.*

Proof. By Proposition 2.2.2, C_0 is closed and convex so it suffices to prove that if $v \in C_0$ and $-v \in C_0$, then $v = 0$. Fix $\xi \in (0,\infty)^{2n}$ and suppose (α, β) satisfies the one bit condition. By the formula in Equation (2.11) it follows that

$$p_\beta - p_\alpha = \delta_i$$

for some $i \in \{1, \ldots, n\}$. Since $\delta_i = \xi_{n+i} > 0$ for all $\xi \in \Xi$, it follows that

$$p_\beta(\xi) - p_\alpha(\xi) > 0,$$

for every (α, β) satisfying the one bit condition. Passing to the representation vectors, it follows that for every $v \in V_{\prec_B}$, we have $v \cdot \xi > 0$. Taking the conic hull, we have that if $v \in C_0 \setminus \{0\}$, then $v \cdot \xi > 0$. It follows that if $v, -v \in C_0$ simultaneously, then $v \cdot \xi \geq 0$ and $-v \cdot \xi \geq 0$ implying $v = 0$. □

2.3.3 The general PSD problem

Given a general PSD problem $\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ we present the construction of a LC-LEP denoted by $(\mathcal{P}', \prec_B, \mathsf{R}^m)$ with the property that $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) \subseteq \mathcal{T}(\mathcal{P}', \prec_B, \mathsf{R}^m)$. The importance of this is that $(\mathcal{P}', \prec_B, \mathsf{R}^m)$ can be solved using Algorithm 3 and hence we obtain a rigorous upper bound on $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$.

Definition 2.3.7. *Given an interaction type $\mathbf{n} \in \mathsf{N}^q$, let $E = E_1 \times \cdots \times E_q$ denote the corresponding Boolean indices. Set $m := \sum_{j=1}^{q} 2^{n_j}$ and define the* linearized evaluation domain *to be*

$$\mathsf{R}^{2^{n_1}} \times \cdots \times \mathsf{R}^{2^{n_q}} \cong \mathsf{R}^m. \tag{2.13}$$

De ne a polynomial ring in m *indeterminates with Boolean indexing by*

$$\mathcal{R} := \mathsf{R}\left[\left\{x_{j,\alpha_j} : \alpha_j \in E_j, 1 \leq j \leq q\right\}\right], \tag{2.14}$$

and de ne a collection of linear polynomials by

$$\mathcal{P}' := \left\{p'_\alpha : \alpha \in E\right\} \subset \mathcal{R} \qquad \textit{where} \quad p'_\alpha := \sum_{j=1}^{q} x_{j,\alpha_j}.$$

The linearized PSD problem *determined by* $\mathbf{n}$ *is to compute* $\mathcal{T}\left(\mathcal{P}', \prec_B, \mathsf{R}^m\right)$.

Theorem 2.3.3. *Fix an interaction type,* $\mathbf{n} \in \mathsf{N}^q$ *and let* $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ *and* $\mathcal{T}\left(\mathcal{P}', \prec'_B, \mathsf{R}^m\right)$ *denote the corresponding PSD and linearized PSD problems, respectively. The following are true.*

1. *Let* $\alpha, \beta \in E$ *and* $\xi \in (0,\infty)^{2n}$. *If* $p_\alpha(\xi) < p_\beta(\xi)$ *and*

$$\xi'_{j,\alpha_j} = \log\left(\sum_{k \in I_j} \xi_k + \alpha_j(k)\xi_{n+k}\right) \in \mathsf{R}^m,$$

   then $p'_\alpha(\xi') < p'_\beta(\xi')$.

2. $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) \subseteq \mathcal{T}\left(\mathcal{P}', \prec'_B, \mathsf{R}^m\right)$.

Proof. To prove the first claim, we define a map, $T : (0,\infty)^{2n} \to \mathsf{R}^m$, by $\xi \mapsto \xi' := T(\xi)$ where the coordinates of ξ' are given by the formula

$$\xi'_{j,\alpha_j} = \log\left(\sum_{k \in I_j} \xi_k + \alpha_j(k)\xi_{n+k}\right). \tag{2.15}$$

Observe that T is defined to satisfy the functional equation

$$\log \circ p_\alpha(\xi) = p'_\alpha \circ T(\xi) \qquad \text{for all} \quad \alpha \in E,\ \xi \in (0,\infty)^{2n}. \tag{2.16}$$

Therefore, if $\alpha, \beta \in E$ and $\xi \in (0,\infty)^{2n}$ satisfies $p_\alpha(\xi) < p_\beta(\xi)$, then $\log\left(p_\alpha(\xi)\right) < \log\left(p_\beta(\xi)\right)$ and it follows from Equation (2.16) that $p'_\alpha(\xi') < p'_\beta(\xi')$ where $\xi' = T(\xi)$ as required.

To prove the second claim, consider $\mathcal{P}$ and $\mathcal{P}'$ equipped with the linear indices as in Definition 2.3.2, and suppose $\sigma \in \mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$. Then, by definition there exists $\xi \in (0,\infty)^{2n}$ satisfying

$$p_{\sigma(0)}(\xi) < p_{\sigma(1)}(\xi) < \cdots < p_{\sigma(2^n-1)}(\xi).$$

Let $\xi' = T(\xi)$ and apply the first result to successive pairs in the ordering which implies that for all $0 \leq k \leq 2^n - 2$, we have

$$p'_{\sigma(k)}(\xi') = \log\left(p_{\sigma(k)}(\xi)\right) < \log\left(p_{\sigma(k+1)}(\xi)\right) = p'_{\sigma(k+1)}(\xi').$$

Thus, we $\xi' \in \mathsf{R}^m$ satisfies

$$p'_{\sigma(0)}(\xi') < p'_{\sigma(1)}(\xi') < \cdots < p'_{\sigma(2^n-1)}(\xi'),$$

and it follows that $\sigma \in \mathcal{T}\left(\mathcal{P}', \prec_B, \Xi'\right)$ which completes the proof. □

Example 2.3.4. *Recall the PSD in Example 2.1.2 with interaction function* $f(z) = (z_1 + z_2)z_3$ *corresponding to interaction type* $\mathbf{n} = (2,1)$*. The polynomials for the linearized PSD problem are*

$$\begin{aligned} p'_0 &= x_{1,0} + x_{2,0} & p'_4 &= x_{1,1} + x_{2,0} \\ p'_1 &= x_{1,0} + x_{2,1} & p'_5 &= x_{1,1} + x_{2,1} \\ p'_2 &= x_{1,2} + x_{2,0} & p'_6 &= x_{1,3} + x_{2,0} \\ p'_3 &= x_{1,2} + x_{2,1} & p'_7 &= x_{1,3} + x_{2,1} \end{aligned}$$

where we have used linear indexing to match the polynomials in Example 2.1.2.

2.3.4 Solving the PSD problem for interaction type $\mathbf{n} = (2,1,\ldots,1)$.

In this section we prove the following theorem.

Theorem 2.3.5. *Let* f *be an interaction function with interaction type,* $\mathbf{n} = (2,1,\ldots,1)$*.*

Let $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ denote the corresponding PSD problem and $(\mathcal{P}', \prec'_B, \mathsf{R}^m)$ the associated linearized PSD problem. Then $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) = \mathcal{T}(\mathcal{P}', \prec_B, \Xi')$ where $\Xi' = \mathsf{R}^m \cap \{-\xi'_{1,0} + \xi'_{1,1} + \xi'_{1,2} - \xi'_{1,3} > 0\}$.

The proof of the theorem is based on the following lemma

Lemma 2.3.8. *Fix parameters, $x_0, x_1, x_2, x_3 \in \mathsf{R}$, and define the function, $g : \mathsf{R} \to \mathsf{R}$ by the formula*

$$g(t) = \exp(tx_0) - \exp(tx_1) - \exp(tx_2) + \exp(tx_3).$$

If $x_0 < x_1 \leq x_2 < x_3$, then g has a positive root if and only if $g'(0) < 0$.

Proof. Suppose first that t_0 is a root of g. Expanding $\exp(t_0 x_1)$ and $\exp(t_0 x_2)$ to first order about x_0 and x_3, respectively, and applying the mean value theorem yields the formula

$$g(t_0) = -t_0 \exp(t_0 c_1)(x_1 - x_0) - t_0 \exp(t_0 c_2)(x_2 - x_3) = 0 \tag{2.17}$$

for some $c_1 \in (x_0, x_1)$ and $c_2 \in (x_2, x_3)$. We define $k = c_2 - c_1$ and multiply Equation (2.17) by $t_0 e^{-kt_0}$ to obtain

$$e^{kt_0}(x_3 - x_2) - (x_1 - x_0) = 0.$$

Noting that $c_1 < x_2 < c_2$, it follows that $k > 0$. Therefore if $t_0 > 0$, then $x_3 - x_2 < x_1 - x_0$ or equivalently, $g'(0) = x_0 - x_1 - x_2 + x_3 < 0$.

Conversely, if $g'(0) < 0$ then g has at least one positive root since clearly $g(0) = 0$ and $\lim_{t\to\infty} g(t) = \infty$. □

Proof of Theorem 2.3.5. Suppose $\sigma \in \mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ and $\xi \in (0,\infty)^{2n}_\sigma$, then by

Theorem 2.3.3 we have $\xi' = T(\xi) \in \mathsf{R}^m_\sigma$. Note that by definition the first four coordinates of ξ' are given by the formulas

$$\begin{aligned}
\xi'_{1,0} &= \log(\xi_1 + \xi_2) \\
\xi'_{1,1} &= \log(\xi_1 + \xi_2 + \xi_{n+2}) \\
\xi'_{1,2} &= \log(\xi_1 + \xi_2 + \xi_{n+1}) \\
\xi'_{1,3} &= \log(\xi_1 + \xi_2 + \xi_{n+1} + \xi_{n+2}).
\end{aligned}$$

Since $\xi_i > 0$ for $i \in \{1, 2, n+1, n+2\}$, it follows that

$$-\xi'_{1,0} + \xi'_{1,1} + \xi'_{1,2} - \xi'_{1,3} > 0,$$

so we have $\sigma \in \mathcal{T}(\mathcal{P}', \prec_B, \Xi')$.

Conversely, suppose $\sigma \in \mathcal{T}(\mathcal{P}', \prec_B, \Xi')$ and $\xi' \in \Xi'_\sigma$. From the Boolean lattice $\prec_B$ we have $\xi'_{1,0} < \xi'_{1,1} \leq \xi'_{1,2} < \xi'_{1,3}$ or $\xi'_{1,0} < \xi'_{1,2} \leq \xi'_{1,1} < \xi'_{1,3}$. Moreover, ξ' also satisfies, $-\xi'_{1,0} + \xi'_{1,1} + \xi'_{1,2} - \xi'_{1,3} > 0$. Hence, Lemma 2.3.8 implies that there exists $t' > 0$ such that $\hat{\xi}' := t'\xi'$ satisfies

$$\exp(\hat{\xi}'_{1,0}) - \exp(\hat{\xi}'_{1,1}) - \exp(\hat{\xi}'_{1,2}) + \exp(\hat{\xi}'_{1,3}) = 0.$$

Next, we define $\hat{\xi} \in (0, \infty)^{2n}$ by

$$\hat{\xi}_j = \begin{cases}
\exp(\hat{\xi}'_{j,0}) & 2 < j \leq n \\
\exp(\hat{\xi}'_{j,1}) - \exp(\hat{\xi}'_{j,0}) & n+2 < j < 2n \\
\frac{1}{2}\exp(\hat{\xi}'_{1,0}) & j = 1, 2 \\
\exp(\hat{\xi}'_{1,2}) - \exp(\hat{\xi}'_{1,0}) & j = n+1 \\
\exp(\hat{\xi}'_{1,1}) - \exp(\hat{\xi}'_{1,0}) & j = n+2
\end{cases}$$

One easily verifies that $\hat{\xi}_j > 0$ for all $1 \leq j \leq 2n$, and that $T(\hat{\xi}) = \hat{\xi}'$. From Theorem 2.3.3, we have $\hat{\xi} \in (0, \infty)^{2n}_\sigma = \{\xi \in (0, \infty)^{2n} : p_{\sigma(0)}(\xi) < p_{\sigma(1)}(\xi) < \cdots < p_{\sigma(2^n-1)}(\xi)\}$

which implies that $\sigma \in \mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$. $\square$

2.3.5 Solving the general PSD problem

In the general case, we have $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) \subsetneq \mathcal{T}\left(\mathcal{P}', \prec_B, \Xi'\right)$, and thus, computing $\mathcal{T}\left(\mathcal{P}', \prec_B, \mathsf{R}^m\right)$ provides only a set of candidates for $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$. This candidate set contains spurious linear extensions so we consider the problem of removing linear extensions which are non-admissible. We have two strategies for doing this efficiently.

The first is to restrict the evaluation domain to a strict subset, $\Xi' \subsetneq \mathsf{R}^m$, such that we still have the inclusion

$$\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) \subseteq \mathcal{T}(\mathcal{P}', \prec_B, \Xi'). \tag{2.18}$$

Restricting to a smaller evaluation domain amounts to imposing more of the algebraic constraints a-priori which results in improved efficiency. In order for the candidate set on the right hand side to be efficiently computable using the algorithm in Section 2.2, it must be an instance of LC-LEP i.e. Ξ' should be a polytope. For example, for the PSD with interaction type $\mathbf{n} = (2, 1, \ldots, 1)$, analyzed in Section 2.3.4, we computed on the restricted domain

$$\Xi' = \mathsf{R}^m \cap \left\{\xi' \in \mathsf{R}^m : -\xi'_{1,0} + \xi'_{1,1} + \xi'_{1,2} - \xi'_{1,3} > 0\right\}.$$

In terms of the algorithm in Section 2.2, this domain restriction amounts to taking our base cone in Algorithm 3 to be $\mathrm{cone}(V_0)$ where

$$V_0 = V_{\prec_B} \cup \{\mathbf{u}\}$$

and $\mathbf{u}$ is the representation vector for the linear functional defined by the formula

$$x \mapsto -x_{1,0} + x_{1,1} + x_{1,2} - x_{1,3}.$$

The requirement that this linear functional must be strictly positive is a special case of the following Lemma whose proof is a trivial computation.

Lemma 2.3.9. *Suppose $\alpha, \alpha', \beta, \beta'$ are Boolean indices such that for any $\xi \in (0,\infty)^{2n}$, the following equations are satis ed.*

$$p_\alpha(\xi) < p_\beta(\xi) < p_{\beta'}(\xi) < p_{\alpha'}(\xi)$$
$$p_\alpha(\xi) + p_{\alpha'}(\xi) = p_\beta(\xi) + p_{\beta'}(\xi).$$

Then,

$$\log(p_\alpha(\xi)) + \log(p_{\alpha'}(\xi)) - \log(p_\beta(\xi)) - \log(p_{\beta'}(\xi)) > 0.$$

Lemma 2.3.9 provides a means to restrict the evaluation domain for the general linearized PSD problem as follows. Fix $j \in \{1,\dots,q\}$ and suppose $\{\alpha, \alpha', \beta, \beta'\} \subset E$ differ only in the j^{th} coordinate with $\alpha \prec_B \beta \prec_B \beta' \prec_B \alpha'$, and also assume that $B(\alpha) + B(\alpha') = B(\beta) + B(\beta)'$ where B is the Boolean indexing map. Then, it follows that for any $\xi \in \Xi$, the values, $\{p_\alpha(\xi), p_{\alpha'}(\xi), p_\beta(\xi), p_{\beta'}(\xi)\}$, satisfy both equations in Lemma 2.3.9. Therefore, if $\mathbf{u}(\{\alpha, \alpha', \beta, \beta'\})$ is the representation vector for the linear functional defined by

$$x \mapsto x_{j,B(\beta)} + x_{j,B(\beta')} - x_{j,B(\alpha)} - x_{j,B(\alpha')},$$

then $v(\{\alpha, \alpha', \beta, \beta'\})$ lies in V_σ for any $\sigma \in \mathcal{T}(\mathcal{P}, \prec_B, (0,\infty)^{2n}$. Equivalently, we may impose the required linear constraint, $x_{j,B(\beta)} + x_{j,B(\beta')} - x_{j,B(\alpha)} - x_{j,B(\alpha')} > 0$ on the evaluation domain of the linearized problem. Hence, for each $1 \le j \le q$, we define

$$V_j := \left\{\mathbf{u}(\{\alpha, \alpha', \beta, \beta'\}) : B(\alpha) + B(\alpha') = B(\beta) + B(\beta)',\ \alpha \prec_B \beta \prec_B \beta' \prec_B \alpha'\right\}$$

and for an arbitrary PSD problem, we may take our base cone to be

$$V_0 = V_{\prec_B} \cup V_\Xi \qquad \text{where} \quad V_\Xi = \bigcup_{j=1}^{q} V_j.$$

Applying Algorithm 3 with the base cone generated by V_0 is equivalent to solving the instance of LC-LEP defined by $(\mathcal{P}', \prec_B, \Xi')$ where Ξ' is the restriction of R^m to the subset for which the linear functionals defined by each $v \in V_j$ are strictly positive for each $1 \leq j \leq q$.

In addition to restricting the computation to the polytopes discussed above, we can reuse solutions of smaller PSD problems in some larger computations. As an example, suppose $\mathcal{P}' = \{p'_0, \ldots, p'_7\}$ is the set of interaction polynomials for the PSD with interaction type $\mathbf{n}' = (2, 1)$ and $\mathcal{P} := \{p_0, \ldots, p_{15}\}$ the polynomials for the PSD problem with interaction type $\mathbf{n} = (2, 1, 1)$. Observe that each admissible linear order on $\mathcal{P}'$ induces an imposed linear order on the even indexed polynomials, $\mathcal{P}_{\text{even}} := \{p_0, p_2, \ldots, p_{14}\} \subset \mathcal{P}$. A similar linear order is induced on the odd indexed polynomials, $\mathcal{P}_{\text{odd}} := \{p_1, p_3, \ldots, p_{15}\} \subset \mathcal{P}$. Hence, a necessary condition to have an admissible linear extension for $\mathcal{P}$ is that the order of $\mathcal{P}_{\text{even}}$ and $\mathcal{P}_{\text{odd}}$ must both be consistent with one of the PSD solutions in $\mathcal{T}\left(\mathcal{P}', \prec_B, (0, \infty)^6\right)$. This implies the inclusion

$$\mathcal{T}\left(\mathcal{P}, \prec_B, (0, \infty)^8\right) \subseteq \bigcup_{\sigma' \in \mathcal{T}(\mathcal{P}', \prec_B, (0,\infty)^6)} \mathcal{T}(\mathcal{P}, \prec_B \cup \prec_{\sigma'}, (0, \infty)^8) \tag{2.19}$$

where $\prec_B \cup \prec_{\sigma'}$ represents the refinement of the Boolean lattice partial order, and the partial order induced by σ' on the even/odd subsets.

To exploit this in general, we say that the PSD problem of type $\mathbf{n}'$ is a sub-problem for the PSD problem of type $\mathbf{n}$ whenever the polynomials for $\mathbf{n}$ must obey an implied partial order determined by the solutions of $\mathbf{n}'$. Notice that the preceding discussion as well as Equation (2.19) applies also to an arbitrary polytope. Therefore, if there are a total of k admissible linear extensions for all sub-problems of the PSD problem of type $\mathbf{n}$ which we have previously computed, then we bootstrap those results when computing $\mathcal{T}\left(\mathcal{P}, \prec_B, (0, \infty)^{2n}\right)$ via the inclusion

$$\mathcal{T}\left(\mathcal{P}, \prec_B, (0, \infty)^{2n}\right) \subseteq \bigcup_{i=1}^{k} \mathcal{T}(\mathcal{P}, \prec_B \cup \prec_{\sigma'_i}, \Xi') \subseteq \mathcal{T}(\mathcal{P}, \prec_B, \Xi')$$

where $\prec_B \cup \prec_{\sigma'_i}$ represents the refinement of the Boolean lattice partial order, and

the partial order induced by σ_i' on the corresponding subsets obtained from any subproblem. This technique has been used in the computation for all the cases of order ≥ 4. Observe that the computation of $\mathcal{T}(\mathcal{P}, \prec_B \cup \prec_{\sigma_i'}, \Xi')$ can be done distributively for $i = 1, \ldots, k$ on different computational nodes, which, as is indicated in Section 2.4, we employed for the PSD problems of orders 5 and 6.

In the special case of Section 2.3.4, we proved that inclusion in Equation (2.18) is actually equality when Ξ' is constructed as we have described. However, in the typical case, these additional algebraic constraints are not sufficient to remove all spurious linear extensions except in the case $\mathbf{n} = (2, 1, \ldots, 1)$. It remains an open problem to determine a smaller set Ξ' such that $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) = \mathcal{T}\left(\mathcal{P}', \prec_B, \Xi'\right)$ for other interaction types. However, in the remainder of this section we consider the problem of extracting $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ from $\mathcal{T}\left(\mathcal{P}', \prec_B, \Xi'\right)$ when they are not equal.

Observe that we may obtain large subsets of $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ simply by sampling. The particular strategy that we adopted is as follows. We uniformly sampled between 10^8 and 10^9 points

$$\xi = (l_1, \ldots, l_n, \delta_1, \ldots, \delta_n) \in \mathsf{Z}_+^{2n} \cap B_\infty^{2n}(r),$$

where $B_\infty^{2n}(r) = \{\|\xi\|_\infty \leq r\}$. We chose $r = 1000$. Mathematically the particular choice of r is not important since the PSD polynomials are homogeneous, though in practice it does have an effect on sampling precision and speed. For each such ξ we evaluated $\{p_\alpha(\xi) : p \in \mathcal{P}\}$. If $\sigma \in S_{2^n}$ denotes the linear order of these values, then ξ serves as a "witness" for the claim that $\Xi_\sigma \neq \emptyset$. This produces

$$\mathcal{S}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) := \left\{\sigma \in \mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) : \sigma \text{ is witnessed by at least one sample}\right\}.$$

Obviously,

$$\mathcal{S}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) \subseteq \mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) \subseteq \mathcal{T}(\mathcal{P}, \prec_B, \Xi').$$

In general, sampling is relatively efficient and in cases where $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ is not

too large (see Table 2.1 for details), we recover the entire solution.

Once we constructed the set $\mathcal{S}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right), \mathcal{T}(\mathcal{P}, \prec_B, \Xi')$ from sampling and algorithms in section 2.2 respectively, we can apply CAD algorithm to check the set

$$\{\xi \in \Xi : p_{\sigma(0)}(\xi) < p_{\sigma(1)}(\xi) < \cdots < p_{\sigma(2^n-1)}(\xi)\}$$

is empty or not for each $\sigma \in \mathcal{T}(\mathcal{P}, \prec_B, \Xi') \backslash \mathcal{S}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ and then $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$ is recovered. The CAD algorithm implementation we are using is CylindricalAlgebraicDecomposition in Mathematica 11 [20].

2.4 Results for some PSD problems

In this section we provide (see Table 2.1) the results of our computations for interaction functions of orders 4, 5, and 6. A slightly different approach was taken to compute orders 5 and 6, from that used for 4. This had to do with the machines being used, but highlights the flexibility of our method.

For interaction functions of order 4, we applied Algorithm 1 using a rational linear programming algorithm. In particular, we used the implementation MixedIntegerLinearProgram from SageMath 8 [29]. This implies that the output of Algorithm 3 is correct. Observe that interaction type $(1,1,1,1)$ is linear and type (4) is log linear, and therefore Algorithm 3 produces $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$. The fact that our output agrees with that of [21] suggests that our code is functioning as desired. To compute the interaction type $(2,1,1)$ we apply Algorithm 3 to obtain $\mathcal{T}\left(\mathcal{P}', \prec_B, \Xi'\right)$. By Theorem 2.3.5 this determines $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$.

To solve the PSD problem from interaction types $(2,2)$ and $(3,1)$ requires that we make use of the strategy discussed in Section 2.3.5. Again, we use Algorithm 3 to obtain $\mathcal{T}\left(\mathcal{P}', \prec_B, \Xi'\right)$. By Theorem 2.3.3, $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right) \subset \mathcal{T}\left(\mathcal{P}', \prec_B, \Xi'\right)$. As indicated in Column 7 of Table 2.1, we chose 10^8 samples from $(0,\infty)^8$ and identified 5344 and 3084 linear orders, respectively. We ran CylindricalAlgebraicDecomposition in Mathematica 11 [20] on each element of $\mathcal{T}\left(\mathcal{P}', \prec_B, \Xi'\right) \setminus \mathcal{S}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$. As can be seen by comparing Columns 6 and 3, none of these elements were admissible.

n	$\#\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$	$\#\mathcal{T}(\mathcal{P}', \prec_B, \Xi')$	$\#\mathcal{T}\left(\mathcal{P}', \prec_B, \Xi'\right)$	$\#\mathcal{S}(\mathcal{P}', \prec_B, (0,\infty)^{2n})$
(1,1,1,1)	336	-	-	-
(4)	336	-	-	-
(2,1,1)	1,344	1,344	2,352	-
(2,2)	5,344	7,920	26,640	5,344
(3,1)	3,084	5,112	68,641	3,084
(1,1,1,1,1)	61,920	-	-	61,920
(5)	61,920	-	-	61,920
(2,1,1,1)	790,200	790,200	*	790,200
(2,2,1)	-	11,035,808	*	6,570,952
(3,2)	-	*	*	$71{,}959{,}088^{\dagger}$
(4,1)	-	*	*	$11{,}213{,}616^{\dagger}$
(1,1,1,1,1,1)	89,414,640	-	-	89,414,640
(6)	89,414,640	-	-	89,414,640

Table 2.1: Computational results for several PSD problems. Column 1 indicates the interaction type. Column 2 provides the number of elements in the AC-LEP of interest. Column 3 provides the number of elements in an associated LC-LEP. This is not relevant where the AC-LEP problem of interest is a LC-LEP problem and is indicated by -. The * indicates that the computation was too large to complete. Column 4 provides the number of elements in the linearized PSD problem without additional constraints. Again the irrelevance for linear problems is indicated by - and * indicates that the computation is large to be performed. The last column indicates the number of cells identified via sampling. We used 10^8 samples for all $n = 4$ cases and 10^9 samples for the $n = 5, 6$ cases. The symbol † indicates that our sampling was not sufficient.

We now turn to the computations of interaction functions of order 5 and 6. As these problems are too big to be done on a laptop we turned to a server for which SageMath was not installed. Thus, we made use of a numerical linear programing algorithm, linprog from Python 3.5 package scipy [33] with the default numerical error 10^{-13}, in Algorithm 1. The interaction type $(1,1,1,1,1)$ and $(1,1,1,1,1,1)$ are linear and type (5) and (6) are log linear, and therefore via Algorithm 3 we obtain $\mathcal{T}_{alg}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$. We use the sampling technique (see Columns 7 and 8 of Table 2.1) to verify each of the elements of $\mathcal{T}_{alg}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$, thereby obtaining $\mathcal{T}\left(\mathcal{P}, \prec_B, (0,\infty)^{2n}\right)$.

The computation for each order 4 case was done on a Mac Pro laptop (2.7 Hz Intel i5 and memory 8GB) with computation time under 4 hours. The computation of the remaining cases were done using a computing server with CentOs, intel 17.1, memory 32 GB, and less than 30 nodes. The computation time for both $(1,1,1,1,1)$ and (5) was less than 4 hours, while the computation time for $(2,1,1,1)$ was on the order of 7

days. The codes which produced all of the computations in Table 2.1 are available on GitHub.

Chapter 3

Computing homology of regions of parameter space

3.1 Introduction

There are a wide variety of reasons why systems biology is a challenging subject, but from a purely mathematical perspective a key factor is that it is a problem in nonlinear dynamics involving many unknown parameters with at best a heuristic understanding of the nonlinearities. The focus of this Chapter is on identifying the topology of parameter space. In particular, given a particular local or global dynamical structure (in the language of biology one should think of phenotype) our goal is to develop a computational framework through which we can compute the homology of the subset of parameter space at which that dynamical structure occurs.

To put this into context consider the toggle switch shown in Figure 1.2. This a symbolic representation of a gene regulatory network where the protein produced by one gene represses the production of the protein of the other gene. As is discussed in greater detail below there are at least *eight* natural parameters for this system. As a function of these parameters the global dynamics of the toggle switch takes two forms: monostability and bistability. The machinery we develop allows us to conclude that bistability occurs over a contractible set of parameters, while the set of parameters that leads to monostability has the homology of a circle. Given the dimensions of parameter space this is a nontrivial statement.

Because of its simplicity, the toggle switch was one of the first examples of a synthetic regulatory network [30]. In general networks of interest either in systems or synthetic biology are more complicated and have much higher dimensional parameter spaces (see Figure 1.1). Thus, there are essentially two problems that need to be solved: identifying cell complexes with which to compute the homology of subsets of parameter space, and

computing homology of large high dimensional complexes. In this paper we focus on the first problem leaving analysis of networks of the type Figure 1.1 for future work.

Obviously, how particular parameters influence dynamics depends on the choice of model. Our results are based on a combinatorial order theoretic model for nonlinear dynamics as instantiated through the software package Dynamic Signatures Generated by Regulatory Networks (DSGRN) introduced in [15]. For examples of applications of this software to problems of biological interest we refer the reader to [15, 14, 18, 34]. Describing how DSGRN identifies a decomposition of parameter space from a regulatory networks is somewhat involved and thus postponed until Section 3.2.

For the moment, with admitted lack of motivation, we focus on the essential mathematical ideas. We fix two positive integers n and n_O. The parameters of interest are $\gamma \in \mathsf{R}_+ = (0, \infty)$, $\theta = (\theta_1, \ldots, \theta_{n_O}) \in \mathsf{R}_+^{n_O}$, $\ell = (\ell_1, \ldots, \ell_n) \in \mathsf{R}_+^n$, and $\delta = (\delta_1, \ldots, \delta_n) \in \mathsf{R}_+^n$. Expressed collectively the parameters are written as $(\gamma, \theta, \ell, \delta) \in \mathsf{R}_+^{1+n_O+2n}$. The decomposition of this parameter space is given in terms of semi-algebraic sets described via the following construction.

Definition 3.1.1. An *interaction function* of order n is a polynomial in n variables, $z = (z_1, \ldots, z_n)$, of the form

$$f(z) = \prod_{j=1}^{q} \left(\sum_{i \in I_j} z_i \right) \tag{3.1}$$

where the indexing sets $\{I_j : 1 \leq j \leq q\}$ form a partition for $\{1, \ldots, n\}$. The *interaction type* of f is denoted by $\mathbf{n} := (n_1, \ldots, n_q)$ where n_j indicates the number of elements in I_j. Since the order of the indexing sets I_j does not matter, for convenience we always assume that

$$n_1 \geq n_2 \geq \cdots \geq n_q.$$

The interaction type of f is used to define 2^n polynomials on the parameters ℓ and δ as follows. For a fixed interaction function of type $\mathbf{n}$ define $\mathcal{P}(\mathbf{n})$ to be the collection of polynomials obtained by evaluating $f(z)$ with each $z_k \in \{\ell_k, \ell_k + \delta_k\}$. In particular,

taking all possible combinations of z_k for $1 \leq k \leq n$ produces

$$\mathcal{P}(\mathbf{n}) = \{p_0, \ldots, p_{2^n-1}\} \subset \mathsf{R}[\ell_1, \ldots, \ell_n, \delta_1, \ldots, \delta_n]. \tag{3.2}$$

The set of polynomials that are used to define the regions in $\mathsf{R}_+^{1+n_O+2n}$ are

$$\mathcal{Q}(\mathbf{n}; n_O) := \{q := \gamma\theta_j - p : p \in \mathcal{P}(\mathbf{n}),\ j = 1, \ldots, n_O\} \tag{3.3}$$

with the additional assumption that $\theta_i \neq \theta_j$. Observe that if $q \in \mathcal{Q}(\mathbf{n}; n_O)$, then $q \in \mathsf{R}[\gamma, \theta, \ell, \delta]$. As explained in Section 3.2 the pair $(\mathbf{n}; n_O)$ is referred to as a *node type*.

To index the regions of interests we set $B(\mathbf{n}; n_O) = \prod_{q \in \mathcal{Q}} \{0, 1\}$. Given $b \in B(\mathbf{n}; n_O)$ we denote its q-th coordinate by $b(q)$. For any element $b \in B(\mathbf{n}; n_O)$, define a region of parameter space by

$$PN(b) = \left\{(\gamma, \theta, \ell, \delta) : (-1)^{b(q)} q(\gamma, \theta, \ell, \delta) < 0, q \in \mathcal{Q}\right\}. \tag{3.4}$$

A remark of practical importance is that for typical $b \in B(\mathbf{n}; n_O)$, $PN(b) = \emptyset$. Define

$$B^*(\mathbf{n}; n_O) := \{b \in B : PN(b) \neq \emptyset\}.$$

Identifying $B^*(\mathbf{n}; n_O)$ is computationally extremely challenging, however for the node types presented in Table 3.1 it has been determined (see [26]).

Given $b \in B^*(\mathbf{n}; n_O)$ define

$$\widetilde{\Upsilon}_{(\mathbf{n}; n_O)}(b) := \mathrm{cl}(PN(b)) \cap \mathsf{R}_+^{1+n_0+2n}. \tag{3.5}$$

For the problems of biological interest we need to add another level of complexity. Networks contain multiple nodes. So assume that we are given a collection of node

types $(\mathbf{n}_i; n_{O,i})$, $i = 1, \ldots, n$. Define

$$B^*_{nw} = \prod_{i=1}^{N} B^*(\mathbf{n}_i; n_{O,i}). \tag{3.6}$$

Given $b = (b_1, \ldots, b_N) \in B^*_{nw}$ define

$$\widetilde{\Upsilon}_{nw}(\{b\}) = \prod_{i=1}^{N} \widetilde{\Upsilon}_{(\mathbf{n}_i; n_{O,i})}(b_i)$$

and more generally, for any $B' \subset B^*$ set

$$\widetilde{\Upsilon}_{nw}(B') = \bigcup_{b \in B'} \widetilde{\Upsilon}_{nw}(\{b\})$$

Our long term goal is to be able to compute $H_*(\widetilde{\Upsilon}_{nw}(B'))$. We are far from this goal. However, this paper provides the primitives to compute homology using Z_2 coefficients for networks where the $(\mathbf{n}_i; n_{O,i})$ are taken from the following three types:

$$((n); n_O), \quad ((1, \ldots, 1); n_O), \quad \text{or} \quad ((2, 1, \ldots, 1); n_O). \tag{3.7}$$

While obviously we would like to be able do more, we remark that this is sufficient to consider a wide variety of biologically realistic networks including that of Figure 1.1.

To explain why it is difficult to compute $H_*(\widetilde{\Upsilon}_{nw}(B'))$, we note that the number of elements of $B^*(\mathbf{n}; n_O)$ grows rapidly as a function of n. This implies that computations are best done using a computer, which in turn suggests that we want to represent $\widetilde{\Upsilon}_{(\mathbf{n}; n_O)}(b)$ via a finite cellular complex. However, for any $b \in B^*(\mathbf{n}; n_O)$, $\widetilde{\Upsilon}_{(\mathbf{n}; n_O)}(b)$ is an unbounded subset of $\mathsf{R}^{1+n_0+2n}_+$ that is neither closed nor open.

The major contribution of this paper is the development of database and pipeline in section 3.9 that takes as input a type $(\mathbf{n}; n_O)$ of the form given in (3.7) and produces as output the *DSGRN* $(\mathbf{n}; n_O)$ *cell complex* $\mathcal{X}_{(\mathbf{n}; n_O)}$ and its homology group over Z_2. The reason the cell complex $\mathcal{X}_{(\mathbf{n}; n_O)}$ is of central importance is as follows. Given a network with nodes $(\mathbf{n}_i; n_{O,i})$, $i = 1, \ldots, N$ of types given by (3.7), $b \in B^*_{nw}$, and $B' \subset B^*_{nw}$ we

type	$\mathcal{X}^{(0)}$	$\mathcal{X}^{(1)}$	$\mathcal{X}^{(2)}$	$\mathcal{X}^{(3)}$	$\mathcal{X}^{(4)}$	$\mathcal{X}^{(5)}$	$\mathcal{X}^{(6)}$	$\mathcal{X}^{(7)}$	$\mathcal{X}^{(8)}$
((1);1)	6	8	3	-	-	-	-	-	-
((1,1);1)	18	48	54	29	6	-	-	-	-
((2);1)	18	48	54	29	6	-	-	-	-
((1,1,1);1)	57	243	484	551	366	132	20	-	-
((2,1);1)	66	272	524	581	378	134	20	-	-
((3);1)	57	243	484	551	366	132	20	-	-
((1,1,1,1);1)	306	1902	5700	10380	12224	9370	4520	1247	150
((2,1,1);1)	433	2593	7406	12842	14451	10634	4948	1323	155
((4);1)	306	1902	5700	10380	12224	9370	4520	1247	150

Table 3.1: Results from explicit DSGRN cell complex computations. Column 1 indicates the node type. The remaining columns indicate how many cells of each dimension are in $\mathcal{X}(\mathbf{n}; n_O)$. A dash means that there are no cells of that dimension. The explicit cell complexes can be accessed at [1].

can define finite cell complexes (see Section 3.7)

$$\mathcal{X}_{nw}(\{b\}) := \prod_{i=1}^{N} \mathcal{X}_{(\mathbf{n}_i; n_{O,i})}(b_i) \quad \text{and} \quad \mathcal{X}_{nw}(B') = \bigcup_{b \in B'} \mathcal{X}_{nw}(\{b\}) \tag{3.8}$$

with the property that

$$H_*(\mathcal{X}_{nw}(B'); \mathsf{Z}_2) \cong H_*(\widetilde{\Upsilon}_{nw}(B'), \mathsf{Z}_2). \tag{3.9}$$

Another contribution of this paper is a collection of completed computations of a few $\mathcal{X}(\mathbf{n}; n_O)$. Table 3.1 give a condensed list of results in the simplest setting; the complete complexes can be accessed here [1]. We include this table to provide the reader some guidance as to how rapidly the size of the cell complexes grow as a function of the interaction types.

In our haste to state the mathematical focus of this paper we have purposely minimized motivation for the problem and only briefly hinted at the difficulties. We attempt to remedy this in Section 3.2 where we briefly discuss the mathematical model behind the DSGRN computations and then use the toggle switch as a concrete example of how the complex $\mathcal{X}((1); 1)$ is derived. We then show how this complex can be used to compute homology groups of interest in two simple canonical examples from synthetic

biology: the toggle switch and the repressilator. This is meant to provide some intuition for the formal proofs presented in later sections, however the reader who is only interested in the mathematical aspects of this work should feel free to skip this section.

The definition of $PN(b)$ is given by (3.4), and hence $\widetilde{\Upsilon}(B')$ is well defined. In contrast no hint as to the form of $\mathcal{X}(\mathbf{n}; n_O)$ has been given in this introduction. The reason is that it is derived through a sequence of spaces that involve technical constructions. This is the content of Sections 3.3 to 3.7.

We conclude the paper in Section 3.8 with an application to an example that arises from the use of DSGRN to propose designs of a gene regulatory network that exhibits switch-like behavior over a large range of parameter values.

3.2 The Toggle Switch

As is indicated in the introduction this section provides motivation for the homology computations. We begin with a minimal introduction to the biological meaning of parameters and how parameter space is decomposed by DSGRN. We then turn to the toggle switch, Figure 1.2(b), to provide an explicit explanation of the associated DSGRN cell complex $\mathcal{X}((1); 1)$.

The DSGRN software requires two inputs. The first is a regulatory network, i.e. an annotated directed graph as indicated in Figure 1.2(b). Conceptually, we assign to each node j in the regulatory network a non-negative quantity x_j, e.g. amount of protein associated with gene j. The output of DSGRN provides information about the dynamics of x_j as a function of parameters that are introduced below.

The disappearance of x_j is due to decay, which is assumed to occur at a constant rate, $\gamma_j > 0$. The rate of production of x_j is dependent upon the second input to DSGRN: for each node j an interaction function $\Lambda_j(x)$ defined as follows.

The edges of a regulatory network take one of two annotations: $k \to j$ that indicates that node k up regulates node j, and $k \dashv j$ that indicates that node k down regulates

node j. This up and down regulation is represented via the functions

$$\lambda^{+}_{j,k}(x_k) := \begin{cases} \ell_{j,k} & \text{if } x_k < \theta_{j,k} \\ \ell_{j,k} + \delta_{j,k} & \text{if } x_k > \theta_{j,k} \end{cases} \tag{3.10}$$

and

$$\lambda^{-}_{j,k}(x_k) := \begin{cases} \ell_{j,k} + \delta_{j,k} & \text{if } x_k < \theta_{j,k} \\ \ell_{j,k} & \text{if } x_k > \theta_{j,k}, \end{cases} \tag{3.11}$$

respectively, where the parameters $\theta_{j,k}$, $\ell_{j,k}$, and $\delta_{j,k}$ are assumed to be positive. This is not an idle assumption: θ is a threshold that affects change, and δ indicates the quantity of change. If these two parameters are allowed to be zero, then the edges in the regulatory network suggest influence where none exists.

For simplicity of expression assume that the incoming edges to node j are from nodes $\{1, \ldots, n_I\}$. To define Λ_j choose $\{I_1, \ldots, I_q\}$, a partition of $\{1, \ldots, n_I\}$, and set

$$\Lambda_j(x) := \prod_{i=1}^{q} \left(\sum_{j \in I_i} \lambda^{*}_{j,k}(x_k) \right)$$

where $* \in \{\pm\}$ is determined by the type of edge. Observe that $\Lambda_j(x)$ has the form given by Definition 3.1.1 and thus has a well defined interaction type $\mathbf{n}$. Furthermore, note that evaluating $\Lambda_j(x)$ over all possible combinations of $\ell_{j,k}$ and $\ell_{j,k} + \delta_{j,k}$ produces 2^{n_I} polynomials $\mathcal{P}(\mathbf{n})$ as in (3.2). If node j has n_O out edges then it is of node type $(\mathbf{n}; n_O)$.

At this point we have introduced all the parameters and thus the parameter space is R_{+}^{N+3E} where N is the number of nodes and E the number of edges in the regulatory network. We now turn to a representation of the parameters in the spirit of that presented in the introduction. A parameter for the full regulatory network takes the form $(\gamma, \theta, \ell, \delta) \in \mathsf{R}_{+}^{N+3E}$ where $\gamma = (\gamma_1, \ldots, \gamma_N)$, $\theta = (\theta_1, \ldots, \theta_N)$, $\ell = (\ell_1, \ldots, \ell_N)$, and $\delta = (\delta_1, \ldots, \delta_N)$. Furthermore, $\theta_j = (\theta_{k_1,j}, \ldots, \theta_{k_{n_O},j})$ where there is an out edge from node j to node k_i, $i = 1, \ldots, n_O$, ($n_O = n_O(j)$ is the number of out edges from node j), and $\ell_j = (\ell_{j,k_1}, \ldots, \ell_{j,n_I})$ and $\delta_j = (\delta_{j,k_1}, \ldots, \delta_{j,n_I})$ where there is an in edge

from node k_i, $i = 1, \ldots, n_I$, ($n_I = n_I(j)$ is the number of in edges to node j) to node j.

Observe that, as in the introduction, we have assigned $1 + n_O + 2n_I$ parameters to each node x_j in the regulatory network, the decay γ_j, the n_O thresholds $\theta_{i,j}$ associated with the n_O out-edges, and the $2n_I$ production rates $\ell_{j,k}$ and $\delta_{j,k}$ associated with the n_I in edges. Furthermore, if we let $P_j = \{(\gamma_j, \theta_j, \ell_j, \theta_j)\}$, then the full parameter space for the regulatory network is

$$P = \prod_{j=1}^{N} P_j = \mathsf{R}_+^{N+3E}. \tag{3.12}$$

DSGRN models dynamics on a crude level, monitoring only the increase and decrease of x_j. To be more precise, observe that on the regions defined by the complements of the hyperplanes $\{x_j = \theta_{k,j}\}$, Λ_j is constant and takes on a value determined by the parameters $\lambda_{j,k}$ and $\lambda_{j,k} + \delta_{j,k}$. Exploiting this observation DSGRN decomposes parameter space for the full regulatory network into regions where

$$\begin{aligned} 0 &\neq -\gamma_1\theta_{k,1} + \Lambda_1(x) \\ &\vdots \\ 0 &\neq -\gamma_N\theta_{k,N} + \Lambda_N(x) \end{aligned} \tag{3.13}$$

and we are considering all $\theta_{k,j}$ for each $j = 1, \ldots, N$. An observation of fundamental importance for this paper is that this decomposition of parameter space takes the form of a product. In particular, if for each $j = 1, \ldots, N$ we determine the regions of parameter space restricted to γ_j, θ_j, ℓ_j, and δ_j determined by the inequalities

$$0 \neq -\gamma_j\theta_{k,j} + \Lambda_j(x), \tag{3.14}$$

then a region of (3.13) can be written as a product over j of regions determined by (3.14). Observe that we have recovered $\mathcal{Q}(\mathbf{n}; n_O)$ as in (3.3), and thus if node j is of type $(\mathbf{n}; n_O)$, then the regions defined by (3.14) are exactly those indexed by $B^*(\mathbf{n}; n_O)$.

The simplest example that provides a concrete perspective to these general remarks is the toggle switch shown in Figure 1.2(b). Each node is of type $((1); 1)$ and the

equations corresponding to (3.13) are

$$0 \neq -\gamma_1\theta_{2,1} + \begin{cases} \ell_{1,2} + \delta_{1,2} & \text{if } x_2 < \theta_{1,2} \\ \ell_{1,2} & \text{if } x_2 > \theta_{1,2} \end{cases} \tag{3.15}$$

$$0 \neq -\gamma_2\theta_{1,2} + \begin{cases} \ell_{2,1} + \delta_{2,1} & \text{if } x_1 < \theta_{2,1} \\ \ell_{2,1} & \text{if } x_1 > \theta_{2,1}. \end{cases} \tag{3.16}$$

As discussed above, since the parameters in (3.15) and (3.16) are independent, the decomposition of parameter space takes the form of a product of the regions determined by (3.15) and the regions determined by (3.16). With this in mind we focus our discussion on (3.15).

The first step is to recognize that

$$\mathcal{P}((1);1) = \{\ell_{1,2}, \ell_{1,2} + \delta_{1,2}\} \quad \text{and} \quad \mathcal{Q}((1);1) = \{\gamma_1\theta_{2,1} - p : p \in \mathcal{P}(\mathbf{1};1)\}.$$

Thus, $B((1);1) = \{0,1\}^2$ and $B^*((1);1) = \{0,1\}^2 \setminus \{(0,1)\}$, where the latter is equivalent to the observation that the nonempty regions of parameter space satisfying (3.15) are defined by the inequalities

$$\gamma_1\theta_{2,1} < l_{1,2} < l_{1,2}+\delta_{1,2},\ l_{1,2} < \gamma_1\theta_{2,1} < l_{1,2}+\delta_{1,2}, \text{ and } l_{1,2} < l_{1,2}+\delta_{1,2} < \gamma_1\theta_{2,1}. \tag{3.17}$$

As in (3.5), given $b \in B^*((1);1)$ define

$$\widetilde{\Upsilon}_{((1);1)}(b) := \mathrm{cl}(PN(b)) \cap (0,\infty)^4.$$

As follows from the discussion in the introduction our objective is to compute $H_*(\widetilde{\Upsilon}_{nw}(B');\mathsf{Z}_2)$ for $B' \subset B^*_{nw} = B^*((1);1) \times B^*((1);1)$. However, as is left to the reader to check, $\widetilde{\Upsilon}_{((1);1)}(b)$ is neither closed (as a subset of R^4) nor bounded, and hence is not a finite CW complex. Thus to accomplish our objective we construct an appropriate finite cellular representation for $\widetilde{\Upsilon}_{nw}(B')$.

As a first step we identify a bounded region that is a strong deformation retract of

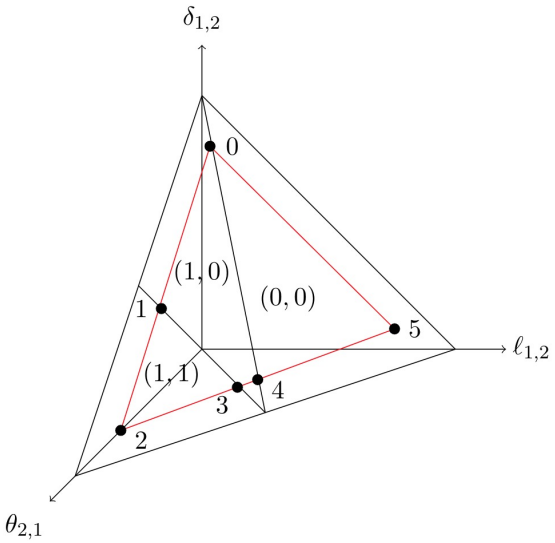


Figure 3.1: Derivation of the DSGRN cellular complex $\mathcal{X}_{((1);1),\epsilon}$. The large black simplex is obtained from the constraint $\theta_{2,1}+\ell_{1,2}+\delta_{1,2}=1$. The three regions defined by (3.17) are indexed by $(0,0),(1,0),(1,1) \in B^*((1);1)$. The red lines are defined by $\theta_{2,1}=\epsilon$, $\ell_{1,2}=\epsilon$, and $\delta_{1,2}=\epsilon$. The vertices of $\mathcal{X}_{((1);1),\epsilon}$ are labeled $0,\ldots,5$.

$\widetilde{\Upsilon}_{((1);1)}(B^*((1);(1))$. Define

$$Z := \{(1,\theta_{2,1},\ell_{1,2},\delta_{1,2}) : \theta_{2,1}+\ell_{1,2}+\delta_{1,2}=1, \theta_{2,1}>0, \ell_{1,2}>0, \delta_{1,2}>0\}\,. \tag{3.18}$$

Let $B' \subset B^*((1);1)$ and define

$$\Upsilon_{((1);1)}(B') := \widetilde{\Upsilon}_{((1);1)}(B') \cap Z.$$

It follows from Proposition 3.3.1 that there is a strong deformation retraction of $(0,\infty)^4$ onto Z, and furthermore, under this strong deformation retraction $\Upsilon_{((1);1)}(B')$ is a strong deformation retract of $\widetilde{\Upsilon}_{((1);1)}(B')$ for any $B' \subset B^*((1);1)$. The images of $\Upsilon_{((1);1)}(b_i)$ for $B^*((1);1) = \{b_1,b_2,b_3\}$ are shown in Figure 3.1. While $\Upsilon_{((1);1)}(B')$ is bounded it is not closed, and therefore we still need to identify an appropriate finite CW complex.

As is discussed above the assumptions that the parameters θ, ℓ, and δ are positive is driven by biological considerations. Observe, from Figure 3.1 that $\Upsilon_{((1);1)}(b_1) \cap \Upsilon_{((1);1)}(b_3) = \emptyset$. This fundamental topological property would change if we made use

of the obvious simplicial complex obtained by considering the cells $\mathrm{cl}(\widetilde{\Upsilon}_{((1);1)}(b_i))$, for $i = 1, 2, 3$. Therefore we take the opposite approach and add the restriction

$$\theta_{2,1} \geq \epsilon, \quad \ell_{1,2} \geq \epsilon, \quad \text{and} \quad \delta_{1,2} \geq \epsilon.$$

This produces $\mathcal{X}_{((1);1),\epsilon}$, the geometric simplicial complex shown in Figure 3.1. More precisely, letting $\mathcal{X}^{(k)}_{((1);1),\epsilon}$ denote the k-dimensional cells, the associated combinatorial cellular complex is

$$\begin{aligned}
\mathcal{X}^2_{((1);1),\epsilon} &= \left\{\xi^2_1 = \{0,4,5\}, \xi^2_2 = \{0,1,3,4\}, \xi^2_3 = \{1,2,3\}\right\} \\
\mathcal{X}^1_{((1);1),\epsilon} &= \{\xi^1_1 = \{0,5\}, \xi^1_2 = \{4,5\}, \xi^1_3 = \{0,4\}, \xi^1_4 = \{0,1\}, \xi^1_5 = \{1,3\}, \xi_6 = \{3,4\}, \\
&\qquad \xi^1_7 = \{1,2\}, \xi^1_8 = \{2,3\}\} \\
\mathcal{X}^0_{((1);1),\epsilon} &= \left\{\xi^0_1 = \{0\}, \xi^0_2 = \{1\}, \xi^0_3 = \{2\}, \xi^0_4 = \{3\}, \xi^0_5 = \{4\}, \xi^0_6 = \{5\}\right\}
\end{aligned}$$

where the incidence number κ is given by

$$\kappa(\xi^i_j, \xi^k_l) = \begin{cases} 1 & \text{if } i = k+1 \text{ and } \xi^k_l \subset \xi^i_j \\ 0 & \text{otherwise} \end{cases}$$

since we restrict our attention in this paper to the number field Z_2. In a slight abuse of notation, we use ξ^i_j to denote either a geometric simplex or its associated combinatorial cell.

As can be seen from Figure 3.1 there is considerable freedom in the choice of ϵ. However, if ϵ is chosen too large, then the homology computed using the complex $\mathcal{X}_{((1);1),\epsilon}$ will differ from the homology associated with Υ. Observe that the inequations used to define cell $\xi^2_1 = \{0,4,5\}$ associated with $b = (0,0) \in B^*$ are as follows

$$\begin{aligned}
\theta_{2,1} + \ell_{1,2} + \delta_{1,2} - 1 &= 0 \\
-\theta_{2,1} + \ell_{1,2} + \delta_{1,2} &\leq 0 \\
\epsilon - \theta_{2,1} \leq 0,\ \epsilon - \ell_{1,2} \leq 0,\ \epsilon - \delta_{1,2} &\leq 0 \\
-\epsilon &< 0.
\end{aligned} \tag{3.19}$$

More generally, we can write this as an inequation system

$$\{A_0 x = 0, B_0(b)x \leq 0, C_0 < 0\}$$

where $x = (\theta_{2,1}, \ell_{1,2}, \delta_{1,2}, \epsilon)$, A_0 is a 1×4 matrix that encodes the restriction to Z_1, $B_0(b)$ is a 4×4 matrix that explicitly depends on b, and C_0 encodes the restriction that $\epsilon > 0$. Because $b \in B^*$ we know that the inequation system has a solution for $\epsilon = 0$. However, we need an explicit positive value of ϵ that we can work with.

Using Fourier-Motzkin elimination (see Section 3.6) we can solve for $\epsilon(b) > 0$, the maximal values such all these inequations are satisfied. We leave it to the reader to check that

$$\frac{1}{4} = \epsilon((1);1) := \min_{b \in B_1^*} \epsilon(b). \tag{3.20}$$

Fourier-Motzkin elimination is an iterative method, and in particular, at each step one of the variables is eliminated. Since the entries of A_0 and $B_0(b)$ are $\{0, \pm 1\}$, the absolute values of the coefficients are bounded by 2^n. For this particular example, since after three steps we have reduced the inequation system to the single variable ϵ, we can conclude that $\epsilon(b) \geq \frac{1}{8}$, which is consistent with (3.20).

In Section 3.6 we prove that if $\epsilon_0, \epsilon_1 \in (0, \epsilon((1);1))$, then $\mathcal{X}_{((1);1),\epsilon_0}$ and $\mathcal{X}_{((1);1),\epsilon_1}$ are isomorphic cell complexes. We refer to any choice of $\mathcal{X}(\epsilon)$, $0 < \epsilon < \epsilon((1);1)$, as the *DSGRN* $((1);1)$ *cell-complex*. Since the cellular structure is determined we simplify the notation and write $\mathcal{X}((1);1)$. Since $\mathcal{X}((1);1)$ is a finite complex we can use it to compute homology.

We now consider the raison d'être for this paper, identifying the homology of parameter regions that lead to a particular phenotype. For this we need to consider the full parameter space associated with the regulatory network. Making use of the notation of (3.8), given $B' \subset B^*_{nw}$ we can compute

$$H_*(\mathcal{X}_{nw}(B'); \mathsf{Z}_2) \cong H_*(\widetilde{\Upsilon}_{nw}(B'); \mathsf{Z}_2).$$

Though an explanation of the details is beyond the scope of this paper (see [15])

for each $b \in B^*_{nw}$, DSGRN computes a Morse graph that provides a combinatorial description of the global dynamics that is valid for all parameters associate with b. In the case of the toggle switch $B^*_{nw} = B^*((1);1) \times B^*((1);1)$ consists of nine elements. As is shown in [18, Figure 2(d)] for one of the elements $\bar{b}$ the associated Morse graph indicates the occurrence of bistability and for the remaining eight elements, $B' = B^* \setminus \{\bar{b}\}$, the associated Morse graph indicates monostability.

Applying standard homology software [29] we obtain

$$H_k(\mathcal{X}(\bar{b}); \mathsf{Z}_2) \cong \begin{cases} \mathsf{Z}_2 & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.21}$$

and

$$H_k(\mathcal{X}(B'); \mathsf{Z}_2) \cong \begin{cases} \mathsf{Z}_2 & \text{if } k = 0, 1 \\ 0 & \text{otherwise.} \end{cases} \tag{3.22}$$

Admittedly in the case of the toggle switch the result of the first computation is not surprising. Since the node type is linear, the region associated with $\bar{b}$ is a convex (unbounded) polygon, and thus is contractible.

As a final comment it is worth noting that application of the specific construction presented of this section is not restricted to the toggle switch. The other fundamental example of synthetic biology is the repressilator [16] which is represented using the regulatory network in Figure 3.2. Observe that each node is of type $((1);1)$ and therefore the full parameter space R^{12}_+ is decomposed into 27 elements indexed by $B^*_{nw} = \prod_{j=1}^{3} B^*((1)_j; 1_j)$. As is indicated in [15, Figure 5] there is a single element $\hat{b} \in B^*_{nw}$ at which stable oscillations can occur and for the remaining regions $\hat{B} = B^*_{nw} \setminus \left\{\hat{b}\right\}$ we have monostability without oscillations observable on the scale of this DSGRN model. Direct computations show that

$$H_k(\mathcal{X}(\hat{b}); \mathsf{Z}_2) \cong \begin{cases} \mathsf{Z}_2 & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.23}$$

and

$$H_k(\mathcal{X}(\hat{B}); \mathsf{Z}_2) \cong \begin{cases} \mathsf{Z}_2 & \text{if } k = 0, 2 \\ 0 & \text{otherwise.} \end{cases} \tag{3.24}$$

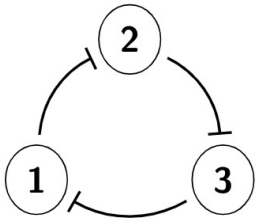


Figure 3.2: Regulatory network for the repressilator.

3.3 Identifying $\widetilde{\Upsilon}$ and Υ

Throughout this section we assume we are given a node with n in edges and n_O out edges. In this section we repeat, at least in spirit, the first part of the discussion of Section 3.2 where we derived a linear space Z (see (3.18)) that contains the desired CW complex. In fact, for the case of nodes with linear interaction type, i.e. $((n); n_O)$, the argument is essentially identical. In the case of nodes with a purely product interaction, i.e. $((1, \ldots, 1); n_O)$, we apply the logarithm to obtain a linear interaction type. Thus, the arguments for $((n); n_O)$ and $((1, \ldots, 1); n_O)$ go hand in hand. Not surprisingly dealing with a node of type $(2, 1, \ldots, 1; n_O)$ is more challenging and needs to be dealt with separately.

3.3.1 Node of type $((1, \ldots, 1); n_O)$

The following result and Theorem 3.7.1 are used to show that with respect to the study of the associated homology we can treat a node of type $(\mathbf{n}; n_O) = ((1, \ldots, 1); n_O)$ as a node of type $((n); n_O)$. We recall from [26] that $B^*((1, \ldots, 1); n_O) = B^*((n); n_O)$.

Theorem 3.3.1. *Let $B' \subset B^*((n); n_O) = B^*((1, \ldots, 1); n_O)$. Denote $\widetilde{\Upsilon}_1(B') = \widetilde{\Upsilon}_{((n);n_O)}(B')$ and $\widetilde{\Upsilon}_2(B') = \widetilde{\Upsilon}_{((1,\ldots,1);n_O)}(B')$ Then $\widetilde{\Upsilon}_1(B')$ and $\widetilde{\Upsilon}_2(B')$ are homotopic equivalent.*

Proof. We show $\widetilde{\Upsilon}_1(B')$ and $\widetilde{\Upsilon}_2(B')$ are homotopic equivalent for any given $B' \subset B$ through a series of strong deformation retractions. Recall that for both cases

$$(\gamma, \theta, \ell, \delta) \in \widetilde{\Upsilon}_1(B^*((n); n_O)) = \widetilde{\Upsilon}_2(B^*((1, \dots, 1); n_O)) = \mathsf{R}_+^{1+n_O+2n}.$$

Define $F\colon \mathsf{R}_+^{1+n_O+2n} \times [0,1] \to \mathsf{R}_+^{1+n_O+2n}$ by

$$F(\gamma, \theta, \ell, \delta, t) = \left(\frac{\gamma}{(1-t)+t\gamma}, ((1-t)+t\gamma)\theta, \ell, \delta \right). \tag{3.25}$$

We use F to define two collections of strong deformation retractions. Set

$$S_1(B') = F(\widetilde{\Upsilon}_1(B'), 1) \quad \text{and} \quad S_2(B') = F(\widetilde{\Upsilon}_2(B'), 1)$$

It is left to the reader to check that for each $B' \subset B^*((n); n_O) = B^*((1, \dots, 1); n_O)$, $F|_{\widetilde{\Upsilon}_1(B')}$ and $F|_{\widetilde{\Upsilon}_2(B')}$ is a strong deformation retraction. Observe that $S_1(B^*) \subset \{1\} \times \mathsf{R}_+^{n_O+2n}$ and $S_2(B^*) \subset \{1\} \times \mathsf{R}_+^{n_O+2n}$.

Define $G_1\colon \mathsf{R}_+^{1+n_O+2n} \times [0,1] \to \mathsf{R}_+^{1+n_O+2n}$ by

$$G_1(\gamma, \theta, \ell, \delta, t) = (\gamma, \lambda(\theta, \ell, \delta, t)^n \theta, \lambda(\theta, \ell, \delta, t)\ell, \lambda(\theta, \ell, \delta, t)\delta) \tag{3.26}$$

where

$$\lambda(\theta, \ell, \delta, t) = (1-t) + t \max_{k,j} \left\{ \frac{e}{\ell_k}, \frac{e}{\delta_k}, \left(\frac{e}{\theta_j} \right)^{\frac{1}{n}} \right\}.$$

Set $S_2(B') = G_1(S_1(B'), 1)$. Observe

$$S_2(B') = S_1(B') \cap \{(\gamma, \theta, \ell, \delta) \in \mathsf{R}_+^{1+n_O+2n} : \min_{k,j}\{\ell_k, \delta_k, \theta_j\} = e\}.$$

Then, $G_1|_{S_1(B')}$ provides a strong deformation retract of $S_1(B')$ onto $S_2(B')$.

Define $G_2\colon \mathsf{R}_+^{1+n_O+2n} \times [0,1] \to \mathsf{R}_+^{1+n_O+2n}$ by

$$G_2(\gamma, \theta, \ell, \delta, t) = (\gamma, n\lambda(\theta, \ell, \delta, t) + \theta, \lambda(\theta, \ell, \delta, t) + \ell, \lambda(\theta, \ell, \delta, t) + \delta) \tag{3.27}$$

where

$$\lambda(\theta,\ell,\delta,t) = t\max_{k,j}\left\{1-\ell_k, 1-\delta_k, \frac{1-\theta_j}{n}\right\}.$$

Set $S_4(B') = G_2(S_3(B'),1)$. Observe

$$S_4(B') = S_3(B') \cap \{(\gamma,\theta,\ell,\delta) \in \mathsf{R}_+^{1+n_O+2n} : \min_{k,j}\{\ell_k,\delta_k,\theta_j\} = 1\}$$

Then, G_2 provides a strong deformation retract of $S_3(B')$ onto $S_4(B')$.

Finally, define the map $\log : \mathsf{R}_+^{1+n_O+2n} \to \mathsf{R}_+^{1+n_O+2n}$

$$\log(\gamma,\theta,\ell,\delta) = (\log(\gamma),\log(\theta),\log(\ell),\log(\ell+\delta)-\log(\ell)) \tag{3.28}$$

where $\log(\theta) = (\log(\theta_1),\ldots,\log(\theta_{n_O}))$, $\log(l) = (\log(l_1),\ldots,\log(l_n))$ and $\log(l+\delta) - \log(l) = (\log(l_1+\delta_1)-\log(l_1),\ldots,\log(l_n+\delta_n)-\log(l_n))$. We claim $\log|_{S_2(B')}$ is from $S_2(B')$ to $S_4(B')$ for any given B'. The details are left to the reader. From this we have $\widetilde{\Upsilon}_1(B')$ and $\widetilde{\Upsilon}_2(B')$ are homotopy equivalent. □

3.3.2 Node of type $((n);n_O)$

Define

$$Z_{((n);n_O)} = \left\{(\gamma,\theta,\ell,\delta) : \gamma = 1, \sum_{j=1}^{n_O}\theta_j + \sum_{k=1}^{n}(2\ell_k+\delta_k) = 1\right\} \tag{3.29}$$

and for $B' \subset B^*((n);n_O)$ define

$$\Upsilon_{((n);n_O)}(B') := \widetilde{\Upsilon}_{((n);n_O)}(B') \cap Z_{((n);n_O)}. \tag{3.30}$$

Proposition 3.3.1. *Let $B' \subset B^*((n);n_O)$. Then $\Upsilon_{((n);n_O)}(B')$ is a strong deformation retraction of $\widetilde{\Upsilon}_{((n);n_O)}(B')$.*

Proof. Define $F\colon \mathsf{R}_+^{1+n_O+2n} \times [0,1] \to \mathsf{R}_+^{1+n_O+2n}$ by

$$F(\gamma,\theta,\ell,\delta,t) = \left(\frac{\gamma}{(1-t)+t\gamma}, \frac{(1-t)+t\gamma}{\lambda(\theta,\ell,\delta,t)}\theta, \frac{1}{\lambda(\theta,\ell,\delta,t)}\ell, \frac{1}{\lambda(\theta,\ell,\delta,t)}\delta\right) \tag{3.31}$$

where

$$\lambda(\theta,\ell,\delta,t) = (1-t) + t\left(\sum_{j=1}^{n_O}\theta_j + \sum_{k=1}^{n}(2\ell_k+\delta_k)\right).$$

We leave it to the reader to check that for $Z_{((n);n_O)}$ given by (3.29)

$$F(\cdot,t)|_{Z_{((n);n_O)}} = \mathrm{id}_{Z_{((n);n_O)}},$$
$$F(\mathsf{R}_+^{1+n_O+2n},0) = \mathrm{id}_{\mathsf{R}_+^{1+n_O+2n}},$$
$$F(\mathsf{R}_+^{1+n_O+2n},1) \subset Z_{((n);n_O)},$$

and

$$q(F(\gamma,\theta,\ell,\delta,t)) = \frac{q(\gamma,\theta,\ell,\delta)}{(1-t)+t(\sum_{j=1}^{n_O}\theta_j+\sum_{k=1}^{n}(2\ell_k+\delta_k))}$$

for any $q \in \mathcal{Q}((n(;n_O)$. Observe that the latter equality guarantees that F restricts to a strong deformation retract of $\widetilde{\Upsilon}_{((n);n_O)}(B')$ to $\Upsilon_{((n);n_O)}(B')$. □

3.3.3 Node of type $((2,1,\ldots,1);n_O)$

. Define

$$h(\theta,\ell,\delta) = \prod_{j=1}^{n_O}\theta_j\cdot\left(\sum_{i=1}^{2}\ell_i\right)\cdot\left(\sum_{i=1}^{2}\ell_i+\delta_1\right)\cdot\left(\sum_{i=1}^{2}\ell_i+\delta_2\right)\cdot\left(\sum_{i=1}^{2}\ell_i+\delta_1+\delta_2\right)\prod_{k=3}^{n}\ell_k\cdot\prod_{k=3}^{n}(\ell_k+\delta_k).$$

Define

$$Z_{((2,1,\ldots,1);n_O)} = \{(\gamma,\theta,\ell,\delta) : \gamma = 1, \ell_1 = \ell_2, h(\theta,\ell,\delta) = 1\} \tag{3.32}$$

and for $B' \subset B^*((2,1,\ldots,1);n_O)$ define

$$\Upsilon_{((2,1,\ldots,1);n_O)} := \widetilde{\Upsilon}_{((2,1,\ldots,1);n_O)} \cap Z_{((2,1,\ldots,1);n_O)}.$$

Proposition 3.3.2. *Let $B' \subset B^*((2,1,\ldots,1);n_O)$. Then $\Upsilon_{((2,1,\ldots,1);n_O)}(B')$ is a strong deformation retraction of $\widetilde{\Upsilon}_{((2,1,\ldots,1);n_O)}(B')$.*

Proof. Define $F\colon \mathsf{R}_+^{1+n_O+2n} \times [0,1] \to \mathsf{R}_+^{1+n_O+2n}$ by

$$F(\gamma,\theta,\ell,\delta,t) = \left(\frac{\gamma}{(1-t)+t\gamma}, \frac{(1-t)+t\gamma}{\lambda(\theta,\ell,\delta,t)^{n-1}}\theta, \frac{1}{\lambda(\theta,\ell,\delta,t)}\ell, \frac{1}{\lambda(\theta,\ell,\delta,t)}\delta\right)$$

where

$$\lambda(\theta,\ell,\delta,t) = ((1-t)+t\cdot h(\theta,\ell,\delta))^{\frac{1}{n_O(n-1)+2n}}.$$

Observe

$$F(\mathsf{R}_+^{1+n_O+2n},1) \subset \left\{(\gamma,\theta,\ell,\delta) \in \mathsf{R}_+^{1+n_O+2n} : \gamma = 1, h(\theta,\ell,\delta) = 1\right\}$$

Define $G\colon \mathsf{R}_+^{1+n_O+2n} \times [0,1] \to \mathsf{R}_+^{1+n_O+2n}$ by

$$G(\gamma,\theta,\ell,\delta,s) = \left(\gamma,\theta,(1-s)\ell_1 + s\frac{\ell_1+\ell_2}{2}, (1-s)\ell_2 + s\frac{\ell_1+\ell_2}{2}, \ell_3,\dots,\ell_n,\delta\right)$$

Observe

$$G(F(\mathsf{R}_+^{1+n_O+2n},1),1) = F(\mathsf{R}_+^{1+n_O+2n},1) \cap \left\{(\gamma,\theta,\ell,\delta) \in \mathsf{R}_+^{1+n_O+2n} : \ell_1 = \ell_2\right\} \subset Z_{((2,1,\dots,1);n_O)}$$

We leave it to the reader to check that for $Z_{((2,1,\dots,1);n_O)}$ given by (3.32)

$$G(F(\cdot,t),s)|_{Z_{((2,1,\dots,1);n_O)}} = \mathrm{id}_{Z_{((2,1,\dots,1);n_O)}}(\cdot),$$

and

$$q(F(\gamma,\theta,\ell,\delta,t)) = \frac{q(\gamma,\theta,\ell,\delta)}{((1-t)+t\cdot h(\theta,\ell,\delta))^{\frac{n-1}{n_O(n-1)+2n}}}.$$

Observe that the latter equality guarantees that

$$H(\cdot,t) = \begin{cases} G(F(\cdot,2t),0), & t \in [0,\frac{1}{2}] \\ G(F(\cdot,1),2t-1) & t \in [\frac{1}{2},1] \end{cases} \tag{3.33}$$

restricts to a strong deformation retract of $\widetilde{\Upsilon}_{((2,1,\dots,1);n_O)}(B')$ to $\Upsilon_{((2,1,\dots,1);n_O)}(B')$ for any given B'. □

3.4 Identifying $\widetilde{\mathcal{X}}$

Our goal is to identify the cellular complexes $\mathcal{X}$ that can be used to perform the desired homological computations and the strategy is summed up in Section 3.2. While this is straightforward for nodes of type $\mathcal{X}((n);n_O)$, we need to perform another step for nodes of type $\mathcal{X}((2,1,\ldots,1);n_O)$. The issue is that while $\Upsilon_{((n);n_O)}(B') \subset Z_{((n);n_O)}$ for any $B' \subset B^*((n);n_O)$, an affine space, this is not the case for $\Upsilon_{((2,1,\ldots,1);n_O)}(B') \subset Z_{((2,1,\ldots,1);n_O)}$, $B' \subset B^*((2,1,\ldots,1);n_O)$. Thus, in this section we make use of the notion of linearization introduced in [26] to produce an intermediate structure $\widetilde{\mathcal{X}}$ from which $\mathcal{X}((2,1,\ldots,1);n_O)$ is derived.

Starting with $\mathcal{P}((2,1,\ldots,1))$ and $\mathcal{Q}((2,1,\ldots,1);n_O)$, as defined in the introduction, we will construct *linearized polynomials* $\mathcal{P}'((2,1,\ldots,1))$ and $\mathcal{Q}'((2,1,\ldots,1);n_O)$, that define linearized parameter regions. We need following notations.

Definition 3.4.1. Suppose $\mathbf{n} \in \mathsf{N}^q$ is the interaction type for $f \in \mathsf{R}[z_1,\ldots,z_n]$. As in Definition 3.1.1 let $\{I_1,\ldots,I_q\}$ denote the indexing sets for each summand of f. Setting $I := \bigcup_{j=1}^{q} I_j$ we denote a typical element of I by $I_j(k)$ which we define as the k^{th} largest element of I_j. Let

$$E := \{\alpha : \{1,\ldots,n\} \to \{0,1\}\}$$

be the set of all Boolean functions defined on I. The *Boolean indexing map*, denoted by $B : E \to \{0,\ldots,2^{n-1}\}$, is defined by the formula

$$B(\alpha) := \sum_{j=1}^{q}\sum_{k=0}^{n_j-1} \alpha(I_j(k))2^{\kappa_{j,k}} \qquad \kappa_{j,k} = k + \sum_{j'=1}^{j-1} n_{j'}.$$

We will also consider, $\alpha \in E$, as a vector of Boolean functions defined as follows. Let E_j denote the set of Boolean functions defined on I_j. Then, elements of E can be represented as vectors of the form

$$\alpha = (\alpha_1,\ldots,\alpha_q) \quad \text{where } \alpha_j := \alpha\Big|_{I_j} \in E_j \quad \text{for } 1 \le j \le q.$$

Note that under this identification, E has the equivalent representation as $E = E_1 \times$

$\cdots \times E_q$.

We denote the polynomial ring in $2n$ indeterminates with Boolean indexing by

$$\mathcal{R} := \mathsf{R}\left[\{x_{r,\alpha_r} : \alpha_j \in E_r, 1 \le r \le n-1\}\right]. \tag{3.34}$$

For convenience, we sometimes use the binary integer represented by α_j to represent it in the subscribe. For example, sometimes we denote $x_{1,2} = x_{1,(1,0)}$ and $x_{1,3} = x_{1,(1,1)}$.

Define a collection of linear polynomials by

$$\mathcal{P}'((2,1,\ldots,1)) := \left\{p'_\alpha : \alpha \in E\right\} \subset \mathcal{R} \qquad \text{where} \quad p'_\alpha := \sum_{r=1}^{n-1} x_{r,\alpha_r}$$

and set

$$\mathcal{Q}'((2,1,\ldots,1);n_O) := \left\{q'_\alpha := \gamma' + \theta'_j - p'_\alpha : p'_\alpha \in \mathcal{P}'((2,1,\ldots,1)),\ j = 1,\ldots,n_O\right\}.$$

For $b \in B((2,1,\ldots,1);n_O)$ we define the *linearized parameter region* as

$$PN'(b) = \left\{(\gamma',\theta',\xi') : (-1)^{b(q)} q(\gamma',\theta',\xi') < 0, q \in \mathcal{Q}'((2,1,\ldots,1);n_0)\right\}$$

The following theorem explains the relationship between $PN(b)$ and $PN'(b)$

Theorem 3.4.1. *Consider a node of type* $((2,1,\ldots,1);n_O)$.

1. *Let* $\alpha \in E, j \in \{1,\ldots,n_O\}$, *and* $(\gamma,\theta,\xi) \in \mathsf{R}_+^{1+n_O+2n}$. *Set*

$$\gamma' = \log\gamma, \quad \theta'_j = \log\theta_j, \quad \textit{and} \quad \xi'_{j,\alpha_j} = \log\left(\sum_{k\in I_j} \xi_k + \alpha_j(k)\xi_{n+k}\right).$$

   Let $q_\alpha \in \mathcal{Q}((2,1,\ldots,1);n_O)$. *If* $q_\alpha(\gamma,\theta,\xi) \neq 0$, *then* $q'_\alpha(\gamma',\theta',\xi') \neq 0$ *and* $\operatorname{sign}(q'_\alpha(\gamma',\theta',\xi')) = \operatorname{sign}(q_\alpha(\gamma,\theta,\xi))$.

2. *Let* $b \in B((2,1,\ldots,1);n_O)$. *If* $PN_i(b) \neq \emptyset$, *then* $PN'(b) \neq \emptyset$.

Proof. To prove (1), without loss of generality suppose $q_\alpha = \gamma\theta_j - p_\alpha(\xi) < 0$, then

$$\gamma\theta_j < p_\alpha(\xi)$$
$$\gamma' + \theta'_j = \log(\gamma\theta_j) < \log(p_\alpha(\xi)) = p'_\alpha(\xi')$$

and hence $q_{\alpha'} < 0$. The same argument applies to $q'_\alpha > 0$.

For (2), it is easy to see if $\xi \in PN_i(b_i)$, then $\log \xi \in PN'_i(b_i)$. □

Now we can define $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}$ for each node type.

3.4.1 Node of type $((1,\ldots,1);n_O)$

From Theorems 3.3.1 and 3.7.1, we can reduce to the equivalent case of node type $((n);n_O)$ that we define in the following.

3.4.2 Node of type $((n);n_O)$

For node type $(\mathbf{n};n_O) = ((n);n_O)$, we define $\widetilde{\mathcal{X}}_{((n);n_O)}(B') = \Upsilon_{((n);n_O)}(B')$ for any $B' \subset B^*((n);n_O)$. This is because $\Upsilon_{((n);n_O)}(B')$ is already polyhedra defined by linear polynomials.

3.4.3 Node of type $((2,1,\ldots,1);n_O)$

Define

$$Z' = \{(\gamma',\theta',x) : \gamma' = 0, \sum_{j=1}^{n_O}\theta'_j + \sum_{j=1}^{n-1}\sum_{\alpha_j\in E_j} x_{j,\alpha_j} = 0,$$
$$-x_{1,0} + x_{1,1} + x_{1,2} - x_{1,3} > 0\}$$

and

$$D = \{(\gamma',\theta',x) : x_{1,0} < x_{1,1}, x_{1,2} < x_{1,3}; x_{j,0} < x_{j,1}, j = 2,\ldots,n-1\}$$

which we call boolean restriction for type $(\mathbf{n};n_O) = ((2,1,\ldots,1);n_O)$ according to paper [26].

Define the set $\widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O)}(B')$ for node of type $((2,1,\ldots,1);n_O)$ as following

$$\widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O)}(B') := \left(\bigcup_{b\in B'} \mathrm{cl}(PN'(b))\right) \cap Z' \cap D'$$

To move forward, we need the following technical lemma

Lemma 3.4.2. *Fix parameters, $x_0, x_1, x_2, x_3 \in \mathsf{R}$, and de ne the function, $g : \mathsf{R} \to \mathsf{R}$ by the formula*

$$g(t) = \exp(tx_0) - \exp(tx_1) - \exp(tx_2) + \exp(tx_3).$$

If $x_0 < x_1 \leq x_2 < x_3$, then g has a unique positive root if and only if $g'(0) < 0$. Denote the unique solution as $a(x_1, x_2, x_3, x_4)$, a function of (x_1, x_2, x_3, x_4), then a is a smooth function.

Proof. It is easy to see $t = 0$ is a solution for $f(t)$ and the differentiation at 0 is $f'(0) = x_0 - x_1 - x_2 + x_3$. If there is a $t_0 > 0$ such that $e^{t_0x_0} - e^{t_0x_1} - e^{t_0x_2} + e^{t_0x_3} = 0$, then by calculus we have $e^{y_1}(t_0x_0 - t_0x_1) + e^{y_2}(t_0x_3 - t_0x_2) = 0$ for some y_1, y_2 such that $t_0x_0 < y_1 < t_0x_1 \leq t_0x_2 < y_2 < t_0x_3$. So we must have $x_0 - x_1 - x_2 + x_3 < 0$ as $\exp^{y_1} < \exp^{y_2}$. That is to say if $f'(0) \geq 0$, $f(t)$ does not have positive solution.

Moreover, for fixed values $x_0, x_1, x_2, x_3 \in \mathsf{R}$, $x_0 < x_1 \leq x_2 < x_3$ and $x_0 - x_1 - x_2 + x_3 < 0$, we want to prove the there is exact one positive solution. The positive solvability is easy to prove, as $f(+\infty) = +\infty$, $f(0) = 0$ and $f'(0) < 0$ indicate there are solutions in $(0, +\infty)$.

The only thing left is uniqueness of positive solution. $f'(t_0) = x_0e^{t_0x_0} - x_1e^{t_0x_1} - x_2e^{t_0x_2} + x_3e^{t_0x_3}$, we want to show $f'(t_0)$ is strictly positive at any positive solution t_0. At $t_0 > 0$, $t_0f'(t_0)$ has the same sign as $f'(t_0)$. We denote $p_i = e^{t_0x_i}$ for $i = 0, 1, 2, 3$, then $t_0f'(t_0) = p_0 \ln p_0 - p_1 \ln(p_1) - p_2 \ln p_2 + p_3 \ln p_3$. Take $s = e^{t_0x_0} + e^{t_0x_3} = e^{t_0x_1} + e^{t_0x_2}$ and denote $q_i = \frac{p_i}{s}$. Then we have $\frac{t_0f'(t_0)}{s} = q_0 \ln q_0 - q_1 \ln(q_1) - q_2 \ln q_2 + q_3 \ln q_3$, $0 < q_0 < q_1 \leqq_2 < q_3 < 1$ and $q_0 + q_3 = q_1 + q_2 = 1$. By entropy inequality we have $\frac{t_0f'(t_0)}{s} > 0$. As $s > 0$, we have $t_0f'(t_0) > 0$ and $f'(t_0) > 0$.

As $f'(t_0) > 0$, we conclude that the positive solution set does not have a limit point

in R. Thus, we can assume there are two adjacent positive solutions u, v s.t $0 < u < v$ and $f(t) \neq 0$ on (u, v). As $f(u) = 0$ and $f'(u) > 0$, we have $f(t) > 0$ on (u, v). And similarly $f(v) = 0$ and $f'(v) > 0$ indicate $f(t) < 0$ on (u, v) which is a contradiction. So there is at most one positive solution for $f(t)$.

By the implicit function theorem and the fact that $f'(t_0) > 0$, the function $t_0 = a(x_1, x_2, x_3, x_4)$ is as smooth as f is smooth. □

Then we have the following important theorem for node type $((2, 1, \ldots, 1); n_O)$

Theorem 3.4.2. *For node type* $(\mathbf{n}; n_O) = ((2, 1, \ldots, 1); n_O)$, $\Upsilon_{((2,1,\ldots,1);n_O)}(B')$ *and* $\widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O)}(B')$ *are homotopic equivalent for any given* $B' \subset B^*((2, 1, \ldots, 1); n_O)$.

Proof. Recall we have $(\gamma, \theta, \ell, \delta) \in \Upsilon_{((2,1,\ldots,1);n_O)} \subset \mathsf{R}_+^{1+n_O+2n}$ and $(\gamma', \theta', x) \in \widetilde{\mathcal{X}}_{(\mathbf{n};n_O)} \subset \mathsf{R}^{1+n_O+2n}$.

Define $\tilde{f} : \mathsf{R}_+^{1+n_O+2n} \to \mathsf{R}^{1+n_O+2n}$ by

$$\tilde{f}(\gamma, \theta, \ell, \delta) = (\log \gamma, \log \theta, x) \tag{3.35}$$

where $x = (x_{j,\alpha_j})$ such that $x_{j,\alpha_j} = \log\left(\sum_{k \in I_j} l_k + \alpha_j(k)\delta_k\right)$.

Define $\tilde{g} : \mathsf{R}^{1+n_O+2n} \to \mathsf{R}_+^{1+n_O+2n}$ by

$$\tilde{g}(\gamma', \theta', x) = (\exp \gamma', \exp \theta', \ell, \delta) \tag{3.36}$$

where $\ell = (\ell_1, \ldots, \ell_n), \delta = (\delta_1, \ldots, \delta_n)$ such that $l_1 = l_2 = \frac{\exp^{x_{1,0}}}{2}, \delta_1 = \exp^{x_{1,1}} - \exp^{x_{1,0}}, \delta_2 = \exp^{x_{1,2}} - \exp^{x_{1,0}}, l_k = \exp^{x_{k-1,0}}\ \delta_k = \exp^{x_{k-1,1}} - \exp^{x_{k-1,0}},\ k = 3, \ldots, n$

Define $\tilde{F} : \mathsf{R}_+^{1+n_O+2n} \times [0, 1] \to \mathsf{R}_+^{1+n_O+2n}$ by

$$\tilde{F}(\gamma, \theta, \ell, \delta, t) = (\gamma, \theta, \ell, \delta) \tag{3.37}$$

i.e. identity map over $\mathsf{R}_+^{1+n_O+2n}$.

In order to move forward, define $a(x_0, x_1, x_2, x_3)$ to the unique positive value t for tuples (x_0, x_1, x_2, x_3), $x_0 < x_1, x_2 < x_3$ such that $\exp(tx_0) - \exp(tx_1) - \exp(tx_2) + \exp(tx_3) = 0$ given in Lemma 3.4.2.

Define $\tilde{G} : \mathsf{R}^{1+n_O+2n} \times [0,1] \to \mathsf{R}^{1+n_O+2n}$ by

$$\tilde{G}(\gamma', \theta, x, t) = (\gamma', ((1-t) + ta(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})) \cdot \theta', y) \tag{3.38}$$

where $y = (y_{\alpha_j,j})$ such that $(y_{1,0}(t), y_{1,1}(t), y_{1,2}(t), y_{1,3}(t)) = ((1-t)+ta(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3}))\cdot (x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})$, $y_{k,\alpha_k} = x_{k,\alpha_k}, k = 2, \ldots, n-1$.

We leave it to reaader to check that, for any given $B' \subset B^*((2,1,\ldots,1); n_O)$,

$$\begin{aligned}
&\tilde{f}(\Upsilon_{((2,1,\ldots,1);n_O)}(B')) \subset \widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O)}(B'),\\
&\tilde{g}(\widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O)}(B')) \subset \Upsilon_{((2,1,\ldots,1);n_O)}(B'),\\
&\tilde{F}(\cdot,0)|_{\Upsilon_{((2,1,\ldots,1);n_O)}(B')} = \mathrm{id}_{\Upsilon_{((2,1,\ldots,1);n_O)}(B')},\\
&\tilde{F}(\cdot,1)|_{\Upsilon_{((2,1,\ldots,1);n_O)}(B')} = \tilde{g} \circ f|_{\Upsilon_{((2,1,\ldots,1);n_O)}(B')},\\
&\tilde{G}(\cdot,0)|_{\widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O)}(B')} = \mathrm{id}_{\widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O)}(B')},\\
&\tilde{G}(\cdot,1)|_{\widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O)}(B')} = \tilde{f} \circ \tilde{g}|_{\widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O)}(B')}.
\end{aligned}$$

In another word, we show for any $B' \subset B((2,1,\ldots,1); n_O)$, $\Upsilon_{((2,1,\ldots,1);n_O)}(B')$ and $\widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O)}(B')$ are homotopic equivalent. □

3.5 Identifying $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}$

Given node type $(\mathbf{n}; n_O)$ and $B' \subset B^*(\mathbf{n}; n_O)$, in the previous section we have defined polyhedra $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')$ which are homotopic equivalent to the set $\widetilde{\Upsilon}_{(\mathbf{n};n_O)}(B')$ we are interested in. Even though $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')$ are polyhera sets, they are unbounded and neither closed nor open. In this section we will approximate $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')$ by a sequence of polyhedra complexes $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B')$ that are closed, bounded and have computable incidence number. As will be made clear $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B') \subset \mathcal{X}(B')$ and $\lim_{\epsilon\to 0, d\to\infty} \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B') = \widetilde{\mathcal{X}}(B')$. We warn the reader that for node types $((n); n_O)$ and $((1,\ldots,1); n_O)$, the definition of $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B')$ is independent of d, however we include it because it simplifies the statements of general results.

3.5.1 Node of type $((1,\ldots,1);n_O)$

From Theorem 3.3.1, we only need to consider its equivalent case when $(\mathbf{n};n_O) = ((n);n_O)$.

3.5.2 Node of type $((n);n_O)$

Define

$$D_{((n);n_O)}(\epsilon,d) = \{l_i \geq \epsilon, \delta_i \geq \epsilon, \theta_j \geq \epsilon\}.$$

For any given $B \subset B^*((n);n_O)$, define

$$\widetilde{\mathcal{X}}_{((n);n_O),\epsilon,d}(B') = \widetilde{\mathcal{X}}_{((n);n_O)}(B') \cap D_{((n);n_O)}(\epsilon,d)$$

It is not difficult to see $\widetilde{\mathcal{X}}_{((n);n_O),\epsilon,d}(B')$ is a (bounded, closed, finite) polyhedra complex defined by linear polynomials.

3.5.3 Node of type $((2,1,\ldots,1);n_O)$

Define

$$D_{((2,1,\ldots,1);n_O)}(\epsilon,d) = \{x_{1,j} - x_{1,0} \geq \epsilon, x_{1,3} - x_{1,j} \geq \epsilon, j = 1,2; x_{j,0} \geq -d, j = 1,\ldots$$
$$n-1; x_{j,1} - x_{j,0} \geq \epsilon, j = 2,\ldots n-1; \theta'_j \geq -d, j = 1,\ldots n_O\}$$

and for any given $B \subset B^*((2,1,\ldots,1)$, define

$$\widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O),\epsilon,d}(B') = \widetilde{\mathcal{X}}_{((2,1,\ldots,1);n_O)}(B') \cap D_{((2,1,\ldots,1)}(\epsilon,d)$$

for some $d, \epsilon > 0$. It is not difficult to see $\widetilde{\mathcal{X}}_{((2,1,\ldots,1),\epsilon,d}(B')$ is a (bounded, closed, finite) polyhedra complex defined by linear polynomials.

Proposition 3.5.1. *For any given node type* $(\mathbf{n};n_O)$ *and* $B' \subset B^*(\mathbf{n};n_O)$, $\epsilon,d > 0$, *then* $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B') \subset \widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')$. *Furthermore, if* $\epsilon_1 > \epsilon_2$ *and* $d_2 > d_1$, *then* $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B') \subset \mathrm{Int}_{\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')} \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B')$.

Proof. From definition $D_{(\mathbf{n};n_O)}(\epsilon_1, d_1) \subset \operatorname{Int} D_{(\mathbf{n};n_O)}(\epsilon_2, d_2)$. Then again from the definition of $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}$, we have

$$\begin{aligned}\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B') &\subset \widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B') \cap \operatorname{Int} D_{(\mathbf{n};n_O)}(\epsilon_2, d_2) \\ &= \operatorname{Int}_{\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')} \widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B') \cap D_{(\mathbf{n};n_O)}(\epsilon_2, d_2) \\ &= \operatorname{Int}_{\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')} \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B').\end{aligned}$$

□

The following theorem is fundamental for the approximation theorems in the following section.

Proposition 3.5.2. *Given node type* $(\mathbf{n}; n_O)$ *and any* $B' \subset B^*(\mathbf{n}; n_O)$*, for any compact set* $K \subset \widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')$*, there are vectors* $\epsilon > 0, d > 0$ *such that* $K \subset \operatorname{Int}_{\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')} \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B')$.

Proof. For definition of $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B')$ and $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')$, we konw

$$\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)} = \cup_{\epsilon>0,d>0} \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}$$

. From the second statement in Proposition 3.5.1, we know

$$\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B') = \cup_{\epsilon>0,d>0} \operatorname{Int}_{\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')} \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B')$$

with the relative topology over $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')$. Then by finite covering property and Proposition 3.5.1 again, we know there are $\epsilon > 0, d > 0$ such that $K \subset \operatorname{Int}_{\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')} \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B')$.

□

3.6 Approximation Theory

We will show there is a natural isomorphism between cell/polyhedra complex $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}$ and $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}$ when ϵ_1, ϵ_2 are small enough and d_1, d_2 are big enough. Furthermore, by using this isomorphism, we show when ϵ, d are chosen appropriately, $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B')$ and $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')$ are homotopic equivalent induced by the set inclusion map from $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B')$

to $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}(B')$. At first, we will introduce some linear algebra results

3.6.1 Application of Fourier–Motzkin elimination

Fourier–Motzkin elimination, abbreviated as FME, is an algorithm for eliminating variables from a system of linear inequalities. We refer the reader to [19] for more details. First, we have following lemma from FME algorithm

Lemma 3.6.1. *Fix matrix $A \in \mathsf{R}^{d_1 \times m}$, $B \in \mathsf{R}^{d_2 \times m}$ and $v \in \mathsf{R}^{d_1}, u \in \mathsf{R}^{d_2}$. Let $x = (x_1, \ldots, x_m)$ and $y = (x_2, \ldots, x_m)$. Consider the inequations systems $P_1 = \{A \cdot x = v, B \cdot x > u\}$ and $P_2 = \{A \cdot x = v, B \cdot x = u\}$ over variables x. If after applying Fourier elimination algorithm to eliminate x_1 we have the inequations $\tilde{P}_1 = \{D_i \cdot y < 0$ or $D_i \cdot y = 0 \mid i \in I_1\}$ and $\tilde{P}_2 = \{D_i \cdot y = 0 \mid i \in I_2\}$ for indexing sets I_1 and I_2 and vectors $D_i \in \mathsf{R}^{m-1}$. Then if $D_i \cdot y < 0$ or $D_i \cdot y = 0 \in \tilde{P}_1$ for some $i \in I_1$, we must have $D_i \cdot y = 0$ or $-D_i \cdot y = 0 \in \tilde{P}_2$.*

Proof. The proof of Lemma 3.6.1 follows directly from the Fourier–Motzkin elimination algorithm, but a formal presentation is tedious and technical. Instead we provide the following example. Suppose we have inequalities $0 < -\sum_{i=2}^{m} a_i x_i + x_1 \in P_1, 0 < \sum_{i=2}^{m} b_i x_i - x_1 \in P_1$, then we must have equations $0 = -\sum_{i=2}^{m} a_i x_i + x_1 \in P_2, 0 = \sum_{i=2}^{m} b_i x_i - x_1 \in P_2$. By using the FME algorithm we have inequality $\sum_{i=2}^{m}(b_i - a_i)x_i > 0 \in \tilde{P}_1$ and equation $\sum_{i=2}^{m}(b_i - a_i)x_i = 0 \in \tilde{P}_1$ or $\sum_{i=2}^{m}(a_i - b_i)x_i = 0 \in \tilde{P}_1$. □

From Lemma 3.6.1 above we have the following propositions

Proposition 3.6.2. *Consider polyhedra of the form*

$$C(\epsilon) = \{x \in \mathsf{R}^m | A \cdot x = v_0 + \epsilon v_1, B \cdot x > v_2 + \epsilon v_3\}.$$

where $A \in \mathsf{R}^{d_1 \times m}$, $B \in \mathsf{R}^{d_2 \times m}$, $v_0, v_1 \in \mathsf{R}^{d_1}, v_2, v_3 \in \mathsf{R}^{d_2}$, $x = (x_1, \ldots, x_m) \in \mathsf{R}^m$ and $\epsilon \in \mathsf{R}$. Then, there exists an $\epsilon' > 0$ with the following properties.

1. *If $C(\epsilon_0) \neq \emptyset$ for some $\epsilon_0 \in (0, \epsilon')$, then $C(\epsilon) \neq \emptyset$ for any $\epsilon \in (0, \epsilon')$.*

2. *This ϵ' is only dependent on the combined matrix* $\begin{pmatrix} A & v_0 & v_1 \\ B & v_2 & v_3 \end{pmatrix}$ *and is independent of the row partition with respect to inequality or equality.*

Proof. Consider inequation systems $P_1 = \{A \cdot x = v_0 + \epsilon v_1, B \cdot x > v_2 + \epsilon v_3\}$ and $P_2 = \{A \cdot x = v_0 + \epsilon v_1, B \cdot x = v_2 + \epsilon v_3\}$ over variables $x' = (x, \epsilon)$. After applying Fourier elimination to $(x_1, \ldots, x_m)$, we end at reduced inequation systems over variable ϵ as $\tilde{P}_1 = \{a_i\epsilon - b_i < 0 \text{ or } a_i\epsilon - b_i = 0\}_{i \in I_1}$ and $\tilde{P}_2 = \{a_i\epsilon - b_i = 0\}_{i \in I_2}$ such that $I_1 \subset I_2$ by Lemma 3.6.1. Take $\epsilon' = \min_{a_i \neq 0, \frac{b_i}{a_i} > 0, i \in I_2} \frac{b_i}{a_i}$ and if $\{i \in I_2 | a_i \neq 0, \frac{b_i}{a_i} > 0\} = \emptyset$, take $\epsilon' = \infty$. If there is a $0 < \epsilon_0 < \epsilon'$ such that $C(\epsilon_0) \neq \emptyset$, $\tilde{P}_1$ can only contain inequality either $\frac{b_i}{a_i} < \epsilon$, if $\frac{b_i}{a_i} < 0$ or $\epsilon < \frac{b_i}{a_i}$, if $\frac{b_i}{a_i} > 0$ for some $i \in I_1$. If $\frac{b_i}{a_i} > 0$ we must have $\epsilon_0 < \epsilon' \leq \frac{b_i}{a_i}$ from the definition of ϵ'. So for any $\epsilon'' \in (0, \epsilon')$, we have $C(\epsilon'') \neq \emptyset$. From the definition of P_2, it is easy to see P_2 is the collection of the row equations of the matrix equation

$$\begin{pmatrix} A & v_0 & v_1 \\ B & v_2 & v_3 \end{pmatrix} \cdot \begin{pmatrix} x \\ -1 \\ -\epsilon \end{pmatrix} = 0$$

over variables (x, ϵ). It is a direct result of application of FME algorithm to the equation system P_2 such that $\tilde{P}_2$ is only dependent on the coefficient matrix $\begin{pmatrix} A & v_0 & v_1 \\ B & v_2 & v_3 \end{pmatrix}$. In another word, the ϵ' derived from $\tilde{P}_2$ is only dependent on the coefficient matrix or the combined matrix. □

The following is a similar proposition with parameters d, ϵ

Proposition 3.6.3. *Consider a polyhedron*

$$C(\epsilon, d) = \{x \in \mathsf{R}^m | A \cdot x = v_0 + \epsilon v_1 + d u_0, B \cdot x > v_2 + \epsilon v_3 + d u_1\}.$$

where $A \in \mathsf{R}^{d_1 \times m}$, $B \in \mathsf{R}^{d_2 \times m}$, $v_0, v_1, u_0 \in \mathsf{R}^{d_1}, v_2, v_3, u_1 \in \mathsf{R}^{d_2}$, $x = (x_1, \ldots, x_m) \in \mathsf{R}^m$ *and* $\epsilon, d \in \mathsf{R}$. *Then, there exists an* $\epsilon', d' > 0$ *with the following properties.*

1. *If* $C(\epsilon_0, d_0) \neq \emptyset$ *for some* $\epsilon_0 \in (0, \epsilon'), d_0 \in (d', \infty)$, *then* $C(\epsilon, d) \neq \emptyset$ *for any*

$\epsilon \in (0, \epsilon'),\ d \in (d', \infty)$.

2. *This ϵ', d' is only dependent on the combined matrix $\begin{pmatrix} A & v_0 & v_1 & u_0 \\ B & v_2 & v_3 & u_1 \end{pmatrix}$ and is independent of the row partition with respect to inequality or equality.*

Proof. The details are left to the reader as the argument is the same as the proof of Proposition 3.6.2, we only show how to compute ϵ', d'. Consider inequation system $P = \{A \cdot x = v_0 + \epsilon v_1 + d_i u_0, B \cdot x = v_2 + \epsilon v_3 + d_i u_1\}$ over variable (x, d, ϵ). Suppose the system after eliminating variables x and (x, d) are inequations systems $P_1 = \{c_i d_i = a_i \epsilon_i + b_i\}_{i \in I_1}$ and $P_2 = \{a_i \epsilon = b_i\}_{i \in I_2}$ respectively. Then $\epsilon' = \min_{a_i \neq 0, \frac{b_i}{a_i} > 0, i \in I_2} \frac{b_i}{a_i}$ and $d' = \max_{c_i \neq 0, i \in I_1,} |\frac{a_i}{c_i}|\epsilon' + |\frac{b_i}{c_i}|$. If $\{i \in I_2 | a_i \neq 0, \frac{b_i}{a_i} > 0\} = \emptyset$, take $\epsilon' = \infty$ and if $\{i \in I_1 | c_i \neq 0\} = \emptyset$ take $d' = 0$. □

Given $B' \subset B^*(\mathbf{n}; n_O)$ we define chain complex of $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B')$ as $C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B'))$ and any $c_{\epsilon,d} \in C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B'))$ can be identified as a nonempty polyhedron, or explicitly identify $c_{\epsilon,d} = \{x \in \mathsf{R}^m | A \cdot x = v_0 + \epsilon v_1 + d u_0, B \cdot x > v_2 + \epsilon v_3 + d u_1\}$. It is direct that $c_{\epsilon,d} \neq \emptyset$ and the combined matrix $\begin{pmatrix} A & v_0 & v_1 & u_0 \\ B & v_2 & v_3 & u_1 \end{pmatrix}$ is only dependent on node type $(\mathbf{n}; n_O)$ from the definition of $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}$. The dimension and adjacency of cells in the chain complex $C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}(B'))$ are completely determined by the defining equations of their corresponding polyhedron. Observe the emptiness of the polyhedron $c_{\epsilon,d}$ can vary with different ϵ, d. For some ϵ, d, $c_{\epsilon,d} = \emptyset$, for others $c_{\epsilon,d} \neq \emptyset$. However, Propositions 3.6.2 and 3.6.3 show that for each node type $(\mathbf{n}; n_O)$ there is $\epsilon_{(\mathbf{n};n_O)}, d_{(\mathbf{n};n_O)}$ such that if $0 < \epsilon_1, \epsilon_2 < \epsilon_{(\mathbf{n};n_O)}$ and $0 < d_{(\mathbf{n};n_O)} < d_1, d_2$, given any $c_{\epsilon_1,d_1} \in C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1})$, we have $c_{\epsilon_2,d_2} \neq \emptyset$ and $c_{\epsilon_2,d_2} \in C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2})$. This observation can induce a chain isomorphism and a homeomorphism in the following sections.

3.6.2 Maps

From definition $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}$, We have the following theorem over the cell complex

Theorem 3.6.1. *Given node type $(\mathbf{n}; n_O)$ and $B' \subset B^*(\mathbf{n}; n_O)$, there is $\epsilon_{(\mathbf{n};n_O)} > 0, d_{(\mathbf{n};n_O)} > 0$ such that for any $0 < \epsilon_1, \epsilon_2 < \epsilon_{(\mathbf{n};n_O)},\ d_1, d_2 > d_{(\mathbf{n};n_O)}$, there is a well*

de ned homomorphism

$$f : C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B')) \to C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B'))$$

$$\textit{s.t. } f(\sum_i a_i c^i_{\epsilon_1,d_1}) = \sum_i a_i c^i_{\epsilon_2,d_2}.$$

Moreover, f is an chain isomorphism over coe cients eld Z_2, i.e. $a_i \in \mathsf{Z}_2$.

Proof. From the definition of the cell complex in $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}$, we know

$$c_{\epsilon,d} = \{x \in \mathsf{R}^m | A \cdot x = v_0 + \epsilon v_1 + d u_0, B \cdot x > v_2 + \epsilon v_3 + d u_1\}$$

where $A \in \mathsf{R}^{d_1 \times m}$, $B \in \mathsf{R}^{d_2 \times m}$, $v_0, v_1, u_0 \in \mathsf{R}^{d_1}, v_2, v_3, u_1 \in \mathsf{R}^{d_2}$, $x = (x_1, \ldots, x_m) \in \mathsf{R}^m$ and $\epsilon, d \in \mathsf{R}$, and the matrix $\begin{pmatrix} A & v_0 & v_1 & u_0 \\ B & v_2 & v_3 & u_1 \end{pmatrix}$ is only dependent on the $\mathcal{Q}(\mathbf{n}; n_O)$, $Z_{(\mathbf{n};n_O)}(\epsilon, d)$, $D_{(\mathbf{n};n_O)}(\epsilon, d)$ and $\epsilon_{(\mathbf{n};n_O)}, d_{(\mathbf{n};n_O)}$, i.e. only dependent on node type $(\mathbf{n}; n_O)$. Then by using proposition 3.6.3, there are $0 < \epsilon_{(\mathbf{n};n_O)}, d_{(\mathbf{n};n_O)}$ only dependent on node type such that if $0 < \epsilon_1, \epsilon_2 < \epsilon_{(\mathbf{n};n_O)}, d_{(\mathbf{n};n_O)} < d_1, d_2$ then

$$c_{\epsilon_1,d_1} \in C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B')) \text{ if and only if } c_{\epsilon_2,d_2} \in C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B')),$$

or equivalently $c_{\epsilon_1,d_1} \neq \emptyset$ if and only if $c_{\epsilon_2,d_2} \neq \emptyset$. We also have

$$\dim c_{\epsilon_1,d_1} = \dim c_{\epsilon_2,d_2} = \dim \ker A.$$

Moreover, suppose $c'_{\epsilon_1,d_1} = \{x \in \mathsf{R}^m | A' \cdot x = v'_0 + \epsilon v'_1 + d' u_0, B' \cdot x > v'_2 + \epsilon v'_3 + d' u_1\} \in C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B'))$ then $c'_{\epsilon_1,d_1} \prec c_{\epsilon_1,d_1}$ if and only if the inclusion of equation system holds, i.e. $\{A \cdot x = v_0 + \epsilon v_1 + d u_0\} \subset \{A' \cdot x = v'_0 + \epsilon v'_1 + d u'_0\}$ over variables of x, ϵ, d. In another word, $c'_{\epsilon_1,d_1} \prec c_{\epsilon_1,d_1}$ if and only if $c'_{\epsilon_2,d_2} \prec c_{\epsilon_2,d_2}$. So we have f defined by

$$f : C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B')) \to C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B'))$$

$$\text{s.t. } f(\sum_i a_i c^i_{\epsilon_1,d_1}) = \sum_i a_i c^i_{\epsilon_2,d_2}.$$

is a chain isomorphism. In the following, we will only discuss homology group under coefficient field Z_2. □

It is worth noting that even though the above variant of FME can compute $\epsilon_{(\mathbf{n};n_O)}, d_{(\mathbf{n};n_O)}$, the complexity of the algorithm is doubly exponential i.e. m^{2^n}, where m, n are the row and column number of the combined matrix. We do not know if it is possible to improve on these complexity bounds. The following theorem leads to strict candidates for $\epsilon_{(\mathbf{n};n_O)}$, $d_{(\mathbf{n};n_O)}$ which depends on m, n, but the value for $\epsilon_{(\mathbf{n};n_O)}$ rapidly becomes unreasonably small.

Theorem 3.6.2. *Consider the combined matrix, denote as D, given in proposition 3.6.3, suppose the first k columns of D has $||D(:,:k)||_{L^\infty} \leq a$ and the rest columns has $||D(:,k:)||_{L^\infty} \leq b$ and $D_{i,j} \in \mathsf{Z}$. After eliminating the k variables corresponding to the first k columns of D and the remaining inequaitons system is expressed by a matrix denoted D^k and $D^k_{i,j} \in \mathsf{Z}$, then we have $||D^k||_{L^\infty} \leq 2^{2^k-1}a^{2^k}b$.*

Proof. Suppose we have $v_1 \cdot (x_2, \ldots, x_m) - a_1 x_1 < 0$, $v_2 \cdot (x_2, \ldots, x_m) - a_2 x_1 > 0$ in the inequation system defining $C(\epsilon, d)$ for some $v_1, v_2 \in \mathsf{Z}^{m-1}$ and the coefficients $a_1, a_2 \in \mathsf{Z}_+$. From Fourier elimination, the induced inequation over x_{-1} is given by $a_2 v_1(x_2, \ldots, x_m) - a_1 v_2(x_2, \ldots, x_m) < 0$. It is direct to see the coefficients of $a_2 p_1(x_{-1}) - a_1 p_2(x_{-1})$ are bounded by $2a^2$ and are integers which shows $||D^1(:,:k-1)||_{L^\infty} \leq 2a^2$ and $||D^1(:,k-1:)||_{L^\infty} \leq 2a^2 b$. Apply this formula k times, we get the bound for D^k. □

Remark 3.6.3. In this paper or all the computation in section 3.8, we use FME algorithm to compute the $\epsilon_{(\mathbf{n};n_O)}$ and $d_{(\mathbf{n};n_O)}$ directly. However, the strict bound given by Theorem 3.6.2, will be useful in the future application when deal with more complicated nodes in network [24].

From Proposition 3.6.3 and Theorem 3.6.2, given node i with an interaction type of n_i inputs, m_i outputs, we can choose any $\epsilon_{(\mathbf{n};n_O)} \leq \frac{1}{2^{2^{2n_i+m_i}}}, d_{(\mathbf{n};n_O)} \geq m_i$ where m_i is the number of out edges and as we always have the coefficient of D to be $-1, 0, 1$ for variables l, δ or x_{j,α_j} and bounded by m for ϵ, d.

Before moving on, we need one more theorem demonstrating homotopic equivalence.

Theorem 3.6.4. *Given node type* $(\mathbf{n}; n_O)$, $B' \subset B^*(\mathbf{n}; n_O)$ *and the* $\epsilon_{(\mathbf{n};n_O)}, d_{(\mathbf{n};n_O)}$ *chosen in Theorem 3.6.1. Suppose* $0 < \epsilon_2 < \epsilon_1 < \epsilon_{(\mathbf{n};n_O)}$, $d_2 > d_1 > d_{(\mathbf{n};n_O)}$, *then there is a homeomorphism*

$$\check{f} : \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B') \to \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B')$$

such the set inclusion map

$$ic : \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B') \to \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B')$$

are homotopic equivalent.

Proof. The $\check{f}$ is constructive. Suppose the set of 0-cells in $C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1})(B'))$ is given by $\{c^1_{\epsilon_1,d_1}, \ldots, c^n_{\epsilon_1,d_1}\}$. From the definition of cells $C_\bullet(\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B'))$, we know the coordinates of $c^k_{\epsilon_1,d_1}$ can be identified as linear functions $c_{\epsilon,d}$ evaluated at ϵ_1, d_1. From theorem 3.6.1, $\{c^1_{\epsilon_2,d_2}, \ldots, c^n_{\epsilon_2,d_2}\}$ is the set of 0-cells of $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B')$. Furthermore by the linearity of coordinates for those 0-cells, we have $tc^k_{\epsilon_1,d_1} + (1-t)c^k_{\epsilon_2,d_2} = c^k_{t\epsilon_1+(1-t)\epsilon_2,td_1+(1-t)d_2}, t \in [0,1]$. For convenience we denote

$$\epsilon_t = t\epsilon_1 + (1-t)\epsilon_2, d_t = d_1 + (1-t)d_2.$$

Now we assume that $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B')$ and $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B')$ are both simiplicial complex. We can construct a homotopic map

$$\check{F} : \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B') \times [0,1] \to \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B'(B'))$$

by using the fact each point in $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon,d}$ is an unique convex combination of the vertex of the face that the point resides. More precisely, define

$$\begin{aligned} \check{F}(t,x) &= \sum \alpha_k((1-t)c^k_{\epsilon_1,d_1} + tc^k_{\epsilon_2,d_2} \\ &= \sum \alpha_k c^k_{\epsilon_t,d_t} \end{aligned} \tag{3.39}$$

for any $x = \sum_{k\in K} \alpha_k c^k_{\epsilon_1,d_1} \in \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B')$ where x in in the relative interior of the simplex spanned by $\{c^k_{\epsilon_1,d_1}\}_{k\in K}$. Observe this map is well defined and continuous. Moreover

$$\check{F}(x,0) = \mathrm{ic},$$

i.e. the set inclusion map from $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B')$ to $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B')$. Define

$$\check{f} = \check{F}(x,1) \tag{3.40}$$

which is a homoemorphism , actually a homeomorphic simplicial isomorphism, from construction above and

$$\check{F}(x,t) \subset \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_t,d_t}(B') \subset \widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B').$$

If the complex is not simplicial, then we consider its barycentric subdivision over each cells which are convex polytope and denote the resulting barycentric subdivision as $\mathrm{sub}\,\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B')$ and $\mathrm{sub}\,\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B')$. As the barycentric subdivision is only dependent on adjacency and dimension of cells, from Theorem 3.6.1, we know the 0-cells, denote as $p^0_{\epsilon_1,d_1},\ldots,p^m_{\epsilon_1,d_1}$, in $\mathrm{sub}\,\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B')$ are linear combination of $c^1_{\epsilon_1,d_1},\ldots,c^n_{\epsilon_1,d_1}$ and again $p^i_{\epsilon_1,d_1}$ can be identified as a linear function $p^i_{\epsilon,d}$ evaluated at ϵ_1,d_1. By barycentric subdivision and Theorem 3.6.1, we know $p^0_{\epsilon_2,d_2},\ldots,p^m_{\epsilon_2,d_2}$ is the set of 0-cells for $\mathrm{sub}\,\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B')$ and $\{p^0_{\epsilon_1,d_1}\}_{k\in K}$ spans a cell in $\mathrm{sub}\,\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B')$ if and only if $\{p^k_{\epsilon_2,d_2}\}_{k\in K}$ spans a cell in $\mathrm{sub}\,\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B')$. Then we can prove the same result for simplicial complex $\mathrm{sub}\,\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_1,d_1}(B')$ and $\mathrm{sub}\,\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon_2,d_2}(B')$. □

3.7 Results for the network

In this section we will prove theorems for the network parameter space based the facts of factor parameter space for each node that we built in all previous sections. In order to move forward, we suppose the network have N nodes and each of them is of node type $(\mathbf{n};n_O)_i$.

For $B' \subset B^*_{nw}$ define

$$\begin{aligned}
\Upsilon_{nw}(B') &= \widetilde{\Upsilon}_{nw}(B') \cap \prod_{i=1}^{N} Z_{(\mathbf{n}_i; n_{O,i})} \\
\widetilde{\mathcal{X}}_{nw}(\{b'\}) &= \prod_{i=1}^{N} \tilde{\mathcal{X}}_{(\mathbf{n}_i; n_{O,i})}(\{b'_i\}) \text{ s.t. } b' = (b'_i)_{i=1}^{N} \in B^*_{nw} \\
\widetilde{\mathcal{X}}_{nw}(B') &= \bigcup_{b' \in B'} \widetilde{\mathcal{X}}_{nw}(\{b'\}) \\
\widetilde{\mathcal{X}}_{nw,\epsilon,d}(B') &= \widetilde{\mathcal{X}}_{nw}(B') \cap \prod_{i=1}^{N} D_{(\mathbf{n}_i; n_{O,i}),\epsilon_i,d_i} \\
&\text{such that } \epsilon = (\epsilon_1, \ldots, \epsilon_N) > 0, d = (d_1, \ldots, d_N) > 0
\end{aligned} \tag{3.41}$$

In order to move on we need three useful Lemmas

Lemma 3.7.1. *Consider topological spaces $\prod_{i=1}^{N} X_{i,j} \subset \prod_{i=1}^{N} S_i$ and $\prod_{i=1}^{N} X'_{i,j} \subset \prod_{i=1}^{N} S'_i$ where $j = 1, \ldots, K$. Assume that there are maps $f_i : S_i \to S'_i$, $g_i : S'_i \to S_i$, $F_i : S_i \times [0,1] \to S_i$, and $G_i : S'_i \times [0,1] \to S'_i$ that satisfy the following properties:*

$$\begin{aligned}
\operatorname{Img} f_i|_{X_{i,j}} &\subset X'_{i,j} \\
\operatorname{Img} g_i|_{X'_{i,j}} &\subset X_{i,j} \\
\operatorname{Img} F_i(\cdot,\cdot)|_{X_{i,j} \times [0,1]} &\subset X_{i,j} \\
F_i(\cdot,0)|_{X_{i,j}} &= \mathrm{id}_{X_{i,j}} \\
F(\cdot,1)|_{X_{i,j}} &= g_i|_{X'_{i,j}} \circ f_i|_{X_{i,j}} \\
\operatorname{Img} G_i(\cdot,\cdot)|_{X'_{i,j} \times [0,1]} &\subset X'_{i,j} \\
G_i(\cdot,0)|_{X'_{i,j}} &= \mathrm{id}_{X'_{i,j}} \\
G_i(\cdot,1)|_{X'_{i,j}} &= f_i|_{X_{i,j}} \circ g_i|_{X'_{i,j}}
\end{aligned}$$

for any $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, K\}$. Then,

$$\bigcup_{j=1}^{K} \prod_{i=1}^{N} X_{i,j} \quad \textit{and} \quad \bigcup_{j=1}^{K} \prod_{i=1}^{N} X'_{i,j}$$

are homotopy equivalent.

Proof. For convenience we denote $X_j = \prod_{i=1}^N X_{i,j}$, $X'_j = \prod_{i=1}^N X'_{i,j}$, $X = \bigcup_{j=1}^K X_j$, and $X' = \bigcup_{j=1}^K X'_j$. Define $f = \prod_{i=1}^N f_i$, $g = \prod_{i=1}^N g_i$, $F(\cdot,t) = \prod_{i=1}^N F_i(\cdot,t)$, and $G(\cdot,t) = \prod_{i=1}^N G_i(\cdot,t)$. We will show that for the maps $f|_X$ and $g|_{X'}$, $g|_{X'} \circ f|_X$ is homotopic to id_X and $f|_X \circ g|_{X'}$ is homotopic to id_Y. From the hypothesis we know that $\mathrm{Img}\, f_X \subset X'$, and $\mathrm{Img}\, g_{X'} \subset X$, so $f|_X$, $g|_{X'}$, and $g|_{X'} \circ f|_X$ are well defined. Similarly, we have $\mathrm{Img}\, F|_{I\times X} = \cup_j \mathrm{Img}\, F|_{X_j\times[0,1]} \subset \cup_j X_j = X$ and $F|_{X\times[0,1]}$ is a homotopy between id_X and $g|_{X'} \circ f|_X$. Symmetrically, $\mathrm{Img}\, G|_{X'\times[0,1]} = \cup_j \mathrm{Img}\, G|_{X'_j\times[0,1]} \subset \cup_j X'_j = X'$ and $G|_{X'\times[0,1]}$ is a homotopy between $\mathrm{id}_{X'}$ and $f|_X \circ g|_{X'}$. These show that X and X' are homotopic equivalent. □

Lemma 3.7.2. *Consider topological spaces $\prod_{i=1}^N X_{i,j}$, $j = 1,\ldots,K$, $\prod_{i=1}^N Z_i$, $\prod_{i=1}^N X_{i,j} \subset \prod_{i=1}^N S_i$ and $\prod_{i=1}^N Z_i \subset \prod_{i=1}^N S_i$. Assume there are maps $F_i : S_i \times [0,1] \to S_i$ that satisfy the following properties:*

$$\begin{aligned}
\mathrm{Img}\, F_i|_{X_{i,j}\times[0,1]} &\subset X_{i,j} \\
F_i(\cdot,0)|_{X_{i,j}} &= \mathrm{id}_{X_{i,j}} \\
\mathrm{Img}\, F_i(\cdot,1)|_{X_{i,j}} &\subset X_{i,j} \cap Z_i \\
F_i(\cdot,t)|_{X_{i,j}\cap Z_i} &= \mathrm{id}_{X_{i,j}\cap Z_i}
\end{aligned}$$

for any $i \in \{1,\ldots,N\}$ and $j \in \{1,\ldots,K\}$. Then $(\cup_j \prod_{i=1}^N X_{i,j}) \cap \prod_{i=1}^N Z_i$ is a strong deformation retraction of $\cup_j \prod_{i=1}^N X_{i,j}$.

Proof. We prove it directly from the definition. Define $F(\cdot,t) = \prod_{i=1}^N F_i(\cdot,t)$ and denote $X = \cup_j \prod_{i=1}^N X_{i,j}$, $Z = \prod_{i=1}^N Z_i$. Then from the definition we have $\mathrm{Img}\, F|_{I\times X} \subset X$, $F(\cdot,0)|_X = \mathrm{id}_X$, $\mathrm{Img}\, F(\cdot,1)|_X \subset X \cap Z$ and $F(\cdot,t)|_{X\cap Z} = \mathrm{id}_{X\cap Z}$. So $X \cap Z$ is a strong deformation retraction of X. □

Lemma 3.7.3. *Consider topological spaces $\prod_{i=1}^N X_{i,j}$, $\prod_{i=1}^N X'_{i,j}$, $j = 1,\ldots,K$, $\prod_{i=1}^N S_i$, $\prod_{i=1}^N S'_i$ and $\prod_{i=1}^N X_{i,j} \subset \prod_{i=1}^N S_i$, $\prod_{i=1}^N X'_{i,j} \subset \prod_{i=1}^N S'_i$. Assume there are maps*

$f_i : S_i \to S_i'$, $g_i : S_i \to S_i'$, $F_i : S_i \times [0,1] \to S_i'$ that satisfy the following properties:

$$\begin{aligned}
\operatorname{Img} f_i|_{X_{i,j}} &\subset X'_{i,j} \\
\operatorname{Img} g_i|_{X_{i,j}} &\subset X'_{i,j} \\
\operatorname{Img} F_i(\cdot,\cdot)|_{X_{i,j}\times[0,1]} &\subset X'_{i,j} \\
F_i(\cdot,0)|_{X_{i,j}} &= f_{X_{i,j}} \\
F(\cdot,1)|_{X_{i,j}} &= g_i|_{X_{i,j}}
\end{aligned}$$

for any $i \in \{1,\dots,N\}$, $j \in \{1,\dots,K\}$. In another word, $F_i|_{X_{i,j}}$ is the homotopy between $f_i|_{X_{i,j}}, g_i|_{X_{i,j}}$. De ne

$$\begin{aligned}
f &= \prod_{i=1}^{N} f_i \\
g &= \prod_{i=1}^{N} g_i \\
F(\cdot,t) &= \prod_{i=1}^{N} F_i(\cdot,t).
\end{aligned}$$

Then $F|_{X\times[0,1]}$ is a homotopy between $f|_X, g|_X$ where $X = \cup_{j=1}^{K} \prod_{i=1}^{N} X_{i,j}$.

Proof. Direct from definition, details are left to readers. □

3.7.1 Equivalence

In this section we show the node type $(\mathbf{n}; n_O) = ((1,\dots,1); n_O)$ can be replaced by node type $(\mathbf{n}; n_O) = ((n); n_O)$ in the study of homology by the the following Theorem.

Theorem 3.7.1. *Given two networks nw_1, nw_2 with node type $((\mathbf{n}_i; n_{O,i}))_{i=1}^N$ and $((\mathbf{n}_i; n_{O,i})')_{i=1}^N$ such that*

$$\begin{aligned}
(\mathbf{n}_i; n_{O,i}) &= (\mathbf{n}_i; n_{O,i})', \text{ if } (\mathbf{n}_i; n_{O,i}) \neq ((1,\dots,1); n_O) \\
(\mathbf{n}_i; n_{O,i})' &= ((n); n_O), \text{ if } (\mathbf{n}_i; n_{O,i}) = ((1,\dots,1); n_O)
\end{aligned}$$

It easy to see $B^*_{nw_1} = B^*_{nw_2}$. *For any* $B' \subset B^*_{nw_1} = B^*_{nw_2}$, $\widetilde{\Upsilon}_{nw_1}(B')$ *and* $\widetilde{\Upsilon}_{nw_2}(B')$ *are homtopic equivalent.*

Proof. The proof is a simple application of Lemma 3.7.1. For convenience, denote $B' = \{b'_k\}_{k=1}^K$ and $b'_k = (b'_{i,k})_{i=1}^N$. Define

$$X_{i,j} = \widetilde{\Upsilon}_{(\mathbf{n}_i;n_{O,i})}(\{b_{i,j}\}); X'_{i,j} = \widetilde{\Upsilon}_{(\mathbf{n}_i;n_{O,i})'}(\{b_{i,j}\})$$

$$S_i = S'_i = (0,\infty)^{1+2n_i+n_{O,i}}$$

$$f_i = \begin{cases} \mathrm{id}_{(0,\infty)^{1+2n_i+n_{O,i}}}, & (\mathbf{n}_i;n_{O,i}) \neq ((1,\ldots,1);n_{O,i}) \\ \bar{f}, & (\mathbf{n}_i;n_{O,i}) = ((1,\ldots,1);n_{O,i}) \end{cases}$$

$$g_i = \begin{cases} \mathrm{id}_{(0,\infty)^{1+2n_i+n_{O,i}}}, & (\mathbf{n}_i;n_{O,i}) \neq ((1,\ldots,1);n_{O,i}) \\ \bar{g}, & (\mathbf{n}_i;n_{O,i}) = ((1,\ldots,1);n_{O,i}) \end{cases}$$

$$F_i(\cdot,t) = \begin{cases} \mathrm{id}_{(0,\infty)^{1+2n_i+n_{O,i}}}, & (\mathbf{n}_i;n_{O,i}) \neq ((1,\ldots,1);n_{O,i}) \\ \bar{F}(\cdot,t), & (\mathbf{n}_i;n_{O,i}) = ((1,\ldots,1);n_{O,i}) \end{cases}$$

$$G_i(\cdot,t) = \begin{cases} \mathrm{id}_{(0,\infty)^{1+2n_i+n_{O,i}}}, & (\mathbf{n}_i;n_{O,i}) \neq ((1,\ldots,1);n_{O,i}) \\ \bar{G}(\cdot,t), & (\mathbf{n}_i;n_{O,i}) = ((1,\ldots,1);n_{O,i}) \end{cases}$$

where n_i is the number of inputs of node i or the sum of vector $\mathbf{n}_i$ and the $\bar{f}, \bar{g}, \bar{F}, \bar{G}$

are defined as

$$
\begin{aligned}
&\bar{f} : (0,\infty)^{1+2n_i+n_{O,i}} \to (0,\infty)^{1+2n_i+n_{O,i}} \\
&\text{s.t. } \bar{f} = \log \circ G_1(F(\cdot,1),1) \\
&\bar{g} : (0,\infty)^{1+2n_i+n_{O,i}} \to (0,\infty)^{1+2n_i+n_{O,i}} \\
&\text{s.t. } \bar{g} = \exp \circ G_2(F(\cdot,1),1) \\
&\bar{F} : (0,\infty)^{1+2n_i+n_{O,i}} \times [0,1] \to (0,\infty)^{1+2n_i+n_{O,i}} \\
&\bar{F}(\cdot,t) = \begin{cases} G_1(F(\cdot,2t),0), & t \in [0,\frac{1}{2}] \\ G_1(F(\cdot,1),2t-1) & t \in [\frac{1}{2},1] \end{cases} \\
&\bar{G} : (0,\infty)^{1+2n_i+n_{O,i}} \times [0,1] \to (0,\infty)^{1+2n_i+n_{O,i}} \\
&\bar{G}(\cdot,t) = \begin{cases} G_2(F(\cdot,2t),0), & t \in [0,\frac{1}{2}] \\ G_2(F(\cdot,1),2t-1) & t \in [\frac{1}{2},1] \end{cases}
\end{aligned}
\tag{3.42}
$$

where $F, G_1, G_2, \log$ are defined in Theorem 3.3.1 by (3.25, 3.26, 3.27, 3.28). By using Theorem 3.3.1 and Lemma 3.7.1, we have $\widetilde{\Upsilon}_{nw_1}(B')$ and $\widetilde{\Upsilon}_{nw_2}(B')$ are homtopic equivalent. □

From Theorem 3.7.1, we can assume the network only has node type $(\mathbf{n}; n_O) = ((n); n_O)$ and $((2,1,\ldots,1); n_O)$ in the following discussion.

3.7.2 Retraction set

In this section we show, given a network nw and $B' \subset B^*_{nw}$, $\Upsilon_{nw}(B')$ is a strong deformation retraction of $\widetilde{\Upsilon}(B')$. The proof is based on Lemma 3.7.2

Theorem 3.7.2. *Given network nw and $B' \subset B^*_{nw}$, $\Upsilon_{nw}(B')$ is a strong deformation retraction of $\widetilde{\Upsilon}_{nw}(B')$. Hence homotopic equivalent.*

Proof. The proof is a simple application of Lemma 3.7.2. For convenience, denote

$B' = \{b'_k\}_{k=1}^K$ and $b'_k = (b'_{i,k})_{i=1}^N$. Define

$$X_{i,j} = \widetilde{\Upsilon}_{(\mathbf{n}_i;n_{O,i})}(\{b_{i,j}\}); Z_{i,j} = \Upsilon_{(\mathbf{n}_i;n_{O,i})'}(\{b_{i,j}\})$$
$$S_i = (0,\infty)^{1+2n_i+n_{O,i}}$$
$$F_i(\cdot,t) = \begin{cases} (3.31), (\mathbf{n}_i; n_{O,i}) = ((n); n_O) \\ (3.33), (\mathbf{n}_i; n_{O,i}) = ((2,1,\ldots,1); n_O) \end{cases}$$

where n_i is the number of inputs of node i or the sum of vector $\mathbf{n}_i$. By using Propositions 3.3.1, 3.3.2 and Lemma 3.7.2, we have $\Upsilon_{nw}(B')$ is a strong deformation retraction of $\widetilde{\Upsilon}_{nw}(B')$, hence homotopic equivalent. □

3.7.3 Linearized space

In this section we show, given a network nw and $B' \subset B^*_{nw}$, $\Upsilon_{nw}(B')$ and $\tilde{\mathcal{X}}_{nw}(B')$ are homotopic equivalent. The proof is based on Lemma 3.7.2.

Theorem 3.7.3. *Given network nw and $B' \subset B^*_{nw}$, $\Upsilon_{nw}(B')$ and $\widetilde{\mathcal{X}}_{nw}(B')$ are homotopic equivalent.*

Proof. The proof is a simple application of Lemma 3.7.1. For convenience, denote

$B' = \{b'_k\}_{k=1}^K$ and $b'_k = (b'_{i,k})_{i=1}^N$. Define

$$
\begin{aligned}
&X_{i,j} = \Upsilon_{(\mathbf{n}_i;n_{O,i})}(\{b'_{i,j}\}); X'_{i,j} = \widetilde{\mathcal{X}}_{(\mathbf{n}_i;n_{O,i})}(\{b'_{i,j}\}) \\
&S_i = (0,\infty)^{1+2n_i+n_{O,i}} \\
&S'_i = \begin{cases} (0,\infty)^{1+2n_i+n_{O,i}}, & (\mathbf{n}_i;n_{O,i}) \neq ((1,\ldots,1);n_{O,i}) \\ (-\infty,\infty)^{1+2n_i+n_{O,i}}, & (\mathbf{n}_i;n_{O,i}) = ((2,1,\ldots,1);n_{O,i}) \end{cases} \\
&f_i = \begin{cases} \mathrm{id}_{(0,\infty)^{1+2n_i+n_{O,i}}}, & (\mathbf{n}_i;n_{O,i}) \neq ((2,1,\ldots,1);n_{O,i}) \\ \tilde{f}, & (\mathbf{n}_i;n_{O,i}) = ((2,1,\ldots,1);n_{O,i}) \end{cases} \\
&g_i = \begin{cases} \mathrm{id}_{(0,\infty)^{1+2n_i+n_{O,i}}}, & (\mathbf{n}_i;n_{O,i}) \neq ((2,1,\ldots,1);n_{O,i}) \\ \tilde{g}, & (\mathbf{n}_i;n_{O,i}) = ((2,1,\ldots,1);n_{O,i}) \end{cases} \\
&F_i(\cdot,t) = \begin{cases} \mathrm{id}_{(0,\infty)^{1+2n_i+n_{O,i}}}, & (\mathbf{n}_i;n_{O,i}) \neq ((2,1,\ldots,1);n_{O,i}) \\ \tilde{F}(\cdot,t), & (\mathbf{n}_i;n_{O,i}) = ((2,1,\ldots,1);n_{O,i}) \end{cases} \\
&G_i(\cdot,t) = \begin{cases} \mathrm{id}_{(0,\infty)^{1+2n_i+n_{O,i}}}, & (\mathbf{n}_i;n_{O,i}) \neq ((2,1,\ldots,1);n_{O,i}) \\ \tilde{G}(\cdot,t), & (\mathbf{n}_i;n_{O,i}) = ((2,1,\ldots,1);n_{O,i}) \end{cases}
\end{aligned}
$$

where n_i is the number of inputs of node i or the sum of vector $\mathbf{n}_i$ and the $\tilde{f}, \tilde{g}, \tilde{F}, \tilde{G}$ are defined in Theorem 3.4.2 by (3.35, 3.36, 3.37, 3.38). By Theorem 3.4.2 and Lemma 3.7.1, we have $\Upsilon_{nw}(B')$ and $\widetilde{\mathcal{X}}_{nw}(B')$ are homotopic equivalent. □

3.7.4 Bounded space

In this section we proof $\tilde{\mathcal{X}}_{nw,\epsilon_1,\delta_1}(B')$ and $\tilde{\mathcal{X}}_{nw,\epsilon_2,\delta_2}(B')$ are homotopic equivalent under set inclusion map with appropriately chosen $\epsilon_1, d_1, \epsilon_2, d_2$, where $\epsilon_1 = (\epsilon_{1,1},\ldots,\epsilon_{1,N})$, $d_1 = (d_{1,1},\ldots,d_{1,N})$ and $\epsilon_2 = (\epsilon_{2,1},\ldots,\epsilon_{2,N})$, $d_2 = (d_{2,1},\ldots,d_{2,N})$. For network nw with N nodes, define

$$
\begin{aligned}
\epsilon_{nw} &= (\epsilon_{(\mathbf{n}_i;n_{O,i})})_{i=1}^N \\
d_{nw} &= (d_{(\mathbf{n}_i;n_{O,i})})_{i=1}^N.
\end{aligned}
\tag{3.43}
$$

Theorem 3.7.4. *Given network nw, $B' \subset B^*_{nw}$ and $\epsilon_{nw} > \epsilon_1 > \epsilon_2 > 0$ and $d_2 > d_1 > d_{nw} > 0$. Then there exits a homeomorphism*

$$f : \tilde{\mathcal{X}}_{nw,\epsilon_1,\delta_1}(B') \to \tilde{\mathcal{X}}_{nw,\epsilon_2,\delta_2}(B')$$

suhc that f is homotopic to the set inclusion map

$$ic : \tilde{\mathcal{X}}_{nw,\epsilon_1,\delta_1}(B') \to \tilde{\mathcal{X}}_{nw,\epsilon_2,\delta_2}(B')$$

Proof. The proof is a simple application of Lemma 3.7.3. For convenience, denote $B' = \{b'_k\}_{k=1}^K$ and $b'_k = (b'_{i,k})_{i=1}^N$. Define

$$\begin{aligned}
&X_{i,j} = \tilde{\mathcal{X}}_{(\mathbf{n}_i;n_{O,i}),\epsilon_{1,i},d_{1,i}}(\{b_{i,j}\}); X'_{i,j} = \tilde{\mathcal{X}}_{(\mathbf{n}_i;n_{O,i}),\epsilon_{2,i},d_{2,i}}(\{b_{i,j}\});\\
&S_i = \begin{cases} (0,\infty)^{1+2n_i+n_{O,i}}, & (\mathbf{n}_i;n_{O,i}) \neq ((2,1,\ldots,1);n_{O,i}) \\ (-\infty,\infty)^{1+2n_i+n_{O,i}}, & (\mathbf{n}_i;n_{O,i}) = ((2,1,\ldots,1);n_{O,i}) \end{cases}\\
&f_i = \check{f}\\
&F_i(\cdot,t) = \check{F}
\end{aligned}$$

where n_i is the number of inputs of node i or the sum of vector $\mathbf{n}_i$ and the $\check{f}, \check{F}$ are defined in Theorem 3.6.4 by (3.39, 3.40). Define $f = \prod_{i=1}^N$. By Theorem 3.6.4 and Lemma 3.7.3, we have f is a homeorphism, f and ic are homotopic equivalent. □

3.7.5 Homology group identification

Theorem 3.7.5. *For a given network nw, $B' \subset B^*_{nw}$, $0 < \epsilon_1 < \epsilon_{nw}$ and $0 < d_{nw} < d_1$, take $ic_{\epsilon_1,d_1} : \tilde{\mathcal{X}}_{nw,\epsilon_1,d_1}(B') \to \tilde{\mathcal{X}}_{nw}(B')$ be the set inclusion map. Then the induced map $ic_{\epsilon_1,d_1,*} : \mathrm{Hom}_\bullet(\tilde{\mathcal{X}}_{nw,\epsilon_1,d_1}(B')) \to \mathrm{Hom}_\bullet(\tilde{\mathcal{X}}_{nw}(B'))$ is an isomorphism over the singular homology groups.*

Proof. First we show it is a surjection. Suppose a cycle $c \in Z_k(\widetilde{\mathcal{X}}_{nw}(B'))$, then from Proposition 3.5.1 , we know there are $\epsilon_2, d_2 > 0$ such that $\epsilon_2 < \epsilon_1$ and $d_2 > d_1$ and

$c \in Z_k(\tilde{\mathcal{X}}_{nw,\epsilon_2,d_2}(B'))$. From Theorem 3.7.4 and define the set inclusion map

$$\mathrm{ic}_{\epsilon_1',d_1',\epsilon_2',d_2'} : \widetilde{\mathcal{X}}_{nw,\epsilon_1',d_1'},(B')) \to \widetilde{\mathcal{X}}_{nw,\epsilon_2',d_2'}(B')(B')),$$

we have the induced homomorphism between chain groups

$$\mathrm{ic}_{\epsilon_1',d_1',\epsilon_2',d_2',*} : \mathrm{Hom}_k(\tilde{\mathcal{X}}_{nw,\epsilon_1,d_1}(B'))) \to \mathrm{Hom}_k(\tilde{\mathcal{X}}_{nw,\epsilon_2,d_2}(B')))$$

is an isomorphism. It easy to see

$$\mathrm{ic}_{\epsilon_2,d_2,*} \circ \mathrm{ic}_{\epsilon_1,d_1,\epsilon_2,d_2,*} = \mathrm{ic}_{\epsilon_1,d_1,*}.$$

So we have $c' = \mathrm{ic}^{-1}_{\epsilon_1,d_1,\epsilon_2,d_2}(c) \in \mathrm{Hom}_k(\tilde{\mathcal{X}}_{nw,\epsilon_1,d_1}(B'))$ such that $\mathrm{ic}_{\epsilon_1,d_1,*}(c') = c \in \mathrm{Hom}_k(\widetilde{\mathcal{X}}_{nw}(B'))$.

Next we show it is an injection. Suppose $\mathrm{ic}_{\epsilon_1,d_1,*}(c) = 0$ for some $c \in Z_k(\tilde{\mathcal{X}}_{nw,\epsilon_1,d_1}(B'))$, then there is $b \in C_{k+1}(\widetilde{\mathcal{X}}_{nw}(B'))$ such that $\partial b = c$. From Proposition 3.5.1, we know there are $\epsilon_2, d_2 > 0$ such that $0 < \epsilon_2 < \epsilon_1$ and $d_2 > d_1 > 0$ and $b \in C_{k+1}(\tilde{\mathcal{X}}_{nw,\epsilon_2,d_2}(B'))$ or equivalently we have $c = 0 \in \mathrm{Hom}_k(\widetilde{\mathcal{X}}(B', \epsilon_2', d_2'))$. However $i_{\epsilon_1',d_1',\epsilon_2',d_2',*}(c) = c = 0 \in \mathrm{Hom}_k(\tilde{\mathcal{X}}_{nw,\epsilon_2,d_2}(B'))$ is an isomorphism, so we have $c = 0 \in \mathrm{Hom}_k(\tilde{\mathcal{X}}_{nw,\epsilon_1,d_1}(B'))$. □

3.8 Applications

From Theorem 3.7.5, for each node of type $(\mathbf{n}; n_O)$ in a network, we can choose $0 < \epsilon'_{(\mathbf{n};n_O)} < \epsilon_{(\mathbf{n};n_O)}, d_{(\mathbf{n};n_O)} < d'_{(\mathbf{n};n_O)}$ such that $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon',d'}(B')$, $B' \subset B^*(\mathbf{n}; n_O)$, is constant under isomorphism as a chain complex. For simplification, we can denote $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon',d'}$, as $\mathcal{X}_{(\mathbf{n};n_O)}$ if we have already computed the complex with some fixed $\epsilon'_{(\mathbf{n};n_O)}, d'_{(\mathbf{n};n_O)}$ under above restrictions. Then we can construct the chain complex $\widetilde{\mathcal{X}}_{nw,\epsilon',d'}$ from $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}$ as in (3.41) and also for simplification we deonte it as $\widetilde{\mathcal{X}}_{nw}$. In all examples of this section, we get In all the examples of this section, we have used FME to compute ϵ_{nw} and found the choice $\epsilon' = \frac{1}{1000} \leq \epsilon_{(\mathbf{n}_i,n_{O,i})}, d' = 10 \geq d_{(\mathbf{n}_i,n_{O,i})}$ is appropriate for each node type in all those networks. In another word we can choose $\epsilon' = (\frac{1}{1000})_{i=1}^N$, $d' = (10)_{i=1}^N$ in all the following examples in constructing the chain

complex $\mathcal{X}_{\epsilon',d',nw}$, or equivalently $\mathcal{X}_{nw}$.

3.8.1 Toggle switch

In the case of the toggle switch the DSGRN database indicates that the only parameter node at which the Morse graph indicates bistability is node 5. As all the defining inequalities of node 5 is linear when $\gamma_i = 1$, the set of parameter values at which DSGRN reports bistability is a contractible subset of $(0, \infty)^8$.

For all nodes complementary to node 5 the DSGRN database reports monostability. Then B' denote this set of these nodes. Using SageMath 8.5 ChainComplex function, we determine that

$$\mathrm{Hom}_k(\mathcal{X}_{nw}(B')) \cong \begin{cases} \mathsf{Z}_2 & \text{if } k = 0, 1 \\ 0 & \text{otherwise.} \end{cases}$$

.

3.8.2 Repressilator

In the case of the repressilator the DSGRN database indicates that the only parameter node at which the Morse graph indicates oscillation is node 13. As all the defining inequalities of node 13 is linear when $\gamma_i = 1$, the set of parameter values at which DSGRN reports no oscillation is a contractible subset of $(0, \infty)^{12}$.

For all nodes complementary to node 13 the DSGRN database reports oscillation. Then B' denote this set of these nodes. Using SageMath 8.5 ChainComplex function, we determine that

$$\mathrm{Hom}_k(\mathcal{X}_{nw}(B')) \cong \begin{cases} \mathsf{Z}_2 & \text{if } k = 0, 2 \\ 0 & \text{otherwise.} \end{cases}$$

3.8.3 Hysteresis of three nodes

In DSGRN, for a three nodes network, with nodes x_0, x_1, x_2, we can study the topology of the parameter regions, associated to the dynamics of x_1, x_2, which enables hysteresis path in the dynamics of x_0. Suppose the total number of parameter regions in x_1, x_2

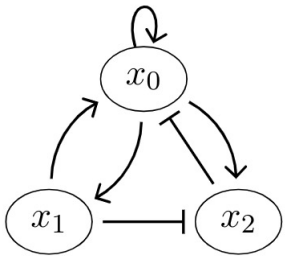


Figure 3.3: Regulatory network for top ranked hysteresis.

$\mathcal{X}^{(0)}$	$\mathcal{X}^{(1)}$	$\mathcal{X}^{(2)}$	$\mathcal{X}^{(3)}$	$\mathcal{X}^{(4)}$	$\mathcal{X}^{(5)}$	$\mathcal{X}^{(6)}$	$\mathcal{X}^{(7)}$
120	490	889	919	579	220	46	4

Table 3.2: Cell information of $\widetilde{\mathcal{X}}(B', \epsilon, d)$ over the x_1, x_2 parameter regions for network in Figure 3.3

parameter spaces is N and the number of selected parameter regions in x_1, x_2 parameter space, which can generate hysteresis in x_0, is N_1, we will call $\frac{N_1}{N}$ as the hysteresis robustness index. In all three nodes networks, we select one of the highest ranked network, Figure 3.3, with respect to the hysteresis index. Then we study those parameter regions of x_1, x_2 that can generate hysteresis in x_0 dynamics for the network. The corresponding hysteresis generating parameter regions over x_1, x_2 is given by DSGRN and we compute the homology group of the region as

$$\mathrm{Hom}_k(\mathcal{X}_{nw}(B')) \cong \begin{cases} \mathsf{Z}_2 & \text{if } k = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The cell information of $\mathcal{X}_{nw}(B')$ is given in Table 3.2.

All the code is given in DSGRN parameter region homology repository or url: `https://github.com/lunzhang1990/parameterHomology`

3.9 Methods

We created a database and a pipeline to compute homology. First we compute the database of the face lattice for $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon',d'}(B^*(\mathbf{n}; n_O))$ of some selected node types

$(\mathbf{n}; n_O)$ with fixed ϵ', d' such that $0 < \epsilon' < \epsilon'_{(\mathbf{n};n_O)}$, $0 < d_{(\mathbf{n};n_O)} < d'$. This database includes all the information, like chain groups and incidence number, of the cellular complex $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O),\epsilon',d'}(B^*(\mathbf{n}; n_O))$ over Z_2. As from Theorem 3.6.4, we know the these the computation results are independent of the ϵ', d', i.e. the cellular complexes are isormophic with each other, we denote the database as $\widetilde{\mathcal{X}}_{(\mathbf{n};n_O)}$ for simplification. We created a pipeline to compute homology. The pipeline takes DSGRN representing string of a network nw and some $B' \subset B^*_{nw}$ as input, uses the database to build up the face lattice or cellular complex over Z_2 of $\widetilde{\mathcal{X}}_{nw}(B)$, computes the homology of $\widetilde{\mathcal{X}}_{nw}(B)$. All the codes are developed under Python 3.x and SageMath 8.5 and are given in DSGRN parameter region homology repository or url: `https://github.com/lunzhang1990/parameterHomology`.

References

[1] https://github.com/lunzhang1990/parameterhomology.

[2] A. Loewer, E. Batchelor, G. Gaglia, and G. Lahav. Basal dynamics of p53 reveal transcriptionally attenuated pulses in cycling cells. *Cell*, pages 89–100, 2010.

[3] David L Applegate, William Cook, Sanjeeb Dash, and Daniel G Espinoza. Exact solutions to linear programming problems. *Operations Research Letters*, 35(6):693–699, 2007.

[4] Dennis S. Arnon, George E. Collins, and Scott McCallum. Cylindrical algebraic decomposition i: the basic algorithm. *SIAM Journal on Computing*, 13(4):865–877, 1 1984.

[5] Saugata Basu, Richard Pollack, and Marie Roy. Algorithms in Real Algebraic Geometry. pages 1–676.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*, 2004.

[7] Bree Cummins, Tomas Gedeon, Shaun Harker, Konstantin Mischaikow, Kafung Mok. Combinatorial representation of parameter space for switching systems. *Siam Journal on Applied Dynamical Systems*, 15(4):2176–2212, 2016.

[8] Graham Brightwell and Peter Winkler. Counting Linear Extensions is #P-complete. In *Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing*, STOC '91, pages 175–181, New York, NY, USA, 1991. ACM.

[9] Graham R Brightwell. The Number of Linear Extensions of Ranked Posets. Technical report, 2003.

[10] Christopher W. Brown. Improved projection for cylindrical algebraic decomposition. *Journal of Symbolic Computation*, 32(5):447–465, 2001.

[11] Christopher W. Brown. Constructing a single open cell in a cylindrical algebraic decomposition. *Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC*, pages 133–140, 2013.

[12] Christopher W. Brown and James H. Davenport. The complexity of quantifier elimination and cylindrical algebraic decomposition. *Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC*, pages 54–60, 2007.

[13] George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In H. Brakhage, editor, *Automata Theory and Formal Languages*, pages 134–183, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg.

[14] Bree Cummins, Tomas Gedeon, Shaun Harker, and Konstantin Mischaikow. Model rejection and parameter reduction via time series. *SIAM J. Appl. Dyn. Syst.*, 17(2):1589–1616, 2018.

[15] Bree Cummins, Tomas Gedeon, Shaun Harker, Konstantin Mischaikow, and Kafung Mok. Combinatorial Representation of Parameter Space for Switching Networks. *SIAM Journal on Applied Dynamical Systems*, 15(4):2176–2212, 2016.

[16] MB Elowitz and S Leibler. A synthetic oscillatory network of transcriptional regulators. *NATURE*, 403(6767):335–338, JAN 20 2000.

[17] T. Fine and J. Gill. The enumeration of comparative probability relations. *Ann. Prob.*, 4:667–673, 1976.

[18] Tomas Gedeon, Bree Cummins, Shaun Harker, and Konstantin Mischaikow. Identifying robust hysteresis in networks. *PLOS COMPUTATIONAL BIOLOGY*, 14(4), APR 2018.

[19] Jiří Gärtner, Bernd; Matoušek. In *Understanding and Using Linear Programming*, pages 81–104. Berlin: Springer, 2006.

[20] Wolfram Research, Inc. Mathematica, Version 11.

[21] D. Maclagan. Boolean Term Orders and the Root System B_n. *Order*, 15:279–295, 1999.

[22] Scott McCallum. An improved projection operation for cylindrical algebraic decomposition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Arti cial Intelligence and Lecture Notes in Bioinformatics)*, 204 LNCS:277–278, 1985.

[23] OEIS Foundation Inc. *The On-Line Encyclopedia of Integer Sequences*, 2020. `https://www.sagemath.org`.

[24] In preparation. Topology of High Dimensional Parameter Space in DSGRN.

[25] Jichang Sha and D. J. Kleitman. The number of linear extensions of subset ordering. *Discrete Mathematics*, 63(2-3):271–278, 1987.

[26] Shane Kepley , Konstantin Mischaikow , and Lun Zhang. Computing linear extensions for Boolean lattices with algebraic constraints. *arXiv:2006.02622v1*, Jun 23 2020.

[27] Richard P. Stanley. An introduction to hyperplane arrangements. In *Geometric combinatorics*, volume 13 of *IAS/Park City Math. Ser.*, pages 389–496. Amer. Math. Soc., Providence, RI, 2007.

[28] Herry Suprajitno and I Bin Mohd. Linear programming with interval arithmetic. *Int. J. Contemp. Math. Sciences*, 5(7):323–332, 2010.

[29] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 8)*. `https://www.sagemath.org`.

[30] James J. Collins Timothy S. Gardner, Charles R. Cantor. Construction of a genetic toggle switch in escherichia coli. *Nature*, 20 January 2000.

[31] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

[32] Robert J. Vanderbei. *Linear Programming Foundations and Extensions*, Third Edition.

[33] Virtanen, Pauli, and others. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[34] Ying Xin, Bree Cummins, and Tomas Gedeon. Multistability in the epithelial-mesenchymal transition network. *BMC Bioinformatics*, 21(1), 2 2020.