Stacked Graphs and Symmetric Groups: Modelling the Conformation Space of Nucleic Acids

by

Morgan Antony Roosenmaallen

B.Sc., The University of British Columbia, 2014
B.A., Lakehead University, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE COLLEGE OF GRADUATE STUDIES

(Interdisciplinary Studies)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

April 2017

Thesis Committee

The undersigned certify that they have read, and recommend to the College of Graduate Studies for acceptance, a thesis entitled *Stacked Graphs and Symmetric Groups: Modelling the Conformation Space of Nucleic Acids*, submitted by Morgan Antony Roosenmaallen in partial fulfilment of the requirements of the degree of Master of Science:

Javad Tavakoli, Barber School of Arts and Sciences
Supervisor, Professor

Paramjit Gill, Barber School of Arts and Sciences
Supervisory Committee Member, Professor

Yong Gao, Barber School of Arts and Sciences
Supervisory Committee Member, Professor

Robert Lalonde, Barber School of Arts and Sciences
University Examiner, Professor

Monday April 10 2017
Date Submitted to Grad Studies

Abstract

This work introduces a technique for approximating the conformation space of rigid and semi-rigid kinematic chains with finite inverse kinematic solutions using graph-like constructs called stacked graphs. The technique is developed in the context of nucleic acids, in particular ribonucleic acid (RNA), whose phosphate backbone can be modelled as a kinematic chain. Additionally, a process for identifying stable RNA conformations and likely conformational pathways is demonstrated. As a secondary result, a potentially novel relationship between stacked graphs and symmetric groups is uncovered and briefly investigated.

A Note on Notation

To facilitate ease of reading as well as to maintain consistency, a standardized naming and indexing notations will be used. Set hierarchies are denoted by specific cases and font styles

$$a \in A \in \mathcal{A} \in \mathbb{A} \in \mathfrak{A}$$

where the lowest level element is always indicated by a lower-case English character. Element indices are denoted by lower-case letters using a right-hand, 'ascending curl' format, beginning with the elements hierarchical location. For example

$$\begin{aligned} a_i \in A &\rightarrow a_i \in A \\ a_i \in A_j \in \mathcal{A} &\rightarrow a_i^j \in A_j \in \mathcal{A} \\ a_i \in A_j \in \mathcal{A}_k \in \mathbb{A} &\rightarrow {}^k a_i^j \in A_j^k \in \mathcal{A}_k \in \mathbb{A} \\ a_i \in A_j \in \mathcal{A}_k \in \mathbb{A}_l \in \mathfrak{A} &\rightarrow {}^k_l a_i^j \in {}^l A_j^k \in \mathcal{A}_k^l \in \mathbb{A}_l \in \mathfrak{A}. \end{aligned}$$

A single term, such as ${}^l A_j^k$, encodes not only the elements place in a hierarchy, but the hierarchy's total depth as well.

Two-valued scripts denote start-stop indices, as in the case of graph edges

$$e_{ij} \in E.$$

As numerical index values never exceed 9 in this work, commas between such indices will be avoided to condense notation. Thus, e_{17} should be read as $e_{1,7}$. Occasionally, indices of $(n-1)$ and the like will be used, producing terms like $e_{4(n-1)}$.

Superscripts in the absence of subscripts denote notions of size, such as a group composition of length n

$$\pi^n$$

and a cycle stack of length n and size m

$$C^{nm}.$$

Graphs with vertex sets V and edge sets E are denoted by $\mathcal{G} = (V, E)$. Walks, paths, and cycles are written in edge-edge format. For example

$$e_{13} - e_{32} - e_{27}. \tag{1}$$

All indices - vector or script - begin at 0. Vectors are accessed using $[\cdot]$ notation. Thus if $a = \left\langle 1, 7, -2, 11 \right\rangle$, then $a[1] = 7$ and $a[3] = 11$.

Lower-case Greek letters are reserved for group theoretic concepts (Chapter 3), except in the case of kinematic chain parametrization, where they are used to maintain consistency with reference works (Chapter 5). Upper-case Greek letters are reserved for general spaces.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank my supervisor Javad Tavakoli for his assistance and patience in what has been a lengthy endeavour, as well as Al Vaisius for introducing me to the research topic some four or five years ago.

Chapter 1

Introduction

Ribonucleic acid, or RNA, is a complex, polymeric biomolecule which performs a wide range of cellular functions. For example, messenger RNA (mRNA) synthesized from a DNA template is the mechanism for translating the nucleotide codons into the amino acid sequences of proteins (Figure 1.1(a)) [LNC93]. In this process, RNA molecules are structural components of ribosomes, which are the site of nascent protein synthesis; as well transfer RNA (tRNA) forms covalent bonds with amino acids to deliver them to ribosomes (Figure 1.1(b)) [LNC93]. In addition, there are a myriad of small regulatory RNAs that control gene expression. There also exists viral RNA (vRNA), which is used by a number of viruses for genetic information instead of DNA (Figure 1.1(c)) [LNC93].

RNA and DNA are similarly constructed from long sequences of nucleotides and shares many of the same chemical structures, including a phosphate backbone (Figure 1.2). The functionality of a given RNA strand is greatly influenced by its particular conformation, or the bending, twisting, and folding of the strand about itself and other biomolecules[1]. The phosphate backbone provides the primary level of constraints for RNA conformation. Although the backbone chain of nucleic acids consists entirely of sigma bonds, i.e. single covalent bonds, with full rotation about the bonds, the lengths of atomic bonds in the backbone, as well as the angles between them, place strong limits on how a single nucleotide moves. An important - and still open - question concerning RNA functionality is the exact nature of these limits with regards to neighboring polymeric units.

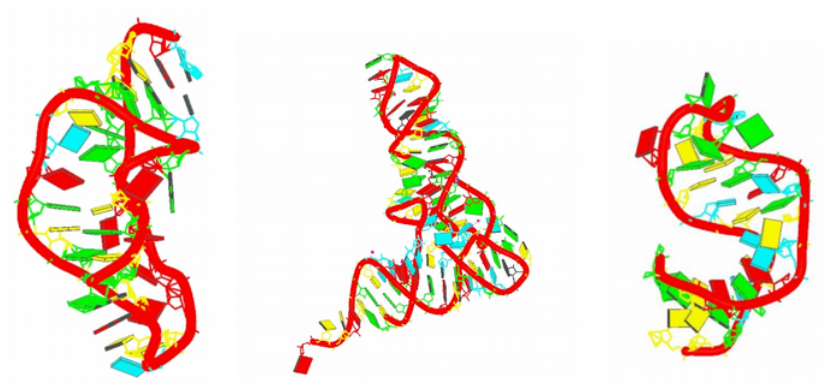
Considerable research efforts have gone into studying the conformation problem [Mak08][KHP11][WKM+08][MLL08]. Loosely stated, the issue is how the rotation of a nucleotide about the chemical bonds in its backbone relates to the physical space the backbone exists in. More formally, the challenge is to find the pre-image of

$$f : [0°, 360°)^6 \rightarrow \mathbb{R}^3 \times [0°, 360°)^3 \tag{1.1}$$

where f is a known function.

A number of approaches have been taken by researchers. Implicitly or explicitly, most of them are based on mathematical constructs known as *kinematic chains*,

[1] From discussions with Al Vaisius.

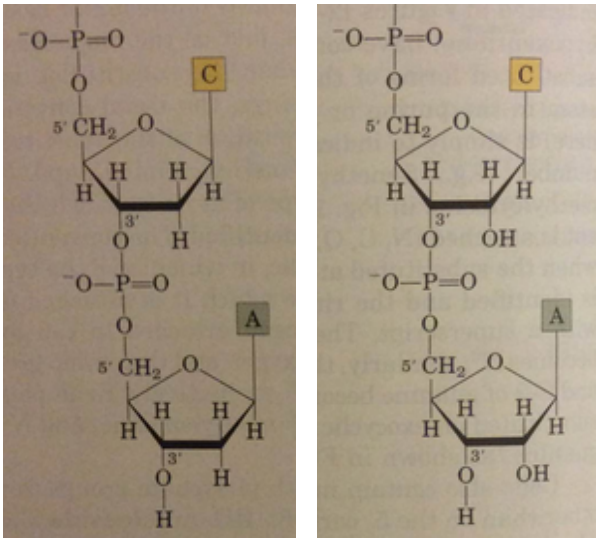


(a) (b) (c)

Figure 1.1: a) Fragment of mouse mammary tumour virus mRNA that causes a frame shift conformation in the mRNA , (b) yeast tRNA, and (c) a regulatory RNA transcribed from the human immunodeficiency virus, HIV, pro-viral DNA. Images from [Dat16], NDB IDs are 1RNK, TRNA03, and 1ANR, respectively.

models of rigid objects that dominate the field of robotics[Ang07]. Chemical bonds and angles are treated as fixed instead of the semi-flexible objects they are and the machinery of kinematic chains is used to generate descriptions of RNA's *conformation space*, or full range of possible bendings and twistings. f (Equation 1.1) becomes a description of the chain, usually represented as a product of transformation matrices.

In the case of individual backbones, [Mak08] developed a numerical technique for finding the pre-image f^{-1} for individual points in the conformation space (i.e. specific conformations), later generalizing it to multiple backbones using Monte Carlo methods [MCM11]. Similar results were produced by [PRT+07] for general chemical structures, though at the cost of computational speed. [KHP11] took the approach of reducing the dimensionality of the problem by replacing the six backbone torsion angles with two. Although the reduction supposedly captures the bulk of information for multi-nucleotide backbones, its nevertheless discards considerable available data. In a different vein, kinematic chains are used in [WKM+08] to correct *steric clashes*, the energetically-unfavourable collision of atoms resulting from certain conformations, in existing crystallographic data. There is at least one case where f^{-1} has been solved analytically, although in the context of proteins and



(a) (b)

Figure 1.2: Dinucleotide strands of (a) DNA and (b) RNA, both consisting of the same two nucleosides, cytosine and adenosine (colored boxes), the structures of which are not shown. The structure of the phosphate backbone, which for a single nucleotide consists of the atoms between successive phosphate atoms, inclusively, is the same for both DNA and RNA. The key difference between the two molecules is the presence of a hydroxyl group (OH) in the sugar ring of RNA. Figures from [LNC93].

at the expense of idealizing bond angle values [MLL08][2].

By defining a phosphate backbone as a special kind of redundant kinematic chain, the problem of solving the entirety of f^{-1} is theoretically possible using the technique of [PRT+07]. Indeed, this approach was originally pursued by myself, however the process was both inelegant and impractical: it more than double the number of variables by introducing seven new degrees of freedom[3].

The primary limitation of previous attempts to solve the RNA conformation problem is the very assumption which allowed them to use kinematic chains in the first place - that phosphate backbones are rigid objects. Chemical bonds are inherently flexible, undergoing compression and extension, while the mathematics modelling them is strictly rigid. The conformation problem for RNA can thus be considered as a special case of a more general problem in the theory of kinematic chains, namely how to incorporate semi-rigid objects in a rigid mathematics. It is the general problem that this work seeks to address, with its development conducted through the lens of RNA to provide a concrete example.

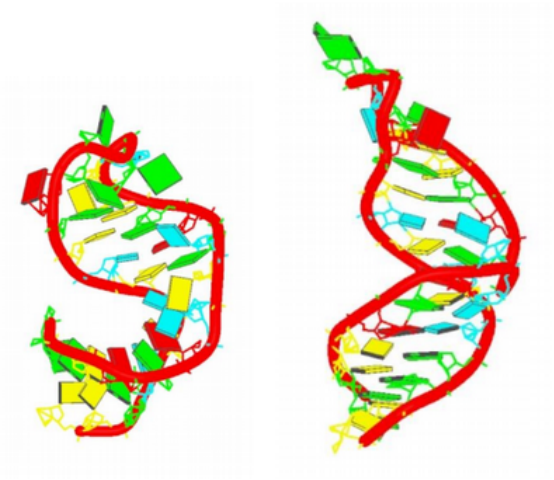
The main aim of this work, technically stated, is to approximate subsets of the conformation space of arbitrary kinematic chains which provide finite inverse kinematic (IK) solutions, or finite values for f^{-1} in Equation 1.1. Infinite solutions are treated as boundaries between these finite regions. The semi-rigidity of chain components is accounted for by repeated approximations of the range of each component's flexibility, then averaging them to form a new approximation.

While the technique applies, in theory, to the finite components of *any* type of kinematic chain, we'll be restricting our analysis almost exclusively to $6R$ chains, the kinematic equivalent of a linear molecule consisting of six sigma bonds, of which the phosphate backbone is one. In addition, the technique is predicated on the existence of high-speed algorithms, specifically, those that can quickly solve IK solutions. Fortunately, such algorithms exist for $6R$ chains [MC94] in general and phosphate backbones specifically [Mak08]. Should these algorithms ultimately prove too slow, however, or should they not exist for certain other types of chains, then the technique is not without merit - it simply anticipates the products of future research.

Two related sub-problems concerning RNA are also addressed. The first involves identifying stable conformations, or the set of energetically favourable states of a backbone, while the second involves identifying likely pathways between any two conformations. *Conformational pathways*, the continuous sequence of bendings and twistings that RNA takes to move from one conformation to another, have important applications in the field of pharmaceutical development. RNA-based

[2] A sequence of three amino acids has an associated kinematic chain similar to that of RNA.

[3] The process involved incorporating two spherical and one prismatic kinematic pairs.



(a) (b)

Figure 1.3: (a) A portion of HIV-1 TAR RNA in an inactive state, prior to binding to another biomolecule, and (b) the same, nearly identical portion in HIV-2 in an active state post-binding. Although the two structures are from different strains of HIV, the binding properties are similar [BW97]. Images from [Dat16], NDB IDs are 1ANR and 1AJU, respectively.

based diseases can be at least partially driven by conformational changes in RNA[4]. In the case of HIV, there exist two conformations for an important trans-activator regulating RNA, TAR, that is involved in the kinetics of HIV provirus transcription and resultant infectivity (Figure 1.3) [BW97]. Should the conformational pathway(s) between the two states be identified, it may be possible to design drugs that specifically target regions of the viral RNA that block the pathway(s), effectively inactivating it.

There are three research goals to this work:

1. Generate approximations for the finite inverse kinematic components of arbitrary kinematic chains, with specific focus on individual RNA phosphate backbones, that account for semi-flexible components;

2. Identify energetically favourable conformations for individual backbones;

3. Identify energetically favourable pathways between conformations for individual backbones.

In order to address these goals, it was necessary to construct novel graph-like structures called *stacked graphs*. Chapters 2 through 4 are dedicated to introducing these structures. Chapter 2 defines stacked graphs and the basics of their mathematics, including the notion of *feasible assignments*, or 'solutions' to stacked graphs that, in our case, approximate conformation spaces. The chapter culminates in Theorem 2.31, which provides necessary and sufficient conditions for assignment feasibility. Chapter 3 explores the close relationship between stacked graphs and the symmetric groups. While the chapter is not required to solve the conformation problem, it does provide a succinct notation for modelling it, as well as being a subject worthy of discussion in and of itself. Chapter 4 constructs a linear program for finding minimally weighted feasible assignments, or optimal approximations to a conformation space that satisfy the conditions of Theorem 2.31.

Chapters 5 through 8 focus on the techniques and mathematics required to write the phosphate backbone conformation space problem in terms of stacked graphs. Chapter 5 reviews the theory of kinematic chains to a level sufficient to understand the conformation space problem. Chapter 6 describes the methodology for sampling a conformation space, in addition to formally defining the conformation problem. These samples form a data set from which, in Chapter 7, a stacked graph will be constructed. The optimal feasible assignment to this stacked graph is shown to be an approximation to the conformation space problem for rigid kinematic chains. In Chapter 8 we define a number of metrics for determining the accuracy of our approximations. In addition, we locate stable conformations for individual phosphate

[4]From discussions with Al Vasius.

backbones as well as likely conformational pathways by converting stacked graphs into Markov chains. Finally, a new kind of 'averaged' stacked graph is defined to account for semi-flexible kinematic chains. Chapter 9 reviews the work and details primary topics that need to be addressed by future research.

As far as I am aware, the theory of stacked graphs is original to this work. I have been unable to find an equivalent to it in existing literature, although a previous discovery is certainly possible.

Chapter 2

Stacked Graphs

This chapter introduces stacked graphs. The first section provides a non-technical example of what is meant by 'approximating a conformation space' using a pair of curves, while presenting stacked graphs in an intuitive manner. Only a limited mathematical background is assumed in this section. The remaining portions of the chapter formally construct the mathematics of stacked graphs.

2.1 Motivating Example

Suppose we have a mapping f between two metric spaces[5]

$$f : X \to Y \tag{2.1}$$

where f is neither one-to-one nor onto[6]. We'll further suppose that we cannot find $f(X)$ explicitly, although we know that $f(X)$ consists of a finite number of connected components. We have, however, a subset S of X where $f(x_i)$ is known for all $x_i \in S$, and where each $f(x_i)$ consists of a finite number of elements in Y

$$f(x_i) = \{y_j^i | j = 0, 1, 2, \dots m, \ \forall x_i \in S_n\}. \tag{2.2}$$

Perhaps f is a real world function describing natural phenomenon we wish to understand and each $f(x_i)$ is an experimental sample. Or, perhaps f is an inverse image of a function and each $f(x_i)$ is a numerical approximation. In either case, we would like to know whether we can use the elements S to approximate the entirety of $f(X)$. The following example suggests that this is possible in some cases.

Let $f(X)$ be the stretched annulus of Figure 2.1(a), although we don't *know* it is. We'll take our sample point-set S in X to be

$$S := \{x_i | x_i < x_{i+1}, i = 0, 1, \dots, 12, \ x_i \in X\}. \tag{2.3}$$

By looking at Figure 2.1(a), the following two properties for are intuitively true if $|S|$ is sufficiently large and S provides good coverage of X. First, (almost) every

[5] Spaces that have a notion of 'distance'.

[6] A single point in X may map to many points in Y, but not ever point in Y can be arrived at from a point in X.

two successive values x_i and x_{i+1} will have images $f(x_i)$ and $f(x_{i+1})$, respectively, of equal size

$$|f(x_i)| = |f(x_{i+1})|. \tag{2.4}$$

Second, for (almost) every successive pair x_i and x_{i+1}, for every value y_j^i in $f(x_i)$ there is exactly one 'special' value y_k^{i+1} in $f(x_{i+1})$ such that, as the distance between x_i and x_{i+1} approaches zero, the distance between these special pairs of values also approaches zero. Technically stated,

$$|f(x_i) - f(x_{i+1})| \to \infty \implies \forall y_j^i \in f(x_i)\ \exists!\ y_k^{i+1} \in f(x_{i+1}) \ni |y_j^i - y_k^{i+1}| \to 0. \tag{2.5}$$

Figure 2.1 illustrates both properties. The 'almost' qualifier exists as there are cases where these conditions do not hold (i.e. $f(x_3)$ and $f(x_4)$). However, the number of cases is finite. The properties as they related to the conformation space problem are discussed in detail in Chapter 7. At present, we'll take them as given.

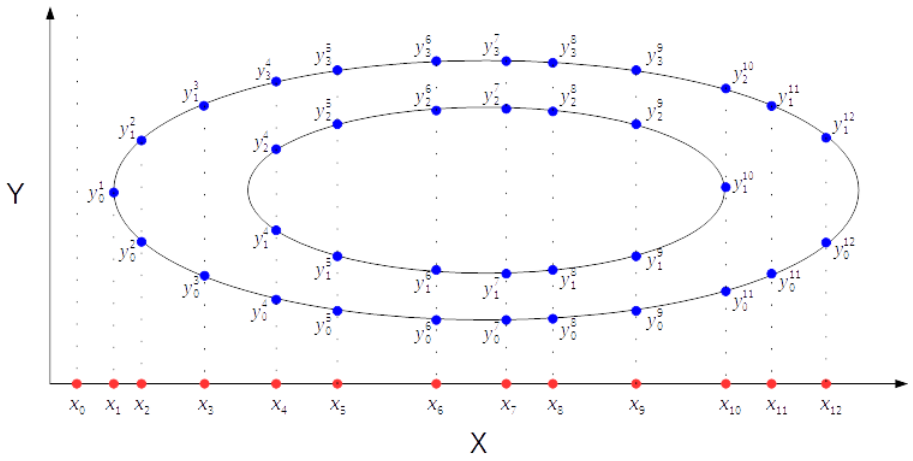
While we know that there exist pairs of values, one each from $f(x_i)$ and $f(x_{i+1})$, that approach each other in the limit, we don't know *which* pair. We only know that such pairs exist. As a result, we must consider every possible approach between successive image sets (Figure 2.2).

To this end, we view each image set as a 'stack' of vertices and the set of possible approaches between each successive image pair $f(x_i)$ and $f(x_{i+1})$ as a 'stack' of edges. These vertex and edge stacks form a succession of complete bipartite graphs (Figure 2.3(a)). Let V_i be the vertex associate with $f(x_i)$ and $E_{i(i+1)}$ be the edge stack between V_i and V_{i+1}. When $|f_{x_i}| = |f(x_{i+1})|$, an approach 'solution' is a perfect matching or set of disjoint edges [Fou92] mapping each vertex in V_i to a unique vertex in V_{i+1}. There are many possible matchings, or 'assignments'.

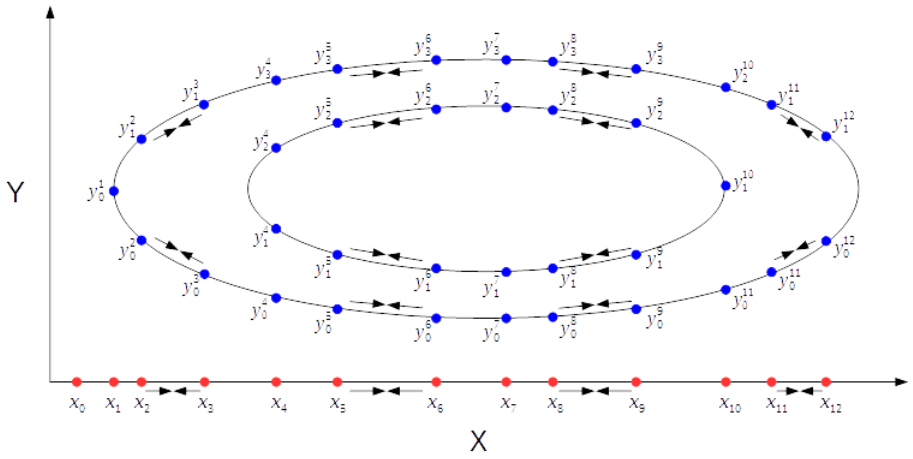
One possible'assignment for our sample set S is show in bold line in Figure 2.3(b). The assignment gives a reasonable approximation of $f(X)$, though there is an obvious error in the lower portion between V_6 andV_7. There is also the issue of what to make of edge stacks whose vertex stacks are of different sizes, which results when $|f_{x_i}| \neq |f(x_{i+1})|$, such as between V_3 and V_4, and V_{10} and V_{11}. These 'singularities' add a layer of complication that cannot be address in this preliminary section.

Determining the best set of approaches for an edge stack is related to the classic weighted bipartite matching, or *assignment*, problem [AMO93], and involves linear optimization. If S is large enough and covers X, then the optimal] assignment on S forms a piece-wise linear approximation of $f(X)$. For our choice of S above the best approximation of $f(X)$ is that of Figure 2.4.

The above annulus example is a mathematical introduction to the primary aim of this work. Instead of an annulus, the mapping f we wish to approximate, which

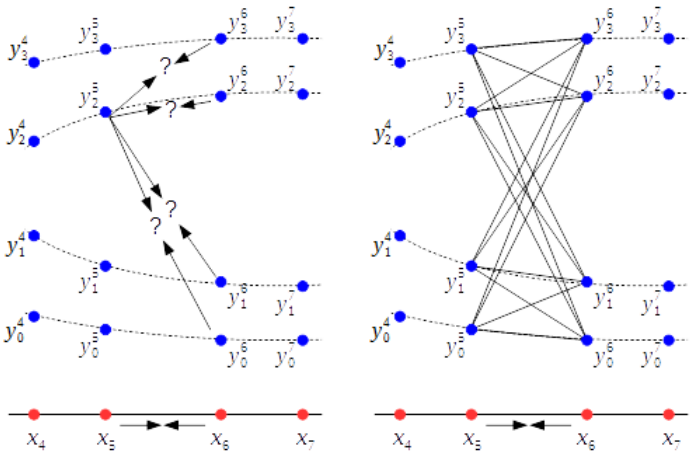


(a)



(b)

Figure 2.1: (a) The annulus $f(X)$ with a well-distributed collection of sample points x_i. Note that $f(x_0)$ is empty. Most x_i have the same number of image values $f(x_i)$ as x_{i+1}. (b) In each of these cases, one value in $f(x_i)$ is on the same portion of the annulus as exactly one value in $f(x_{i+1})$, so that, if x_i and x_{i+1} were to approach each other, the two corresponding values in $f(x_i)$ and $f(x_{i+1})$ would do likewise, as with $f(x_5)$ and $f(x_6)$.



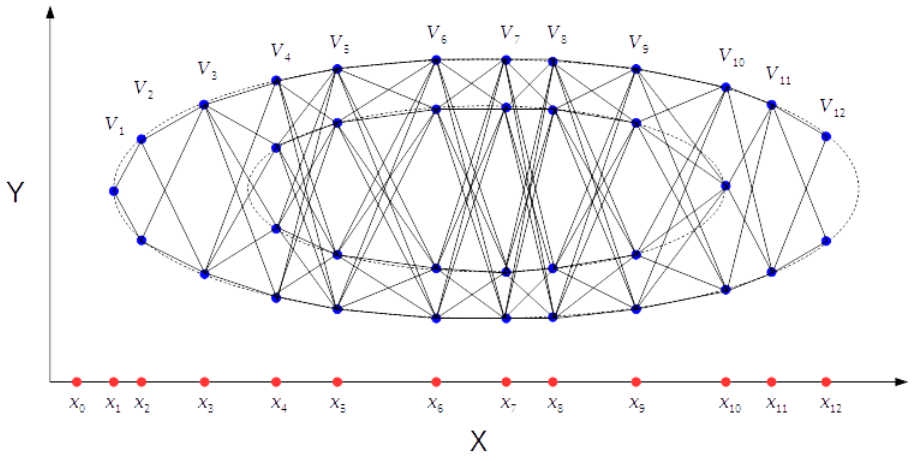
(a) (b)

Figure 2.2: (a) As $f(X)$ is unknown, its possible that y_2^5 approaches any one of y_0^6, y_1^6, y_2^6, and y_3^6. (b) We must therefore assume all possible approaches.

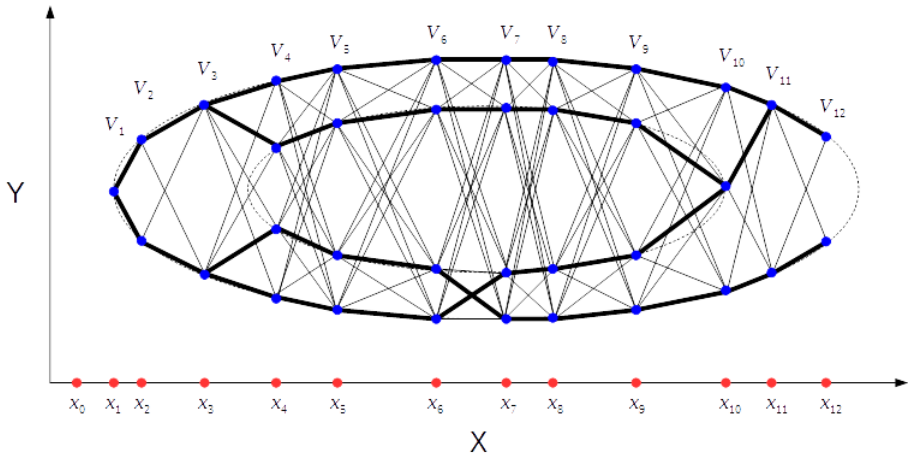
corresponds to the phosphate backbone conformation space, is actually the inverse image of a six-dimensional manifold. The limit behaviour as $f(x_i)$ approaches $f(x_{i+1})$ is expressed as shrinking neighbourhoods of open sets, while the singularities produced when $|f(x_i)| \neq |f(x_{i+1})|$ result, in part, when there is an intervening $x_k \in X$

$$x_i < x_k < x_{i+1} \tag{2.6}$$

where $|f(x_k)|$ is uncountable. The rest of this chapter is dedicated to developing the mathematics of stacked graphs necessary for our project. The optimization component for finding minimally weighted assignments is the subject of Chapter 4. The intervening Chapter 3 develops a method of writing stacked graphs, which are rather complicated, in simple algebraic terms. That the approximation in Figure 2.4 does not produce the disjoint components of $f(X)$ is a consequence of how singularities are handled. It's possible, however, to interpolate disjointedness in certain cases. This is covered briefly in Section 7.1.



(a)



(b)

Figure 2.3: (a) The conversion of each $f(x_i)$ to vertex stack V_i produces a complete bipartite graph between every successive pair V_i and V_{i+1}. (b) Creating an assignment (bold), or set of approach solutions, for every edge stack $E_{i(i+1)}$, or bipartite graph, produces a linear approximation to $f(X)$ (dashed).

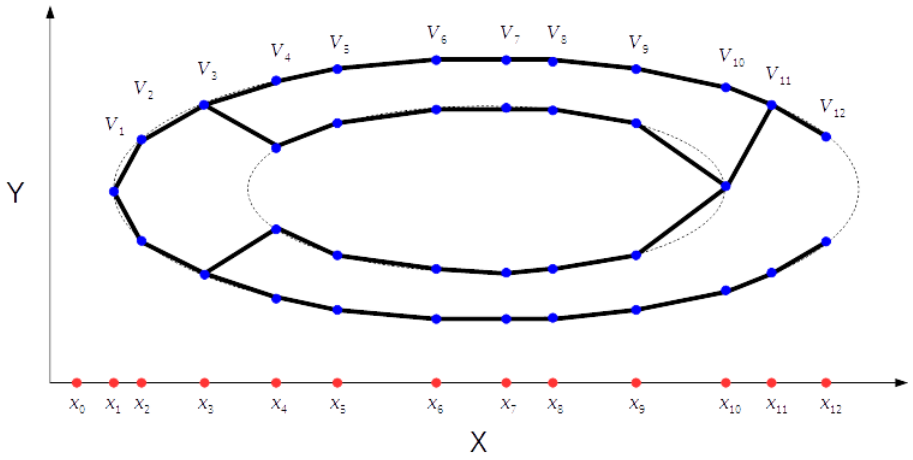


Figure 2.4: An optimal linear approximation to $f(X)$ based on S.

2.2 Constructing Stacked Graphs

Stacked graphs are a class of graphs formed by replacing each vertex with a set of indexed vertices and each edge with a complete bipartite graph between adjacent vertex stacks.

Definition 2.1 (Stacked Graphs)**.** A *vertex stack* is an indexed set of vertices

$$V_k := \{v_i^k | i = 0, 1, \dots, m\}. \tag{2.7}$$

The *size* of a vertex stack is $|V_k|$. An *edge stack* is the complete bipartite graph formed between the vertices of two vertex stacks, V_p and V_q[7]

$$E_{pq} := \{e_{ij}^{pq} | i = 0, 1, \dots, |V_p| - 1; j = 0, 1, \dots, |V_q| - 1\}. \tag{2.8}$$

Let $\mathcal{V} = \{V_k \mid k = 0, 1, \dots\}$ be a set of vertex stacks and $\mathcal{E} = \{E_{pq} \mid V_p, V_q \in \mathcal{V}\}$ be a set of edge stacks formed on $\mathcal{V}$. Then $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ is a *stacked graph*.

If $E_{pq} \in \mathbb{G}$, then V_p and V_q are *adjacent*. If $|V_p| = |V_q|$, then the E_{pq} is *asingular*. Otherwise, it's *singular*. The size of an asingular E_{pq}, denoted by $|E_{pq}|$, is $|V_p|$. If E_{pq} is singular and $|V_p| > |V_q|$, then V_p is the *collapsing* vertex stack and V_q is the *expanding* vertex stack of E_{pq}.

[7] A complete bipartite graph consists of two sets of vertices V_a and V_b where every vertex in V_a is adjacent to every vertex in V_b, but no edge exists between vertices within V_a or V_b [Bol79].

If $\mathbb{G}$ is composed entirely of singular edge stacks, then $\mathbb{G}$ is *singular*. The same applies in the *asingular* case. If $\mathbb{G}$ is neither singular nor asingular, it's *mixed*. If $\mathbb{G}$ is asingular and connected, then the *size* of $\mathbb{G}$, $|\mathbb{G}|$, is $|E_{pq}|$ for some $E_{pq} \in \mathbb{G}$[8].

Examples of singular and asingular edge stacks are shown in Figure 2.5.

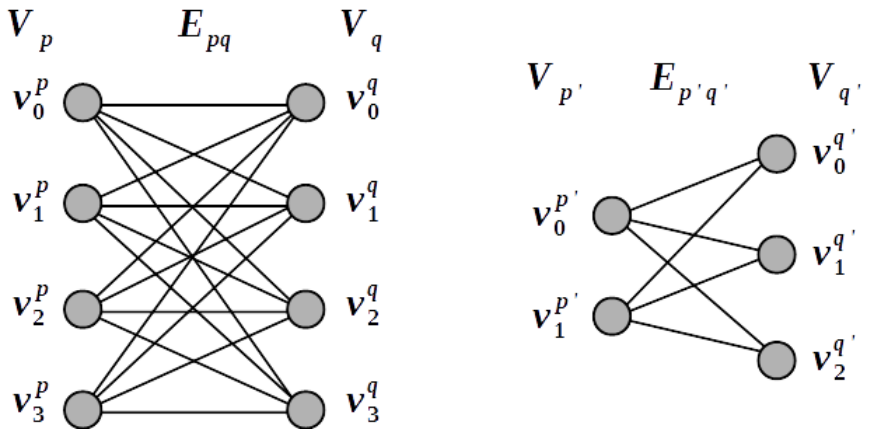


Figure 2.5: Adjacent stacks V_p and V_q form an asingular stack E_{pq} since $|V_p| = |V_q| = 4$. However, $E_{p'q'}$ is singular as $|V_{p'}| = 2$ and $|V_{q'}| = 3$.

Stacked graphs maintain all the properties of 'unstacked' graphs: every graph concept has an analogous stacked version.

Definition 2.2 (Stacked Analogues)**.** A *stacked analogue* is a graph structure in which every vertex stack V_k or edge stack E_{pq} is viewed as a single vertex v_k or edge e_{pq}.

For example, a *path stack* is a path composed of vertex and edge stacks

$$V_i - E_{ij} - V_j - E_{jk} - V_k - \dots$$

or

$$E_{ij} - E_{jk} - \dots .$$

The naming convention for stacked analogues is to either *postfix* a term with the word 'stack', or *prefix* it with 'stacked'. Both are legitimate.[9]

[8] It's possible that an asingular *disconnected* $\mathbb{G}$ contains vertex stacks of different sizes, as isolated vertex stacks of different sizes can exist in asingular $\mathbb{G}$.

[9] With the exception of stacked graphs - such as a stacked lattice graph - preference is given to using 'stack' over 'stacked'. Saying 'stacked path' and 'stacked cycle' is rather awkward.

As with stacked graphs in general, stacked analogues take one of three following forms, depending on the nature of the constituent edge stacks: *singular*, *asingular*, or *mixed*. If an analogue is asingular, its *size* is the size of its constituent edge stacks.

Due to their ubiquity, it will help to have a succinct notation for path and cycle stacks.

Definition 2.3 (Path and Cycle Stacks)**.** A path stack of length n is $\mathcal{P}^n$. If it's asingular size m, then its denoted by $\mathcal{P}^{nm}$. Cycle stacks are similarly defined for $\mathcal{C}^n$ and $\mathcal{C}^{nm}$.

It may at times be helpful to view stacked graphs as unstacked graphs, and vice versa.

Definition 2.4 (Un/Packing Graphs)**.** If a stacked graph $\mathbb{G}$ is viewed as an unstacked graph, that is, as a collection of vertices and edges, not vertex and edge stacks, then $\mathbb{G}$ is denoted by $\mathbb{G}_{\mathcal{G}}$, read as $\mathbb{G}$ *unpacked as* $\mathcal{G}$.

Likewise, if $\mathcal{G}'$ is graph isomorphic to some $\mathbb{G}_{\mathcal{G}}$, then $\mathcal{G}'$ can be viewed as the stacked graph, $\mathcal{G}_{\mathbb{G}}$, read $\mathcal{G}$ *packed as* $\mathbb{G}$.

Packing and unpacking can be restricted to paths, cycles, and their stacked analogues, such as $\mathcal{P}^{nm}_{\mathcal{G}}$ and $\mathcal{C}^{nm}_{\mathcal{G}}$.

The process of unpacking a stacked graph exposes the edges and nodes, while packaging an unstacked graph hides them. We mention packings only in passing as they're not necessary for this work. In general, packings are not unique and every $\mathcal{G}$ has at least one trivial packaging, formed by assigning to each vertex in $\mathcal{G}$ a vertex stack of size one. Figure 2.6 gives an example of different packings for a small graph.

Given any graph $\mathcal{G}$, it's possible to generate an asingular stacked graph $\mathbb{G}$ with the same underlying structure, and vice versa.

Definition 2.5 (Graph Collapse and Expansion)**.** The m^{th} *expansion of* $\mathcal{G}$, $\hat{\mathcal{G}}^m$, is the asingular stacked graph formed by replacing all vertices and edges in $\mathcal{G}$ with vertex and edge stacks of size m.

Likewise, if $\mathbb{G}$ is singular, asingular, or mixed, then the *collapse of* $\mathbb{G}$, $\check{\mathbb{G}}$, is the graph formed by replacing each vertex and edge stack in $\mathbb{G}$ with a single vertex and edge, respectively.

As with un/packings, collapses and expansions can be restricted to structures

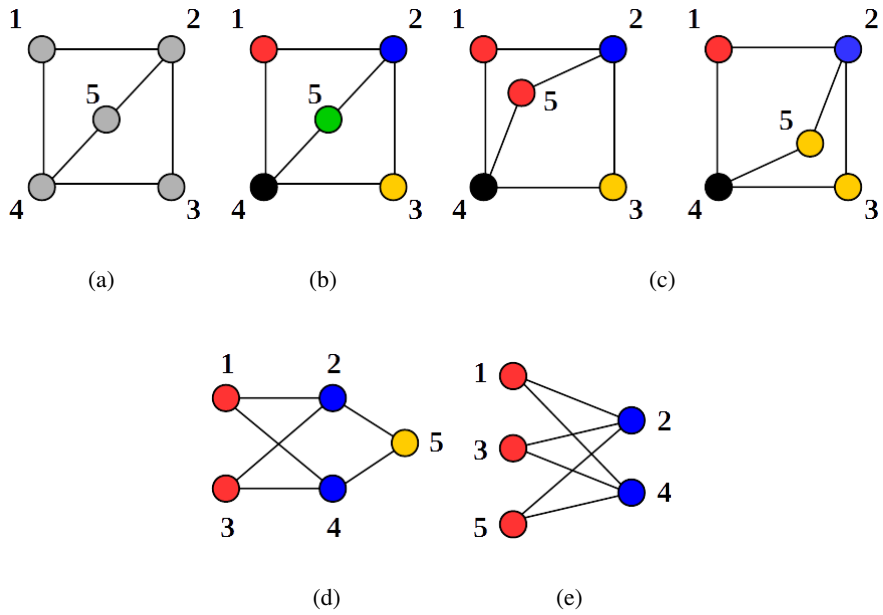


Figure 2.6: (a) An unstacked graph $\mathcal{G}$ and five packings $\mathcal{G}_{\mathbb{G}}$: (b) the trivial packing and packings producing (c) four, (d) three, and (e) two vertex stacks. Vertices of the same color are part of the same stack.

such as paths (stacks) and cycles (stacks)[10]

$$\begin{aligned} \check{\boldsymbol{P}}^{nm} &\to \boldsymbol{P}^{n} \\ \hat{\boldsymbol{P}}^{nm} &\to \boldsymbol{\mathcal{P}}^{nm} \\ \check{\boldsymbol{C}}^{nm} &\to \boldsymbol{C}^{n} \\ \hat{\boldsymbol{C}}^{nm} &\to \boldsymbol{C}^{nm} \end{aligned}$$

Expansions, collapses, and unpacking introduce five of ways of modifying or viewing the graphical structures $\mathcal{G}$ and $\mathbb{G}$: $\hat{\mathcal{G}}^{m}$, $\hat{\mathcal{G}}^{m}_{\mathcal{G}}$, $\mathbb{G}_{\mathcal{G}}$, $\check{\mathbb{G}}$, and $\check{\mathbb{G}}_{\mathcal{G}}$.

Lemma 2.6. *Every unstacked graph $\mathcal{G}$ can be treated as a stacked graph of size 1 by taking the first expansion, $\hat{\mathcal{G}}^{1}$. The processes can be reversed by unpacking $\hat{\mathcal{G}}^{1}$. That is*

$$\mathcal{G} = \hat{\mathcal{G}}^{1}_{\mathcal{G}}.$$

Unpacking is thus the inverse function of the first expansion.

Proof. The proof is obvious. □

First expansions $\hat{\mathcal{G}}^{1}$ provide a sort of 'wrapper' function for unstacked graphs, allowing us to treat them as being stacked. Analogues, on the other hand, let us treat stacked graphs as 'graph-like' objects in their own right: path stack connectivity is identical to path connectivity - except with path stacks. To limit confusion, however, we need to keep track of the words 'stack' and 'stacked'. Unless they're explicitly used, it's assumed that we're dealing with unstacked structures/properties.

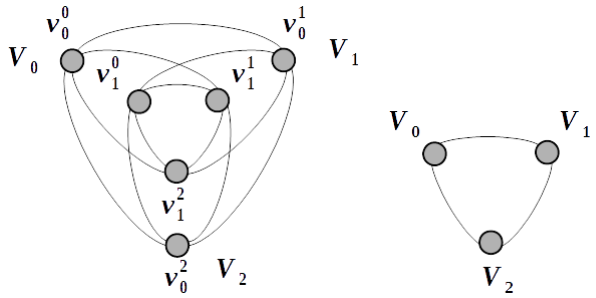
Drawing stacked graphs is difficult. Explicitly displaying edges e^{pq}_{ij} of some $\mathbb{G}$ while making the underlying structure of $\mathbb{G}$ obvious leads to extremely crowded diagram. This is true in even the simplest cases where $\mathbb{G}$ consists of few edge stacks, all of which are asingular and of small size. To facilitate ease of reading, two different graphical notations are used, depending on the needs at hand.

Definition 2.7 (Packed and Upacked Diagrams)**.** For any stacked graph $\mathbb{G}$, the *unpacked* diagram of $\mathbb{G}$ show the vertices and edges of each constituent vertex and edge stack, while the *packed* diagram shows only the vertex and edge stacks.

Figure 2.7 illustrates the difference between packed an unpacked diagrams. When large unpacked stacked graphs are used, the vertex labels will often be suppressed and vertices will be coloured instead, as in Figures 2.6 and 2.18.

Singular and asingular edge stacks are wildly different creatures. In the case of mixed $\mathbb{G}$, being able to work with its singular and asingular components separately will prove useful.

[10] This is one reason why consistent usage of style is important: $\hat{\boldsymbol{P}}^{nm}$ and $\boldsymbol{\mathcal{P}}^{nm}$ have two different meanings, even if though the former produces the latter.



(a) (b)

Figure 2.7: The (a) unpacked and (b) packed diagrams of $\mathcal{C}^{32}$.

Definition 2.8 (De/Singularization of $\mathbb{G}$)**.** Let the set of singular edge stacks of any $\mathbb{G}$ be $\hat{\mathcal{E}}$. Then $\mathring{\mathbb{G}}$ is the *desingularization of* $\mathbb{G}$, or the *asingular stacked graph induced by* $\mathbb{G}$

$$\mathring{\mathbb{G}} := (\mathcal{V}, \mathcal{E} - \hat{\mathcal{E}}). \tag{2.9}$$

Similarly, $\hat{\mathbb{G}}$ [11] is the *singularization of* $\mathbb{G}$, or the *singular stacked graph induced by* $\mathbb{G}$

$$\hat{\mathbb{G}} := (\mathcal{V}, \hat{\mathcal{E}}). \tag{2.10}$$

Fact 2.9. *If* $\mathbb{G}$ *is asingular, than* $\mathring{\mathbb{G}} = \mathbb{G}$ *and* $\hat{\mathbb{G}}$ *has no edge stacks. If* $\mathbb{G}$ *is singular, then* $\hat{\mathbb{G}} = \mathbb{G}$ *and* $\mathring{\mathbb{G}}$ *has no edge stack.*

The connected component stacks of $\mathring{\mathbb{G}}$ and the edge stacks of $\hat{\mathbb{G}}$ are special.

Definition 2.10 (Regions)**.** The set of connected component stacks of $\mathring{\mathbb{G}}$, $\mathbb{G}_{\mathcal{R}}$, are the *regions of* $\mathbb{G}$

$$\mathbb{G}_{\mathcal{R}} := \{\mathcal{R}_i \mid i = 0, 1, \dots\}. \tag{2.11}$$

All vertex stacks $V_k \in \mathcal{R}_i$ are of the same size[12], which is also the *size* of $\mathcal{R}_i$, $|\mathcal{R}_i|$.

Definition 2.11 (Regional Boundaries)**.** If $\mathcal{R}_r$ and $\mathcal{R}_t$ are two regions of $\mathbb{G}$ such that $|\mathcal{R}_r| \neq |\mathcal{R}_t|$, with $V_p \in \mathcal{R}_r$, $V_q \in \mathcal{R}_t$ and $E_{pq} \in \mathbb{G}$, then $E_{pq} \in \hat{\mathbb{G}}$. The set of all such edge stacks for $\mathcal{R}_r$ and $\mathcal{R}_t$ is the *(singular) boundary between* $\mathcal{R}_r$ *and* $\mathcal{R}_t$, denoted by $\hat{\mathcal{E}}_{rt}$. If $|\mathcal{R}_r| > |\mathcal{R}_t|$, then $\mathcal{R}_r$ is the *collapsing* region and $\mathcal{R}_t$ is the *expanding* region.

[11] Concerning the accents used, $\mathring{}$ resembles two disjoint regions, while $\hat{}$ resembles the negative space of $\mathring{}$.

[12] This is easily derivable and need not be part of the definition.

If $\hat{\mathcal{E}}_{rt}$ is non-empty, then $\mathcal{R}_r$ and $\mathcal{R}_t$ are *adjacent*, otherwise they are *not adjacent*.

Regions are connected component *stacks*, not components. If $\mathbb{G}$ is singular, then $\mathring{\mathbb{G}}_{\mathcal{G}}$ has no edges and every node is disconnected. $\mathring{\mathbb{G}}_{\mathcal{G}}$ would thus have a large number of components, but $\mathring{\mathbb{G}}$ would have only $|\mathcal{V}|$ component stacks, or regions. Asingular cycle stacks are a core feature of stacked graphs and will be used to form the stacked analogue of a cycle basis.

Definition 2.12 (Fringe and Core of $\mathbb{G}$)**.** Let E_{pq} be any asingular edge stack in $\mathbb{G}$. If E_{pq} belongs to an asingular cycle stack, then E_{pq} is a *cyclic* edge stack. Otherwise E_{pq} is an *acyclic* edge stack.

The stacked sub-graph of $\mathbb{G}$ composed of all acyclic edges stacks of $\mathbb{G}$ is the *fringe* $\tilde{\mathbb{G}}$ *of* $\mathbb{G}$, while the stacked sub-graph composed of all cyclic edge stacks is the *core of* $\mathring{\mathbb{G}}$[13].

Fact 2.13. $\tilde{\mathbb{G}}$ *is a tree or forest.*

Fact 2.13 follows immediately from the definition of $\tilde{\mathbb{G}}$.

Lemma 2.14 (Fringe, Core, and Singular Edges of $\mathbb{G}$)**.** *For any* $\mathbb{G}$*, the three sub-graph stacks* $\tilde{\mathbb{G}}$*,* $\mathring{\mathbb{G}}$*, and* $\hat{\mathbb{G}}$ *partition the edge stacks* $\mathcal{E}$ *of* $\mathbb{G}$ *into disjoint sets.*

Proof. Every edge stack is singular, asingular and part of a cycle stack, or asingular and not part of a cycle stacks. □

Thus there exists an equivalence relation on $\mathcal{E}$ of $\mathbb{G}$ with three classes defined by $\tilde{\mathbb{G}}$, $\mathring{\mathbb{G}}$, and $\hat{\mathbb{G}}$.

2.3 Assignments and Feasibility

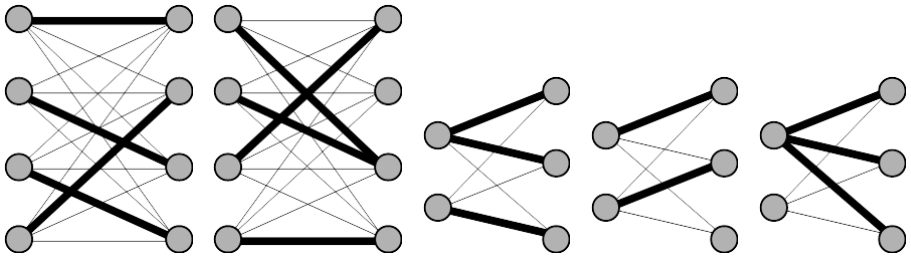
Assignments are essentially special kinds of network flow solutions to $\mathbb{G}_{\mathcal{G}}$.

Definition 2.15 (Edge Stack Assignment)**.** An *assignment A on* E_{pq}

$$A_{pq} := A(E_{pq})$$

is an onto mapping from the collapsing vertex stack of E_{pq} to the expanding one. If E_{pq} is asingular, then the mapping is one-to-one and onto.

[13] Concerning the accents used, $\circ$ resembles a cycle while $\sim$ resembles a path.



(a) (b) (c) (d) (e)

Figure 2.8: Asingular edge stacks assignments (bold lines) that are (a) valid and (b) invalid. For singular edge stacks, the number of assignment edges is equal to the largest size of the two vertex stacks. A (c) valid assignment ensures that every vertex in one stack is adjacent to at least one vertex in the other while (d) invalid singular assignments may have insufficient edges and/or (e) incomplete adjacency.

The *flow value*[14] of e_{ij}^{pq} under A_{pq} is denoted by x_{ij}^{pq}. Unless otherwise stated, flow values are binary

$$x_{ij}^{pq} \in \{0, 1\}. \tag{2.12}$$

If A_{pq} maps v_i^p to v_j^q, then $x_{ij}^{pq} = 1$ and $x_{ij}^{pq} = 0$ otherwise.

Let $|V_p| \geq |V_q|$. The i^{th} *edge flow of* A_{pq}, denoted by x_i^{pq}, corresponds to the edge incident on v_i^p whose flow value is non-zero[15].

Figure 2.8 gives an example of valid and invalid assignments for both singular and asingular edge stacks.

Definition 2.16 (Stacked Graph Assignment)**.** An assignment A on $\mathbb{G}$, $A(\mathbb{G})$, provides an assignment A_{pq} for every $E_{pq} \in \mathbb{G}$.

If $\mathbb{H}$ is a sub-stacked graph of $\mathbb{G}$ and A an assignment on $\mathbb{G}$, then $A(\mathbb{H})$ is the restriction of A to $\mathbb{H}$.

We'll often treat assignments as unstacked graphs in their own right. When referring to 'an edge of A_{pq}', we actually mean an edge in $A(E_{pq})$ with non-zero flow. This allows us to write things like

$$e_{ij}^{pq} \in A_{pq} \tag{2.13}$$

[14]Flow values form solutions to network flow problems [AMO93].

[15]By the definition of an assignment, such an edge always exist and exists.

which, though technically incorrect, provides a useful shorthand. Diagramatically, assignments will always be indicated as bold edges.

Assignments on E_{pq} are related to the classic assignment problems on bipartite graphs [AMO93]. Similarly, assignments on $\mathbb{G}$ can be thought of as solutions to a collection of inter-related assignment problems.

Not all assignments are created equal, however.

Definition 2.17 (Assignment Feasibility)**.** Let A be an assignment on $\mathbb{G}$. Then $A(\mathbb{G})$ is *feasible* if, for every $V_p \in \mathbb{G}$, the vertices of V_p are path disjoint in $A(\breve{\mathbb{G}})$. Otherwise, $A(\mathbb{G})$ is *infeasible*.

Equivalently, $A(\mathbb{G})$ is feasible if, for any path connected vertices v_i^p and v_j^p in $A(\mathbb{G})$, every path connecting them contains at least one edge belonging to a singular edge stack.

The remainder of this section is dedicated to proving necessary and sufficient conditions for assignment feasibility, culminating in Theorems 2.31 and 2.32. The process begins by looking at the application of assignments to path stacks.

Theorem 2.18 (Paths of a Path Stack)**.** *For any assignment A on an asingular path stack $\mathcal{P}^{nm}$, $A(\mathcal{P}^{nm})$ consists of exactly m disjoint paths of length n. If $\mathcal{P}^{nm}$ begins at V_a and ends at V_b, then each path of length n in $A(\mathcal{P}^{nm})$ begins at a vertex v_i^a and ends at a vertex v_j^b. Furthermore, the maximum length of a path in $A(\mathcal{P}^{nm})$ is n.*

Proof. By definition, for each E_{pq} in $\mathcal{P}^{nm}$, A_{pq} consists of m disjoint edges, or paths of length 1. For any adjacent pair E_{pq} and E_{qr} in $\mathcal{P}^{nm}$, every edge $e_{ij}^{pq} \in A_{pq}$ is adjacent to exactly one edge e_{jk}^{qr}. Thus an assignment on any path stack of length 2

$$E_{pq} - E_{qr}$$

consists of m disjoint paths of length 2. By induction on the length of $\mathcal{P}^{nm}$, $A(\mathcal{P}^{nm})$ consists of m disjoint paths of length n. □

Two corollaries follow from Theorem 2.18.

Corollary 2.19. *All assignments A on any asingular $\mathcal{P}^{nm}$ are feasible.*

Proof. The proof follows from that of Theorem 2.18 by induction on n and m. □

Corollary 2.20. *The disjoint paths P_i of $A(\mathcal{P}^{nm})$ for any A are isomorphic to $\mathcal{P}^{nm}$ and form a disjoint vertex cover of $\mathcal{P}^{nm}_{\mathcal{G}}$.*

Proof. By definition of an assignment on asingular edge stacks, every $v_i^p \in V_p$ maps to exactly one $v_j^q \in V_q$. Thus e_{ij}^{pq} is isomorphic to E_{pq} and each vertex in V_p and V_q is covered by some edge in A_{pq}. By induction on the edge stacks, every path $P_i \in A(\mathcal{P}^{nm})$ is isomorphic to $\mathcal{P}^{nm}$ and form a covering of the vertices in $\mathcal{P}^{nm}_{\mathcal{G}}$. □

Figure 2.9 gives an example of a feasible assignment on $\mathcal{P}^{44}$. From it, it's easy to visualize the proof of Corollary 2.20.

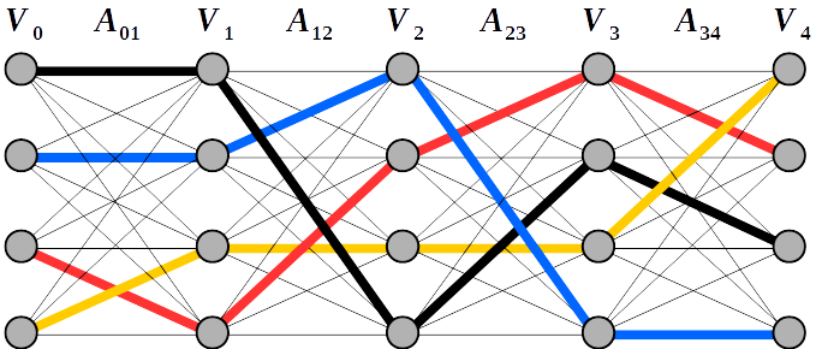


Figure 2.9: Any assignment A on a path stack $\mathcal{P}^{44}$ will create 4 paths of length 4.

We now turn to maintaining assignment feasibility on more complicated graphs, starting with cycle stacks.

Consider an asingular, 3-cycle stack, $\mathcal{C}^{32}$, formed by V_0, V_1, and V_2. Assignment $A_0(\mathcal{C}^{32})$ (Figure 2.10(a)) is infeasible as v_0^0 and v_1^0 are path connected by

$$v_0^0 - v_0^1 - v_1^2 - v_1^0. \tag{2.14}$$

In fact, *all* vertices are path connected since $A_0(\mathcal{C}^{32})$ describes a vertex cycle cover consisting of a single cycle. $A_1(\mathcal{C}^{32})$, however, is feasible (Fig 2.10(b)). The two connected components of $A_1(\mathcal{C}^{32})$ each contain exactly one vertex from each vertex stack.

$A_1(\mathcal{C}^{32})$ suggests a necessary condition for assignment feasibility of cycle stacks.

Lemma 2.21. *If A is a feasible assignment on asingular $\mathcal{C}^{nm}$, then $A(\mathcal{C}^{nm})$ forms a disjoint cycle cover of the vertices of $\mathcal{C}_{\mathcal{G}}^{nm}$.*

Proof. The proof relies on the asingular nature of $\mathcal{C}^{nm}$. It follows from the definition of edge stack assignments (Definition 2.15) that, for any two adjacent edge stacks E_{pq} and E_{qw} in $\mathcal{C}^{nm}$, every v_j^q is adjacent on exactly two other vertices v_i^p and v_k^w in $A(\mathcal{C}^{nm})$. This implies that every vertex belongs to a cycle[16]. By assumption, $A(\mathcal{C}^{nm})$ is feasible, so that intra-vertex stack vertices are path disconnected. Thus, all vertices in a given vertex stack belong to disjoint cycles. $A(\mathcal{C}^{nm})$ is therefore a disjoint cycle cover of the vertices of $\mathcal{C}_{\mathcal{G}}^{nm}$. □

[16] If they did not, then they would necessarily belong to a path. However, paths of non-zero length consist of at least two vertices incident on only a single edge.

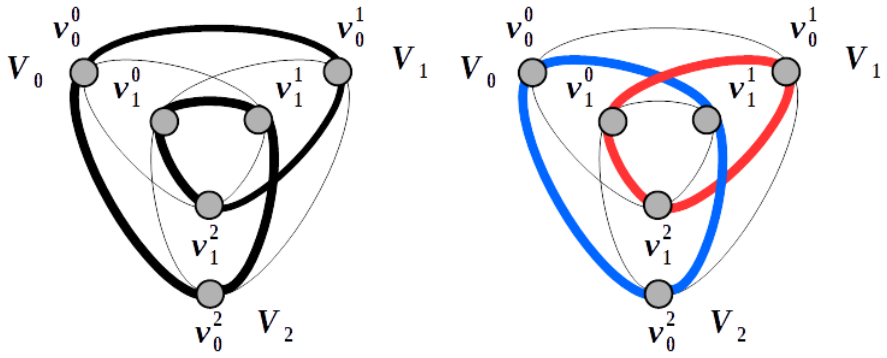


Figure 2.10: (a) $A_0(C^{32})$ does not enforce path disjointedness of intra-stack nodes while (b) $A_1(C^{32})$ does.

The converse is not true. Figure 2.11 gives an example of a disjoint cycle cover on C^{42} which is neither feasible nor an assignment.

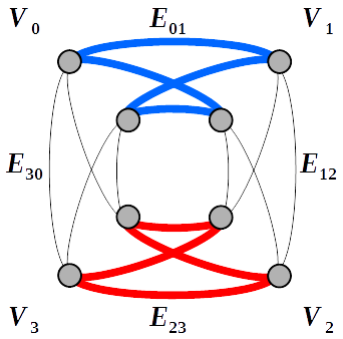


Figure 2.11: A disjoint cycle cover of C_G^{42} that does not form a feasible assignment.

Lemma 2.22. *If A is feasible on some C^{nm}, then there are m disjoint cycles in $A(C^{nm})$, each of length n.*

Proof. By definition of a cycle, a cycle stack is a path stack that starts and ends on the same vertex stack V_k. Viewed as a path stack, by Theorem 2.18 there exist m disjoint paths of length n in $A(C^{nm})$, each starting and ending in V_k. However,

since $A(C^{nm})$ is feasible, it is necessarily the case that each path starts and ends on the same vertex in V_k in order to prevent path connectedness between the nodes of V_k. Thus, each path is a cycle and disjoint. Therefore, there are m disjoint cycles of length n in $A(C^{nm})$. □

Theorem 2.22 leads to a key theorem about asingular cycle stacks and their feasible assignments.

Theorem 2.23 (Disjoint Cycles of Feasible Asingular Cycle Stack Assignments)**.** *Let A be an assignment on asingular C^{nm}. Then, $A(C^{nm})$ is feasible if and only if it forms a disjoint cycle cover of the vertices of $C_{\mathcal{G}}^{nm}$, consisting of m cycles, each of length n.*[17]

Proof. For the forward portion, see proofs for Lemmas 2.21 and 2.22. The proof for the backwards portion, that an assignment which produces an appropriate disjoint cycle cover is necessarily feasible, is as follows.

Let $\mathcal{P}^{(n-1)m}$ be the asingular path stack

$$\mathcal{P}^{(n-1)m} := E_{01} - E_{12} - \cdots - E_{(n-2)(n-1)} \tag{2.15}$$

and let A be some assignment on it. Introduce a new edge stack $E_{(n-1)0}$ and append it to $\mathcal{P}^{(n-1)m}$ to create the asingular cycle stack C^{nm} (Figure 2.12). In addition, introduce an assignment $A_{(n-1)0}$ onto $E_{(n-1)0}$.

Let $A(C^{nm})$ be the extension of A applied to $\mathcal{P}^{(n-1)m}$ by incorporating $A_{(n-1)0}$. In other words, $A(\mathcal{P}^{(n-1)m})$ is the proper sub-graph of $A(C^{nm})$ formed by removing the edges of $A_{(n-1)0}$. By Corollary 2.19, $A(\mathcal{P}^{(n-1)m})$ is feasible, meaning that the feasibility of $A(C^{nm})$ is dependent on $A_{(n-1)0}$.

From Theorem 2.18 we know that $A(\mathcal{P}^{(n-1)m})$ consists of m disjoint paths of length $n-1$. Without loss of generality, choose path P_i. Let v_a^0 and v_b^{n-1} for some $a, b \in \{0, 1, \ldots, (m-1)\}$ be the first and last nodes of P_i, respectively. By Definition 2.15, v_b^{n-1} is adjacent to exactly one vertex in V_0. If v_b^{n-1} is not adjacent to v_a^0, it forms a path with a length of at least $2n-1$ (Figure 2.13(a)) and includes two vertices in V_0, violating the assumption of independence. Thus v_b^{n-1} is adjacent to v_a^0, which forms a cycle C_i of length n containing only a single vertex from each vertex stack (Figure 2.13(b)).

As each cycle C_i contains exactly one vertex from each vertex stack, and as each vertex stack is of size m, it follows that there are m disjoint cycles of length n.

Therefore, if $A(C^{nm})$ is a disjoint cycle cover of the vertices of $C_{\mathcal{G}}^{nm}$ consisting of m cycles of length n, $A(C^{nm})$ is feasible. □

[17] We're assuming the disjoint cover cycle is produced by an assignment. We know there exist disjoint cover cycles for some C^{nm} consisting of m cycles of length n that do not form assignments (Figure 2.11).

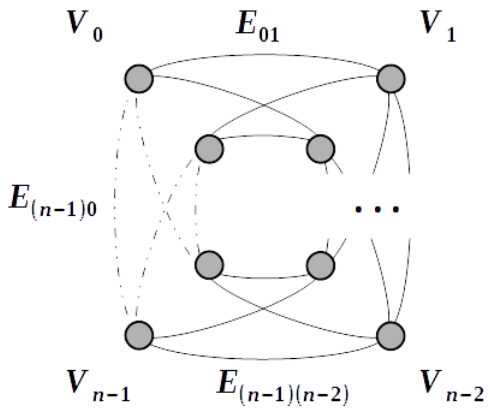


Figure 2.12: $\mathcal{P}^{(n-1)m}$ extended to C^{nm} by the inclusion of $E_{(n-1)0}$, with $m = 2$.

There are two special cases for assignment feasibility on C^{nm}, namely when $n = 1$ and $n = 2$. These are addressed in Section 2.4.

Corollary 2.24. *If A is a feasible assignment on any asingular C^{nm}, then the disjoint cycles C_i of $A(C^{nm})$ are isomorphic to C^{nm}.*

Proof. See proof for Corollary 2.20. □

Theorem 2.23 provides necessary and sufficient conditions for assignment feasibility on arbitrary asingular cycle stacks. This leads immediately to necessary and sufficient conditions for assignment feasibility on any asingular $\mathbb{G}$.

Theorem 2.25 (Feasibility of $\mathbb{G}$)**.** *For any assignment A on $\mathbb{G}$, $A(\mathbb{G})$ is feasible if and only if, for every asingular cycle stack C^{nm} in $\mathbb{G}$, $A(C^{nm})$ is feasible.*

Proof. For the forward component, $A(C^{nm})$ is a sub-graph of $A(\mathbb{G})$, so that any two vertices v_i^p and v_j^q that are path disconnected in $A(\mathbb{G})$ are necessarily so in $A(C^{nm})$.

For the backward component, every vertex stack belongs to a path and/or cycle stack. As feasibility is defined on asingular path and cycle stacks alone, we need only consider vertex stacks in them. If V_i is contained in one or more cycle stacks, then, by assumption, the vertices in V_i are disjoint under A. If V_i is not in some cycle stack, then it is necessarily in one or more path stacks. By Corollary 2.19, all $A(\mathcal{P}^{nm})$ are feasible, and by assumption, all $A(C^{nm})$ are feasible. Therefore $A(\mathbb{G})$ is feasible. □

The conditions of Theorem 2.25 conditions are imposing: it requires that *every* asingular cycle stack be checked for feasibility under A, which is tedious at best and

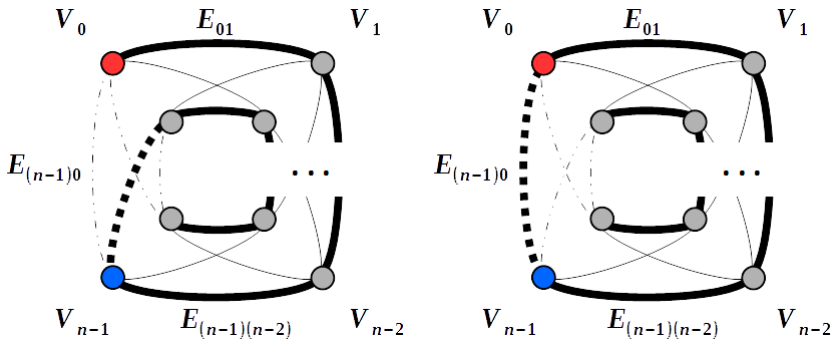


Figure 2.13: Example of an assignment on $\mathcal{P}^{(n-1)m}$ and C^{nm}, with $m = 2$. If the end of path P_i (blue) does not coincide with its start (red) via $E_{(n-1)0}$, a path of length $2n-1$ is formed (a). However, by assumption $A(C^{nm})$ is composed of disjoint cycles of length n, implying that all paths are at most length $n-1$. It must be the case then that P_i ends where it starts via $E_{(n-1)0}$ to form a cycle (b).

highly impractical in general. Fortunately, the conditions can be reduced to apply solely to the *basis* of a stacked graph's cycle stacks.

Lemma 2.26 (Feasibility of Joined Cycle Stacks)**.** *Let $\mathbb{G}$ be asingular, of size m, and composed of two cycle stacks C^{rm} and C^{tm} which share a single common path stack $\mathcal{P}^{sm}$ of length $s < \min\{r, t\}$ (Figure 2.14). In other words,*

$$C^{rm} \cap_E C^{tm} = \mathcal{P}^{sm} \tag{2.16}$$

where $\cap_E$ denotes the edge stacks shared between two stacked graph structures. Then assignment A on $\mathbb{G}$ is feasible if and only if $A(C^{rm})$ and $A(C^{tm})$ are both feasible.

Proof. The forward component is true by Theorem 2.25. For the backward component, the key is that $\mathcal{P}^{sm}$ is shared by both cycle stacks. By Theorem 2.18 and Corollary 2.19, $A(\mathcal{P}^{sm})$ creates m disjoint paths P_k, irrespective of A's details. In addition, by Theorem 2.23, $A(C^{rm})$ and $A(C^{tm})$ each form m disjoint cycles, C_i and C'_j, respectively. Because $\mathcal{P}^{sm}$ is a stacked subgraph of by both cycle stacks, these two properties imply that every disjoint cycle in $A(C^{rm})$ shares a path of length s with exactly one disjoint cycle in $A(C^{tm})$. Thus vertices disjoint in $A(C^{rm})$ and $A(C^{tm})$ are also disjoint in $A(\mathbb{G})$ (Figure 2.15).

Therefore, if both $A(C^{rm})$ and $A(C^{tm})$ are feasible, $A(\mathbb{G})$ is feasible. □

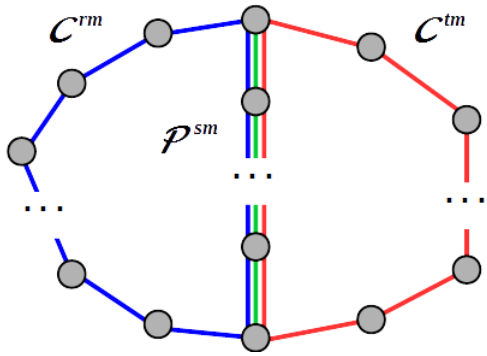


Figure 2.14: Asingular $\mathbb{G}$ formed by C^{rm} (blue and green) and C^{tm} (red and green) which share a single path $\mathcal{P}^{sm}$ (green).

Lemma 2.26 can be generalized to two cycle stacks that share multiple disjoint paths.

Lemma 2.27 (Feasibility of Multi-Joined Cycle Stacks)**.** *Let $\mathbb{G}$ be as in Theorem 2.26 except that C^{tm} and C^{tm} share multiple disjoint path stacks $\mathcal{P}^{am}$, $\mathcal{P}^{bm}$, etc. (Figure 2.16). Then an assignment A on $\mathbb{G}$ is feasible if and only if $A(C^{rm})$ and $A(C^{tm})$ are feasible.*

Proof. The proof is the inductive application of the proof for Theorem 2.26 on each shared path stack. Each path in $A(\mathcal{P}^{am})$ is connected to one path in $A(\mathcal{P}^{bm})$, which is connected to one path in $A(\mathcal{P}^{cm})$, etc., each of which is part of a single cycle in $A(C^{rm})$ and a single cycle in $A(C^{tm})$. □

Lemma 2.26 also applies when $s = 0$. In this case, C^{rm} and C^{tm} share only a single vertex stack. This can be generalized to apply to any two disjoint cycle stacks path stack connected to each other.

Lemma 2.28 (Feasibility of Path Stack Connected Cycle Stacks)**.** *Let $\mathbb{G}$ be asingular, of size m, and composed of two edge stack disjoint cycle stacks C^{rm} and C^{tm} which are path stack connected by $\mathcal{P}^{sm}$ such that*

$$C^{rm} \cap_E \mathcal{P}^{sm} = \emptyset \tag{2.17}$$
$$C^{tm} \cap_E \mathcal{P}^{sm} = \emptyset, \tag{2.18}$$

where is $\cap_E$ is as in Theorem 2.26 (Figure 2.17). Then an assignment A on $\mathbb{G}$ is feasible if and only if $A(C^{rm})$ and $A(C^{tm})$ are both feasible.

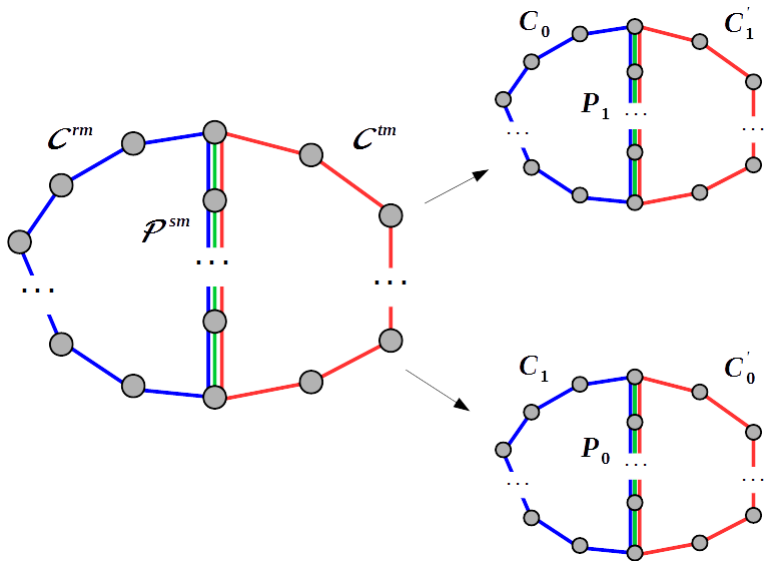


Figure 2.15: If every C_i and C_j' contain exactly one P_k as a stacked subgraph, it's necessarily the case that m disjoint covers of $\mathbb{G}$ are formed. In the case of $m = 2$, for example, two covers can be formed by C_0 and C_1' sharing P_1, and C_1 and C_0' sharing P_0.

Proof. The proof is similar to that of Lemma 2.26 in that every disjoint cycle of $A(C^{rm})$ is connected to a single disjoint cycle in $A(C^{tm})$ via a single path in $A(\mathcal{P}^{sm})$, and vice versa. Again, the details of $A(\mathcal{P}^{sm})$ are irrelevant. □

Lemmas 2.26 through 2.28 provide the components necessary to show that feasibility is preserved under the disjoint union of cycle stacks. The result is that, as a basis can always be found for the cycles of a graph - and consequently the cycle stacks of a stacked graph - and as cycles are formed via the disjoint union of basis cycles, then a feasible cycle stacks basis implies that all cycle stacks are feasible, and, consequently, an the entire assignment $A(\mathbb{G})$.

Theorem 2.29 (Feasibility of Cores)**.** *For asingular $\mathbb{G}$ and assignment A on it, $A(\mathbb{G})$ is feasible if and only if $A(\mathring{\mathbb{G}})$ is feasible.*

Proof. The proof is the same for Theorem 2.25. □

Corollary 2.30 (Feasibility of Fringes)**.** *For asingular $\mathbb{G}$ and assignment A on it, $A(\tilde{\mathbb{G}})$ is always feasible.*

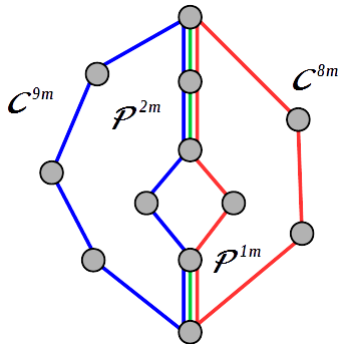


Figure 2.16: $\mathcal{C}^{9m}$ and $\mathcal{C}^{8m}$ multi-joined by $\mathcal{P}^{1m}$ and $\mathcal{P}^{2m}$.

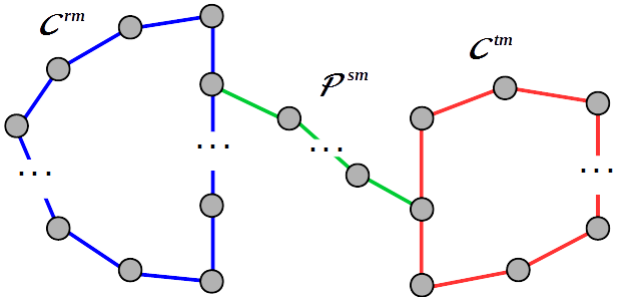


Figure 2.17: Two edge stack disjoint cycle stacks $\mathcal{C}^{rm}$ and $\mathcal{C}^{tm}$ joined by $\mathcal{P}^{sm}$.

Proof. By definition, $\tilde{\mathbb{G}}$ contains only edge stacks that are not part of cycle stacks in $\mathbb{G}$. Therefore, there are no cycle stacks in $\tilde{\mathbb{G}}$. By Corollary 2.19 and Theorem 2.25, $A(\tilde{\mathbb{G}})$ is feasible. □

Recall that a *spanning tree* of a connected graph $\mathcal{G}$ is a connected sub-graph which covers the vertices of $\mathcal{G}$ and contains no cycles. The *fundamental cycles* of a spanning tree are the cycles created by adding back the edges in $\mathcal{G}$ that were removed when creating the spanning tree [AMO93]. For any spanning tree of $\mathcal{G}$, the associated fundamental cycles form a basis for the *cycle space* of $\mathcal{G}$ [Bol79]. That is, every cycle in $\mathcal{G}$ can be written as a disjoint union of some set of fundamental cycles.

This leads to the first cornerstone theorem of this work.

Theorem 2.31 (Feasibility of Fundamental Cycle Stacks)**.** *Let $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ be asingular and of size m, and let $\mathbb{T}$ be a spanning tree stack of $\mathbb{G}$. In addition, let*

$\mathbb{T}_C = \{C_i | i = 0, 1, \ldots, k-1\}$ *be the set of fundamental cycle stacks associated with* $\mathbb{T}$*, where* $k = |\mathcal{E}| - |\mathcal{N}| + 1$ [18]*. Then any assignment* A *on* $\mathbb{G}$ *is feasible if and only if* $A(C_i)$ *is feasible for all* $C_i \in \mathbb{T}_C$*.*

Proof. The forward component is true by Theorem 2.25. For the backwards component, all cycle stacks of $\mathbb{T}_C$ of are feasible. Any two fundamental cycles C_i and C_j form one of three types: they are joined by one or more shared path stacks, or are disjoint but still path stack connected, or both. In the first case, Lemmas 2.26 and 2.27 ensure feasibility is preserved, while Lemma 2.28 does the same for the second case. The third case is the combination of the first to, so that feasibilit is preserved. Therefore, $A(\mathbb{G})$ ia feasible. □

The restriction of Theorem 2.31 to asingular $\mathbb{G}$ is merely one of notational convenience: it applies to any $\mathbb{G}$ when spanning trees and forests are produce for $\mathring{\mathbb{G}}$.

The nice things about Theorem 2.31 is not only that any spanning tree can be used, but also that the associated fundamental cycles can be found in time complexity $\mathcal{O}\left(|\mathcal{N}|^2\right)$ [AMO93], which is independent of the size of the edge stacks in $\mathbb{G}$.

The second cornerstone theorem describes the nature of connected components in feasible $A(\mathbb{G})$.

Theorem 2.32 (Layers and Sublayers)**.** *If* $\mathbb{G}$ *is asingular and of size m, and if A is feasible on* $\mathbb{G}$*, then* $A(\mathbb{G})$ *produces a set* $\mathbb{H}_\mathbb{G}$ *of m connected components* $\mathcal{H}_i$

$$\mathbb{H}_\mathbb{G} := \{\mathcal{H}_i | i = 0, 1, \ldots, m-1\} \tag{2.19}$$

where each $\mathcal{H}_i$ *is isomorphic to* $\mathbb{G}$*.* $\mathbb{H}_\mathbb{G}$ *are the* layers of $A(\mathbb{G})$ *and* $\mathcal{H}_i$ *is the* i^{th} layer of $A(\mathbb{G})$*.*

If $\mathbb{G}$ *is singular or mixed, then any feasible* $A(\mathring{\mathbb{G}})$ *produces* $|\mathbb{G}_\mathcal{R}|$ *regions* $\mathcal{R}_i$*, each consisting of* $|\mathcal{R}_i|$ *components isomorphic to* $\mathcal{R}_i$*. The set* $\mathbb{H}_i$ *of connected components* $\mathcal{H}_j^i$ *in* $A(\mathcal{R}_i)$

$$\mathbb{H}_i := \{\mathcal{H}_j^i | j = 0, 1, \ldots, |\mathcal{R}_i| - 1\} \tag{2.20}$$

are the layers of $A(\mathcal{R}_i)$*, where* $\mathcal{H}_j^i$ *is the* j^{th} layer of $\mathcal{R}_i$*.*

For any connected stacked subgraph $\mathcal{K}$ *of a region* $\mathcal{R}_i$*, the set* $\underline{\mathbb{H}}_\mathcal{K}$ *of connected components* $\underline{\mathcal{H}}_j$ *of* $A(\mathcal{K})$

$$\underline{\mathbb{H}}_\mathcal{K} := \{\underline{\mathcal{H}}_j | j = 0, 1, \ldots, |\mathcal{R}_i| - 1\} \tag{2.21}$$

are the sub-layers of $A(\mathcal{K})$*, where* $\underline{\mathcal{H}}_j$ *is the* j^{th} sub-layer of $A(\mathcal{K})$*.*

[18] The number of fundamental cycles for a graph is derived in [AMO93].

Proof. By Theorems 2.18 and 2.23, all feasible assignments on any C^{nm} or $\mathcal{P}^{nm}$ produce m disjoint cycles or paths that are, by Corollaries 2.20 and 2.24, isomorphic to their parent object. The isomorphism proofs are by induction on the cycle and path stacks of $\mathbb{G}$ or its regions. □

Before ending this section, two more concepts will prove useful: assingular assignment constraints and regional stacked graphs.

Definition 2.33 (Asingular Assignment Constraints)**.** Let E_{pq} be asingular and A an assignment on it. A_{pq} is *fully constrained* if the values of all $x_{ij}^{pq} \in A_{pq}$ are required to take specific value (i.e. they're parameters and not variables). If only some are restricted, then A_{pq} is *partially constrained*. Otherwise, it is *unconstrained*.

If A is an assignment on some $\mathcal{P}^{nm}$ so that all disjoint paths in $A(\mathcal{P}^{nm})$ are required to start and end on specific vertices, then $A(\mathcal{P}^{nm})$ is fully constrained. If only some are so required, then $A(\mathcal{P}^{nm})$ is partially constrained. Otherwise it is unconstrained. If there are additional constraints on $A(\mathcal{P}^{nm})$ so that the paths of $A(\mathcal{P}^{nm})$ must pass through one or more additional, specific vertices, these are termed *internal* constraints with the specified vertices called *waypoints*.

Lemma 2.34 (Fully Constrained Asingular Cycle Stack Assignments)**.** *Assignments on asingular cycle stacks are fully constrained.*

Proof. See the proof of Theorem 2.23. □

Definition 2.35 (Regional Stacked Graphs $\mathbb{R}_{\mathbb{G}}$)**.** For any stacked graph $\mathbb{G}$, the *regional stacked graph of* $\mathbb{G}$, $\mathbb{R}_{\mathbb{G}}$, is the stacked graph formed by replacing each region $\mathcal{R}_s \in \mathbb{G}_{\mathcal{R}}$ with a vertex stack V_s, where $|V_s| = |\mathcal{R}_s|$. Edge stack $E_{st} \in \mathbb{R}_{\mathbb{G}}$ exists only if the singular boundary $\mathring{\mathcal{E}}_{st}$ in $\mathbb{G}$ is non-empty. Each vertex v_i^s is associated with layer $\mathcal{H}_i^s$.

Regional stacked graphs abstract the details of individual layers and focus on the relationship between them. Paths in $\mathbb{R}_{\mathbb{G}}$ represent broad-scale descriptions of connectedness between layers through regional boundaries.

Figure 2.18 provides an example of regional stacked graphs using both packed and unpacked diagrams.

Notice that the vertices in $\mathbb{R}_{\mathbb{G}}$ represent *abstract* layers as layers are only properly defined in the context of assignments (Theorem 2.32). The same theorem, however, tells us that, regardless of the assignment, we always know the *number* of layers a feasible assignment will produce. We can therefore discuss layers in a general sense without the complexities involved when working with assignments.

But what of assignments on $\mathbb{R}_{\mathbb{G}}$? Since $\mathbb{R}_{\mathbb{G}}$ is a stacked graph, we should be able to define assignments on it. While technically true, such assignment will be

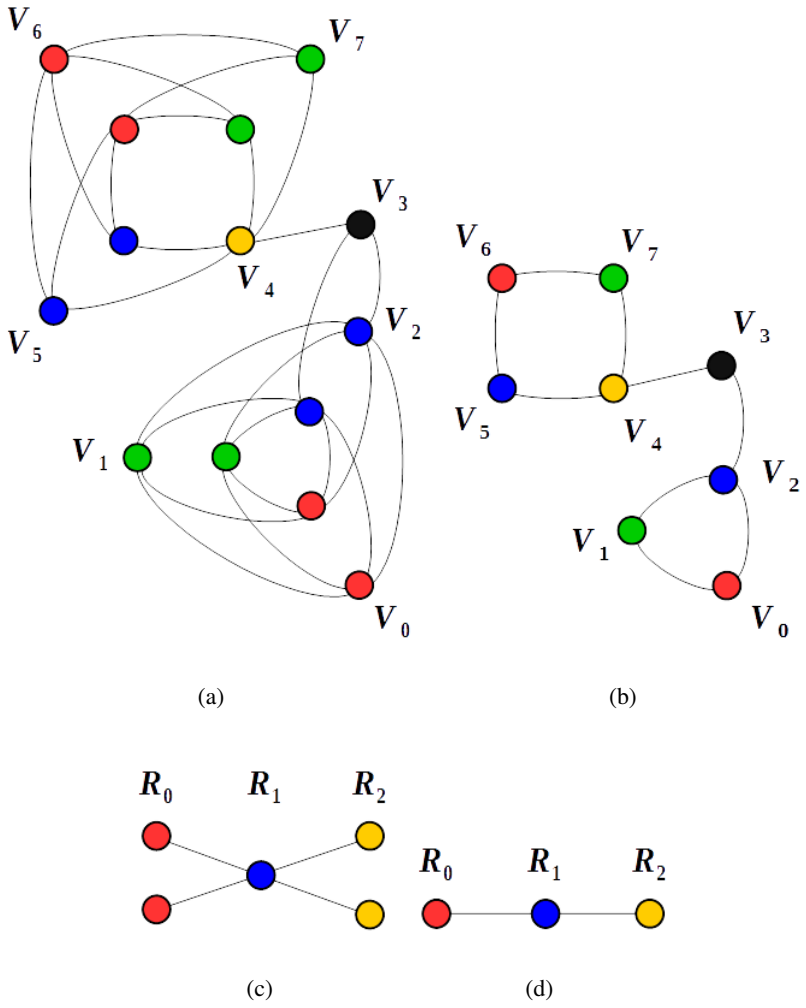


Figure 2.18: The (a) unpacked and (b) packed diagrams of some $\mathbb{G}$ and its regional stacked graph, as well as its (c) unpacked and (d) packed regional stacked graph $\mathbb{R}_{\mathbb{G}}$. $\mathcal{R}_0$ consists of edge stacks E_{56} and E_{67}, $\mathcal{R}_1$ consists solely of E_{34}, and $\mathcal{R}_2$ consists of E_{01}, E_{12}, and E_{20}. Singular boundary $\hat{\mathcal{E}}_{01}$ in $\mathbb{R}_{\mathbb{G}}$ consists of singular edges E_{45} and E_{47}, while $\hat{\mathcal{E}}_{12}$ consists only of E_{23}.

largely meaningless. The value of $x_{ij}^{rt} \in A(\mathbb{R}_{\mathbb{G}})$ for some A indicates whether layer $\mathcal{H}_i^r$ is adjacent to $\mathcal{H}_j^t$ through $\hat{\mathcal{E}}_{rt}$, but little more; we know nothing of which vertices are in which layer.

When we remember that $\mathbb{R}_{\mathbb{G}}$ is a derived structure, it isn't too hard to imagine that derivatives of assignments can also be had.

Definition 2.36 (Derived Assignments $A(\mathbb{R}_{\mathbb{G}})$)**.** Let A be a feasible assignment on some $\mathbb{G}$. Then $A(\mathbb{R}_{\mathbb{G}})$ is a *derived* assignment that can be defined in two ways: *binary* and *proportional*.

In a binary assignment, for each $e_{ij}^{rt} \in E_{rt}$, $E_{rt} \in \mathbb{R}_{\mathbb{G}}$, $x_{ij}^{rt} = 1$ if there is at least one vertex in $\mathcal{H}_i^r$ adjacent to a vertex in $\mathcal{H}_j^t$, and $x_{ij}^{rt} = 0$ if there isn't. In the proportional assignment, x_{ij}^{rt} is a proportional measure of such edges, defined by

$$x_{ij}^{rt} := \frac{\left|\left\{ x_{cd}^{ab} \middle| A_{ab} \in A(\hat{\mathcal{E}}_{rt}), \mathrm{h}(v_c^a) = \mathcal{H}_i^r \text{ and } \mathrm{h}(v_d^b) = \mathcal{H}_j^t \right\}\right|}{\left|\left\{ x_{cd}^{ab} \middle| A_{ab} \in A(\hat{\mathcal{E}}_{rt}), \mathrm{h}(v_c^a) = \mathcal{H}_i^r \right\}\right|} \tag{2.22}$$

where $\mathrm{h}(\cdot)$ is the *layer* function which returns the layer of a vertex[19].

Binary assignments are used to indicate whether it's possible to move directly between layers of adjacent regions. Proportional assignments give a measure of how likely such movement is as the ratio of the number of adjacencies that exist between two layers of two adjacent regions to the total number of possible adjacencies for the same layers. The proportional assignment $A(\mathbb{R}_{\mathbb{G}})$ is one instance of where assignment flow values x_{ij}^{rt} are non-binary.

Note that in a binary assignment $x_{ij}^{rt} = x_{ji}^{tr}$. This is not generally true in a proportional assignment. In Section 2.5 we introduce a technique for ensuring that feasible assignments are such that $A(\mathbb{R}_{\mathbb{G}})$ is always binary, the Uniform Adjacency Constraint. There are theoretical reasons why binary assignments are desirable. These are discussed in Section 7.1 and Chapter 8.

2.4 Feasibility of Non-Simple Stacked Graphs

The distinguishing characteristic between simple and non-simple stacked graphs is that non-simple stacked graphs contain loop stacks and multi-edge stacks, or cycle stacks of length one and two, respectively. Curiously, however, and much to our satisfaction, little difference exists between the two from a feasibility perspective.

[19]The function is undefined in the absence of an assignment.

Lemma 2.37 (Feasible Assignments on Loop Stacks)**.** *For any asingular stacked loop E_{pp}, or C^{1m}, the only feasible assignment A_{pp} is such that*

$$A_{pp} := \{e_{ii}^{pp} | i = 0, 1, \dots, |E_{pp} - 1|\}, \tag{2.23}$$

or $x_{ij}^{pp} = 1$ if $i = j$ and $x_{ij}^{pp} = 0$ otherwise.

Proof. A_{pp} is feasible if v_i^p and v_j^p are disjoint when $i \neq j$. Thus $e_{ij}^{pp} \notin A_{pp}$ when $i \neq j$. Therefore, A_{pp} can consists only of edges e_{ii}^{pp}, which is a valid assignment. □

The assignment defined in Lemma 2.37 will play an important role throughout this work, and so is worthy of its own notation. Figure 2.19 illustrates the assignment.

Definition 2.38 (Identity Assignment)**.** For any asingular edge stack E_{pq}, the *identity assignment* A^* is

$$A_{pq}^* := \{e_{ii}^{pq} | e_{ii}^{pq} \in E_{pq}, i = 0, 1, \dots, |E_{pq}| - 1\}, \tag{2.24}$$

or $x_{ij}^{pp} = 1$ if $i = j$ and $x_{ij}^{pp} = 0$ otherwise. As A^* has the same construction for all E_{pq}, the subscript of A_{pq}^* can be dropped.

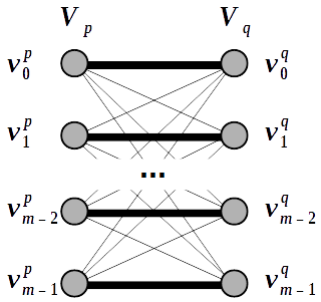


Figure 2.19: The identity assignment A^*.

Lemma 2.39 (Feasible Assignments on Multi-Edge Stacks)**.** *Let C^{2m} be the asingular cycle stack defined by $E_{pq} - E_{qp}$. Then assignment A on C^{2m} is feasible if*

$$e_{ij}^{pq} \in A_{pq} \iff e_{ji}^{qp} \in A_{qp}. \tag{2.25}$$

Equivalently, if E_{pq} and $E'_{p'q'}$, are two different edge stacks between V_p and V_q, then A_{pq} and A'_{pq} are feasible if and only if $A_{pq} = A'_{pq}$.

Proof. The forward and backward components involve identical proofs. If $e_{ij}^{pq} \in A_{pq}$ but $e_{ji}^{qp} \notin A_{qp}$, then $A(\mathcal{C}^{2m})$ is not feasible as $e_{ij}^{pq} - e_{jk}^{pq}$, $k \neq j$, is a path in $A(\mathcal{C}^{2m})$ so that v_i^p and v_k^p are not disjoint. Therefore, $e_{ji}^{qp} \in A_{qp}$. □

Lemmas 2.37 and 2.39 lead directly to the following theorem concerning non-simple stacked graphs.

Theorem 2.40 (Degeneracy of Non-Simple Stacked Graph Assignment Feasibility)**.** *Let $\mathbb{G}$ be an asingular, non-simple stacked graph. In addition, let $\mathbb{G}'$ be the sub-graph of $\mathbb{G}$ formed by removing all loop stacks and all but one edge stack in every multi-edge stack. Then, for any feasible assignment A' on $\mathbb{G}'$, there is exactly one feasible assignment A on $\mathbb{G}$ such that $A' \subset A$.*

Proof. The only edges in $\mathbb{G}$ that are not in $\mathbb{G}'$ are those associated with loop and multi-edge stacks. In addition, the only cycles in $\mathbb{G}$ that are not in $\mathbb{G}'$ are likewise associated with loop and multi-edge stacks. By Lemma 2.37, there is only one feasible assignment on loop stacks, A^*. Thus, if $\mathbb{G}''$ is the stacked graph formed by adding the loop stacks of $\mathbb{G}$ to $\mathbb{G}'$, there is only one feasible assignment A'' on $\mathbb{G}''$ such that $A' \subset A''$.

Similarly, let E_i^{pq} be the i^{th} edge stack for some multi-edge stack in $\mathbb{G}$ and let E_k^{pq} be the one edge stack preserved in $\mathbb{G}'$. Then, by Lemma 2.39, it must be the case that A_i^{pq} and A_k^{pq} have the same structure for all i. Thus, if $\mathbb{G}'''$ is the stacked graph formed by adding the multi-edge stacks of $\mathbb{G}$ not in $\mathbb{G}'$ to $\mathbb{G}'$, there is only one feasible assignment A''' on $\mathbb{G}'''$ such that $A' \subset A'''$.

As A'' and A''' are uniquely feasible and defined by $\mathbb{G}''$ and $\mathbb{G}'''$, it follows that $A := A'' \cup A'''$ is the only feasible assignment on $\mathbb{G}$ for which $A \subset A'$. □

In short, Theorem 2.40 states that every asingular, non-simple stacked graph degenerates into an asingular, simple stacked graph where finding feasible assignments on the latter immediately produces an associated - and unique - assignment on the former. We can therefore ignore non-simple stacked graphs and focus our attention solely on simple ones.

2.5 Singular Constraints

Our definition of assignment feasibility ignores singular edge stacks (Definition 2.17). This does not imply that singularities are entirely free of constraints, however. There may be conditions on singular assignments which need to be met in order for them to be considered feasible in the context of whatever problem is being modelled using $\mathbb{G}$.

The assignment feasibility condition in Definition 2.17 is in a sense the *basic* condition. It may be necessary to impose additional conditions to reflect known or suspected relationships existing in the data that vertex stacks represent. To this end, we introduce *non-basic* singular feasibility constraints.

Definition 2.41 (Non-Basic Singular Constraints)**.** Let $E_{pq} \in \mathbb{G}$ be singular with $|V_p| > |V_q|$. A set of *non-basic* singular feasibility constraints on E_{pq} define additional restrictions that must be satisfied by any A on E_{pq} for A_{pq} to be considered feasible. *Collapsing* constraints define how V_p maps to V_q, while *expanding* constraints define the reverse, from V_q to V_p. When both collapsing and expanding constraints are part of the definition, then the constraint is *mixed*.

If feasibility constraints are at least partially defined on labelled structures[20], then they are *labelled* constraints, otherwise they are *unlabelled* constraints.

If the same constraints are applied to every edge stack in $\hat{\mathbb{G}}$, then singular feasibility is *consistent*, otherwise it is *inconsistent*.

If no additional constraints are used so that only that of Definition 2.17 hold, the singular feasibility constraints are considered *basic*.

One such non-basic constraint is Uniform Layer Adjacency.

Definition 2.42 (Uniform Layer Adjacency)**.** Let A be an assignment $\mathbb{G}$ with regions $\mathcal{R}_r$ and $\mathcal{R}_t$, $|\mathcal{R}_r| > |\mathcal{R}_t|$, and a non-empty $\hat{\mathcal{E}}_{rt}$. The *Uniform Layer Adjacency* constraint (ULA) requires that all vertices in the collapsing vertex stacks of $A(\hat{\mathcal{E}}_{rt})$ of the same layer must be adjacent to vertices in expanding vertex stacks all belonging to the same, single layer.

Formally, $A(\hat{\mathcal{E}}_{rt})$ is feasible if for any $x_{ab}^{cd} = 1$ in $A(\hat{\mathcal{E}}_{rt})$ with $v_a^c \in \mathcal{R}_r$ and $v_b^d \in \mathcal{R}_t$ such that

$$\mathrm{h}(v_a^c) = \mathcal{H}_x^r \text{ and } \mathrm{h}(v_b^d) = \mathcal{H}_y^t \tag{2.26}$$

then

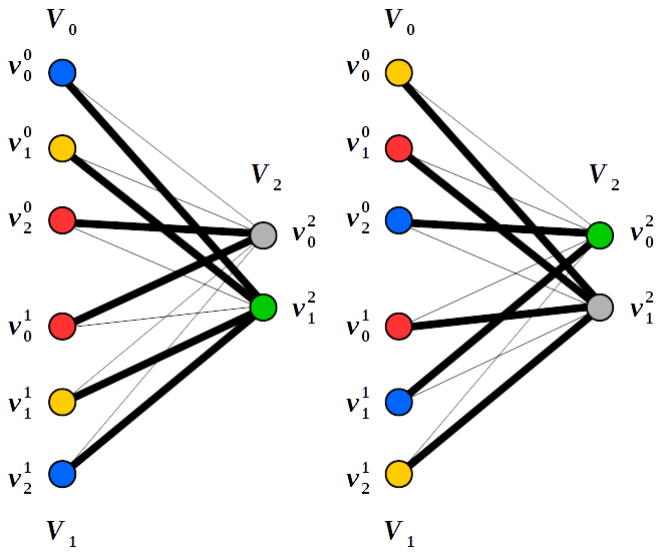
$$\mathrm{h}(v_i^p) = \mathcal{H}_x^r \text{ and } \mathrm{h}(v_j^q) = \mathcal{H}_y^t \tag{2.27}$$

for *all* $x_{ij}^{pq} = 1$ with $v_i^p \in \mathcal{R}_r$ and $v_j^q \in \mathcal{R}_t$.

The ULA constraint means that movement between layers through singular boundaries is consistent (Figure 2.20). It also plays nicely with derived assignments.

Lemma 2.43. *Feasible assignments A on $\mathbb{G}$ satisfying the ULA constraint produce derived assignments $A(\mathbb{R}_{\mathbb{G}})$ whose proportional and binary forms are identical.*

[20] Such as on specific vertex or (sub-)layer numbers.



(a) (b)

Figure 2.20: Let V_0 and V_1 be of size 3 and in the same region $\mathcal{R}_r$, V_2 be size 2 in $\mathcal{R}_t$, and $\hat{\mathcal{E}}_{rt} = \{E_{02}, E_{13}\}$. ULA requires that sub-layers in $\mathcal{R}_r$ (blue, yellow, and red vertices) map uniformly to sub-layers in $\mathcal{R}_t$ (grey and green vertices). Both (a) and (b) are feasible assignments as (a) maps {blue, yellow} → green and {red} → grey, while (b) maps {yellow, red} → grey and {blue} → green.

Proof. Either all vertices in $\mathcal{H}_x^r$ are adjacent to vertices in $\mathcal{H}_y^t$, or none are. Consequently, the set in the numerator of Equation 2.22 is the same as the denominator or empty. The proportional measure x_{ij}^{rt} is therefore either 0 or 1. □

Chapter 3

Stacked Graphs and Symmetric Groups

In this chapter we explore the connection between stacked graphs and the symmetric groups $\mathbb{S}_m$, namely, how any desingularized stacked graph $\mathring{\mathbb{G}}$ can be written as a system of group compositions, and vice versa. The relationship between singular edge stacks and $\mathbb{S}_m$ is not discussed in this work, being unimportant to the project, but is certainly worthy of future consideration

To better reflect their connection with stacked graphs, group compositions of the form

$$\sigma * \tau * \alpha * \beta \tag{3.1}$$

are read left to right, not right to left as is general practice. The basic properties and definitions of group theory below are based on [Fra03].

3.1 Assignments and $\mathbb{S}_m$

The elements of symmetric group $\mathbb{S}_m$ are defined as the $m!$ permutations of m element, which we'll take to be the first m integers, including 0. Elements can be represented in a two-line notation. For example, in

$$\sigma := \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 1 & 3 & 0 \end{pmatrix}, \tag{3.2}$$

where $\sigma \in \mathbb{S}_4$, the sequence

$$0\,1\,2\,3 \tag{3.3}$$

is permuted into

$$2\,1\,3\,0 \tag{3.4}$$

with σ mapping 0 to 2, 1 to 1, 2 to 3, and 3 to 0. These individual mappings are called *transpositions*. The transposition of a with b is written as

$$(a\ b) \tag{3.5}$$

with every element of $\mathbb{S}_m$ capable of being written as a product of transpositions. For example

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 1 & 3 & 0 \end{pmatrix} = (11)(02)(03). \tag{3.6}$$

Theorem 3.1. *[Fra03, Corollary 9.12] Any permutation of a finite set of at least 2 elements can be written as a product of transpositions. [Corollary 9.12, Fraleigh]*

Permutations on m elements can be viewed as perfect matching on the complete bipartite graph $\mathcal{B}_{mm}$[21]. This is equivalent to a bijection between two sets of m elements, or an automorphism on m elements. Figure 3.1 illustrates σ as a perfect matching on $\mathcal{B}_{44}$.

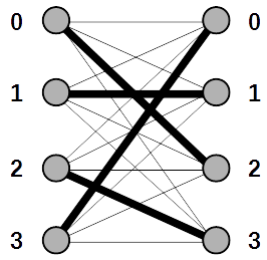


Figure 3.1: The complete bipartite graph $\mathcal{B}_{44}$ consists of every edge, thick and thin, while σ - a perfect matching or, in our case, bijective mapping of the left vertices to the right vertices - consists of only the thick lines. The matching is a graphical representation of the two-line permutation Equation 3.2.

Clearly, there is a relationship between Figure 3.1 and our edge stacks and assignments.

Definition 3.2 (Edge Stack Group Duals)**.** Let A be an assignment on asingular E_{pq} of size m. Then A_{pq} is isomorphic to an element $\alpha_{pq} \in \mathbb{S}_m$. A_{pq} is the *(stacked) graph dual of* α_{pq} and α_{pq} the *(symmetric) group dual of* A_{pq}. If no assignment exist on E_{pq}, then group dual of E_{pq} is denoted by ϵ_{pq}, where ϵ_{pq} is an unidentified element in $\mathbb{S}_m$. Formally

$$e_{ij}^{pq} \in A_{pq} \iff (ij) \in \alpha_{pq}. \tag{3.7}$$

Every edge in an assignment is isomorphic to a transposition. If $e_{ij}^{pq} \in A_{pq}$, then e_{ij}^{pq} is the graph dual of transposition (ij) in α_{pq}, which is denoted by ϵ_{ij}^{pq}.

[21]Matchings on graphs are subsets of edges that are edge-independent (i.e. share no common vertices) while *perfect* matchings are those that cover a graph's vertex set [Fou92].

If the transposition details of α_{pq} are unknown so that the value of j is unknown in ϵ_{ij}^{pq}, then we define the i^{th} *transpose of* α_{pq} to be ϵ_i^{pq}, which always exists[22]. As ϵ_{pq} is a group element, ϵ_i^{pq} is also used to indicated the (unknown) i^{th} transposition of ϵ^{pq} [23].

The two-line notation of α_{pq} is written as

$$\alpha_{pq} = \begin{pmatrix} \nu_0^p & \nu_1^p & \cdots & \nu_{m-1}^p \\ \nu_{w_0}^q & \nu_{w_1} & \cdots & \nu_{w_{m-1}}^q \end{pmatrix} \tag{3.8}$$

where ν_i^p and $\nu_{w_i}^q$ are the group duals of the vertices v_i^p and $v_{w_i}^q$ in $e_{iw_i}^{pq}$ [24]. The group duals of vertices are precisely the integers 0 through $m-1$

$$\nu_i^p, \nu_{w_i}^q \in \{0, 1, \ldots, m-1\} \tag{3.9}$$

with $\nu_i^p = \nu_j^p$ and $\nu_{w_i}^q = \nu_{w_j}^q$ only when $i = j$.

Finally, the group dual of x_{ij}^{pq} is denoted by χ_{ij}^{pq}.

For every assignment A_{pq} on an asingular edge stack, there is a corresponding symmetric group element; for every edge in A_{pq}, a transposition. In light of Figure 3.1, the isomorphisms are obvious enough to avoid a formal proof [25].

The graph dual of σ (Figure 3.2) is identical to Figure 3.1, save for the vertex labels.

The two-line notation for group duals is cumbersome for general α_{pq}. For a concrete example, the two-line notation for σ is

$$\begin{pmatrix} \nu_0^p & \nu_1^p & \nu_2^p & \nu_3^p \\ \nu_2^q & \nu_1^q & \nu_3^q & \nu_0^q \end{pmatrix}. \tag{3.10}$$

Notice that the subscripts in Equation 3.10 are identical to the values in Equation 3.2.

The single most important group dual is, of course, the identity element, whose associated graph dual is a familiar one.

[22] This is the group interpretation of the i^{th} edge flow in Definition 2.15.

[23] There is some abuse of notation here. Technically, ϵ_{pq} should be E_{pq}, or capital epsilon, since group elements are sets of transpositions. However, we've intentionally restricted ourself to lowercase Greek letters for group theoretic concepts.

[24] The group dual is Greek *nu*. The visual similarity is deliberate.

[25] The relationship between perfect matchings and permutations is not novel to this work. However, I've been unable to locate existing literature generalizing the relationship to sequences of matchings - our 'stacked' graphs - and sequences of permutations, or compositions, in $\mathbb{S}_m$. This chapter is therefore original, up to the knowledge of it's author.

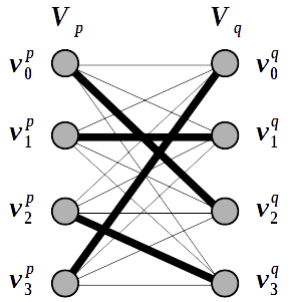


Figure 3.2: To form a graph dual, σ requires - at this stage arbitrary - double subscripts. Thus if $\sigma \to \sigma_{pq}$, the graph dual is A_{pq}.

Fact 3.3. *The identity element ι of $\mathbb{S}_m$*

$$\begin{pmatrix} 0 & 1 & \dots & m\text{-}1 \\ 0 & 1 & \dots & m\text{-}1 \end{pmatrix} \tag{3.11}$$

is the group dual of A^*.

A note of caution. The term ϵ_{pq} represents an abstract element in $\mathbb{S}_m$, or a set of unidentified transpositions. There are some difficulties with using ϵ_{ij}^{pq} and ϵ_{i}^{pq} as it's possible to write group elements as a product of transpositions (Theorem 3.1) in multiple ways. For example

$$\tau := (12)(35)$$

can also be written as

$$\tau = (12)(35)(46)(46).$$

Transpositions (46)(46) - which together can represent the identity ι [Fra03] - are part of the second representation of τ, but not the first. The statement 'transposition (ij) of σ', is therefore poorly defined. To correct this, we'll assume that transposition representations are minimal so that if $\sigma \in \mathbb{S}_m$, exactly m transposition are present, which may include one or more identities (ii) as in the case of σ in Equation 3.6.

Group duals can be extended to path and cycle stacks.

Definition 3.4 (Duality of Path and Cycle Stacks)**.** Let $\mathcal{P}^{nm}$ be the asingular path stack

$$\mathcal{P}^{nm} := E_{01} - E_{12} - \cdots - E_{(n-2)(n-1)}. \tag{3.12}$$

The *symmetric group dual of* $\mathcal{P}^{nm}$ is the composition

$$\pi^n := \epsilon_{01} * \epsilon_{12} * \cdots * \epsilon_{(n-2)(n-1)}. \tag{3.13}$$

If A is an assignment on $\mathcal{P}^{nm}$ such that

$$A(\mathcal{P}^{nm}) = A_{01} - A_{12} - \cdots - A_{(n-2)(n-1)}, \tag{3.14}$$

then the group dual of $A(\mathcal{P}^{nm})$ is the composition

$$\alpha(\pi^n) := \alpha_{01} * \alpha_{12} * \cdots * \alpha_{(n-2)(n-1)}. \tag{3.15}$$

Similarly, for $\mathcal{C}^{nm}$

$$\mathcal{C}^{nm} := E_{01} - E_{12} - \cdots - E_{(n-1)0}, \tag{3.16}$$

the group dual of $\mathcal{C}^{nm}$ is the composition

$$\delta^n := \epsilon_{01} * \epsilon_{12} * \cdots * \epsilon_{(n-1)0} \tag{3.17}$$

while the group dual of A on $\mathcal{C}^{nm}$ is the composition

$$\alpha(\delta^n) := \alpha_{01} * \alpha_{12} * \cdots * \alpha_{(n-1)0}. \tag{3.18}$$

The stacked graph structures $\mathcal{P}^{nm}$ and $\mathcal{C}^{nm}$ are the *(stacked) graph duals* of π^n and δ^n, respectively, while $A(\mathcal{P}^{nm})$ and $A(\mathcal{C}^{nm})$ are the (stacked) graph duals of $\alpha(\pi^n)$ and $\alpha(\delta^n)$, respectively.

If the size of the symmetric group that each dual belongs to is important, group duals of path and cycle stacks can be augmented to π^{nm} and δ^{nm}, respectively.

Fact 3.5. *Path stacks, cycle stacks, and assignments on them are isomorphic to their group duals.*

While it is common practice to omit the operation symbol $*$, it will be maintained here to emphasize the relation between path and cycle stacks with their group compositions as well as to space an otherwise cramped notation. The operation will, however, be dropped when dealing with transpositions.

As with E_{pq} and A_{pq}, it's important to stress the difference between ϵ_{pq} and α_{pq}: ϵ_{pq} is an unspecified element in $\mathbb{S}_m$ while α_{pq} is a particular value that ϵ_{pq} takes on. If π^n is the group dual of $\mathcal{P}^{nm}$ and $\alpha(\pi^n)$ the group dual of an assignment A on $\mathcal{P}^{nm}$, the underlying *structure* of $\alpha(\pi^n)$ is defined by π^n; $\alpha(\pi^n)$ is a particular valuation of π^n.

This highlights a key conceptual difference between stacked graphs and their group duals: even when there is no assignment on some E_{pq}, there is an implicit, unknown assignment associated with ϵ_{pq}. There is *always* a perfect matching or set of network flows x_{ij}^{pq} associated with some ϵ_{pq}, which is not the case with edge stacks. This, as we'll soon see, is a wonderful property.

Being elements of a group, group duals have inverses.

Theorem 3.6 (Inverse of Edge Stack Group Duals)**.** *Let A be an assignment on some asingular E_{pq}. Then the inverse of its group dual α_{pq} is α_{qp}. That is,*

$$\alpha_{pq}^{-1} = \alpha_{qp}.$$

Proof. Let $|E_{pq}| = m$. The edge set of A_{pq} is a vertex disjoint set

$$\{e_{ij}^{pq} | i, j = 0, 1, \dots, m-1\} \tag{3.19}$$

while that of $A_{q'p'}$ is

$$\{e_{ij}^{q'p'} | i, j = 0, 1, \dots, m-1\} \tag{3.20}$$

with $e_{ij}^{q'p'} = e_{ji}^{qp}$ [26]. As a result, the path stack

$$E_{pq} - E_{q'p'} \tag{3.21}$$

has the assignment

$$A_{pq} - A_{q'p'}, \tag{3.22}$$

where each disjoint path P_i is of the form $e_{ij}^{pq} - e_{ji}^{q'p'}$, which is equivalent to $e_{ij}^{pq} - e_{ji}^{qp}$. An example of this is shown in shown in Figure 3.3. In two-line notation, the group dual of $A_{pq} - A_{q'p'}$ is

$$\alpha_{pq} * \alpha_{q'p'} = \begin{pmatrix} v_0^p & \cdots & v_{m-1}^p \\ v_{w_0}^q & \cdots & v_{w_{m-1}}^q \end{pmatrix} * \begin{pmatrix} v_0^{q'} & \cdots & v_{m-1}^{q'} \\ v_{w_0}^{p'} & \cdots & v_{w_{m-1}}^{p'} \end{pmatrix} \tag{3.23}$$

$$= \begin{pmatrix} v_0^p & \cdots & v_{m-1}^p \\ v_{w_0}^q & \cdots & v_{w_{m-1}}^q \end{pmatrix} * \begin{pmatrix} v_{w_0}^q & \cdots & v_{w_{m-1}}^q \\ v_0^p & \cdots & v_{m-1}^p \end{pmatrix} \tag{3.24}$$

$$= \begin{pmatrix} v_0^p & \cdots & v_{m-1}^p \\ v_0^p & \cdots & v_{m-1}^q \end{pmatrix} \tag{3.25}$$

or the identity element ι. Therefore $\alpha_{pq}^{-1} = \alpha_{qp}$. □

Implicit in Theorem 3.6, which also applies to unassigned E_{pq} and its dual ϵ_{pq}, is that while stacked graphs can often be treated as undirected, group duals treat them as inherently directed, with inverses denoting changes in direction.

Theorem 3.7 (Inverse of Path Stack Group Duals)**.** *If A is an assignment on $\mathcal{P}^{nm}$ so that*

$$A(\mathcal{P}^{nm}) = A_{01} - A_{12} - \cdots - A_{(n-1)n} \tag{3.26}$$

[26] Primes are introduced to prevent $E_{pq} - E_{q'p'}$ from being a pair of parallel edges between V_p and V_q.

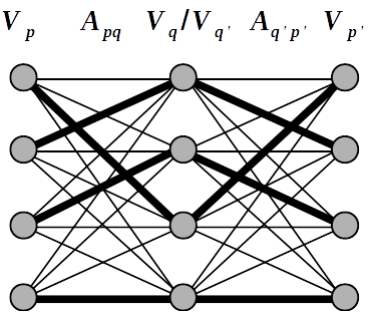


Figure 3.3: An example of $A_{pq} - A_{q'p'}$ where $|E_{pq}| = 4$.

with group dual

$$\alpha(\pi^n) = \alpha_{01} * \alpha_{12} * \cdots * \alpha_{(n-1)n} \tag{3.27}$$

then the inverse of $\alpha(\pi^n)$ is the group dual of

$$A(\bar{\mathcal{P}}^{nm}) = A_{n(n-1)} - \cdots - A_{21} - A_{10}, \tag{3.28}$$

or

$$\alpha(\bar{\pi}^n) = \alpha_{n(n-1)} * \cdots * \alpha_{21} * \alpha_{10}, \tag{3.29}$$

where $\bar{\mathcal{P}}^{nm}$ is $\mathcal{P}^{nm}$ in the opposite direction.

Proof. For any composition $\alpha * \beta$, the inverse of $\alpha * \beta$ is $\beta^{-1} * \alpha^{-1}$ [Rom12]. Thus the inverse of $\alpha(\pi^n)$ is

$$\alpha(\pi^n)^{-1} = \alpha_{(n-1)n}^{-1} * \cdots * \alpha_{12}^{-1} * \alpha_{01}^{-1}. \tag{3.30}$$

By Theorem 3.6, we have

$$\alpha_{(n-1)n}^{-1} * \cdots * \alpha_{12}^{-1} * \alpha_{01}^{-1} = \alpha_{n(n-1)} * \cdots * \alpha_{21} * \alpha_{10} \tag{3.31}$$

so that

$$\alpha(\pi^n)^{-1} = \alpha(\bar{\pi}^n) \tag{3.32}$$

with

$$\begin{aligned} \alpha(\pi^n) * \alpha(\bar{\pi}^n) &= \alpha_{01} * \alpha_{12} * \cdots * \alpha_{(n-1)n} * \alpha_{n(n-1)} * \cdots * \alpha_{21} * \alpha_{10} \\ &= \imath. \end{aligned} \tag{3.33}$$

□

Theorem 3.8 (Inverse of Cycle Stack Group Duals)**.** *If A is an assignment on C^{nm} so that*

$$A(C^{nm}) = A_{01} - A_{12} - \cdots - A_{n0} \tag{3.34}$$

with group dual

$$\alpha(\delta^n) = \alpha_{01} * \alpha_{12} * \cdots * \alpha_{n0} \tag{3.35}$$

then the inverse of $\alpha(\delta^n)$ is the group dual of

$$A(\bar{C}^{nm}) = A_{0n} - \cdots - A_{21} - A_{10}, \tag{3.36}$$

or

$$\alpha(\bar{\delta}^n) = \alpha_{0n} * \cdots * \alpha_{21} * \alpha_{10}, \tag{3.37}$$

where $\bar{C}^{nm}$ is C^{nm} in the opposite direction.

Proof. The proof is similar to that of Theorem 3.7. □

The double indices imposed on group duals by their stacked dual counterparts restricts right and left composition of additional elements. For example, given

$$\epsilon_{45} * \epsilon_{57} * \epsilon_{73} = \beta \tag{3.38}$$

for some $\beta \in \mathbb{S}_m$, one can normally compose both sides by the same arbitrary element

$$\epsilon_{45} * \epsilon_{57} * \epsilon_{73} * \epsilon_{89} = \beta * \epsilon_{89} \tag{3.39}$$

without changing the relationship between the left and right sides of the equation. Unfortunately, the left side of Equation 3.39 is not a group dual for any path stack as ϵ_{73} and ϵ_{89} don't share an intervening index (Figure 3.4).

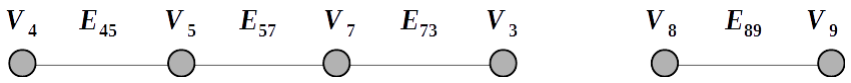


Figure 3.4: The graph dual of $\epsilon_{45} * \epsilon_{57} * \epsilon_{73} * \epsilon_{89}$ is not a path stack.

In general, group dual compositions cannot be operated on arbitrarily since group elements represent edge stacks: pre-pending or appending elements to a group dual composition is equivalent to either changing the connectivity or adding edge stacks to the corresponding stacked dual. There is, however, a notable exception.

Definition 3.9 (Cyclicity Condition)**.** A composition

$$\pi^n := \epsilon_{w_0w_1} * \epsilon_{w_1w_2} * \cdots * \epsilon_{w_{n-1}w_n} \tag{3.40}$$

satisfies the *cyclicity condition* if $w_n = w_0$ and

$$\begin{aligned} \epsilon_{w_0w_1} * \epsilon_{w_1w_2} * \cdots * \epsilon_{w_{n-1}w_n} &= \epsilon_{w_1w_2} * \epsilon_{w_2w_3} * \cdots * \epsilon_{w_{n-1}w_n} * \epsilon_{w_0w_1} \\ &= \epsilon_{w_2w_3} * \epsilon_{2_3w_4} * \cdots * \epsilon_{w_{n-1}w_n} * \epsilon_{w_0w_1} * \epsilon_{w_1w_2} \\ &\vdots \\ &= \epsilon_{w_nw_0} * \epsilon_{w_0w_1} * \epsilon_{w_1w_2} * \cdots * \epsilon_{w_{n-1}w_n}. \end{aligned} \tag{3.41}$$

If π^n satisfies these conditions, then it is a *cyclic* composition. Otherwise it is a *path* composition. The compositions above are the *cyclic forms of* π^n.

Similarly, $\alpha(\pi^n)$ is cyclic if

$$\begin{aligned} \alpha_{w_0w_1} * \alpha_{w_1w_2} * \cdots * \alpha_{w_{n-1}w_n} &= \alpha_{w_1w_2} * \alpha_{w_2w_3} * \cdots * \alpha_{w_{n-1}w_n} * \alpha_{w_0w_1} \\ &= \alpha_{w_2w_3} * \alpha_{2_3w_4} * \cdots * \alpha_{w_{n-1}w_n} * \alpha_{w_0w_1} * \alpha_{w_1w_2} \\ &\vdots \\ &= \alpha_{w_nw_0} * \alpha_{w_0w_1} * \alpha_{w_1w_2} * \cdots * \alpha_{w_{n-1}w_n}. \end{aligned} \tag{3.42}$$

Lemma 3.10 (Inverse of Cycle Stack Group Duals)**.** *Let π^n be a cyclic composition and π_i^n be one of its cyclic forms. Then the inverse of π_i^n is $\bar{\pi}_i^n$.*

Proof. The proof involves the application of Theorem 3.7 to each cyclic form π_i^n. □

Group duality can be generalize to entire stacked graphs.

Definition 3.11 (Group Dual of $\mathbb{G}$)**.** Let $\mathbb{G}$ be asingular and defined by a collection of k asingular path and cycle stacks $\mathcal{C}_i$ and $\mathcal{P}_j$, each possibly of different length, such that the union of their edge stack sets is $\mathcal{E}$ of $\mathbb{G}$ [27]. Then the vector

$$\gamma :=< \gamma_0, \gamma_1, \ldots, \gamma_k > \tag{3.43}$$

is the *(symmetric) group dual of* $\mathbb{G}$, the system of compositions γ_r where each γ_r is the group dual of a unique $\mathcal{P}_i$ or $\mathcal{C}_j$.

If γ is the group duals of $\mathbb{G}$, then $\mathbb{G}$ is the *(stacked) graph dual* of γ.

Similarly, if $\mathbb{G}$ is not asingular, then $\breve{\mathbb{G}}$ and $\breve{\gamma}$ are duals of each other.

[27] Each $\mathcal{C}_i$ and $\mathcal{P}_j$ need not be disjoint on their edge stacks.

The key differences between $\mathbb{G}$ and $\mathring{\mathbb{G}}$ is that each composition $\mathring{\gamma}_i$ in $\mathring{\gamma}$ may belong to symmetric groups of different sizes, depending on the size of each asingular path and cycle stack. For mixed $\mathbb{G}$, $\mathring{\mathbb{G}}$ is a forest, in which case each composition $\mathring{\gamma}_i$ of different sized groups will share no group elements in common. In this case, $\mathring{\mathbb{G}}$ and $\mathring{\gamma}$ are better treated on a regional basis, namely in terms $\mathcal{R}_i$ and their associated group duals.

The group duals of singular edge stacks are mappings between symmetric groups of different sizes. While undoubtedly important, they are not necessary for our project and would be an unprofitable digression. We'll leave treatment of them for a future work.

3.2 Assignment Feasibility and Symmetric Group Duals

We now turn to the group dual equivalent of feasibility.

Definition 3.12 (Constrained Group Duals)**.** If an assignment A on asingular E_{pq} is fully constrained (Definition 2.33) to be A', so that $A_{pq} = A'_{pq}$, then the *(symmetric) group dual constraint of* A_{pq} is β_{pq}, with

$$\beta_{pq} := \alpha'_{pq}. \tag{3.44}$$

Similarly, let A be an assignment on $\mathcal{P}^{nm}$ with

$$\mathcal{P}^{nm} = E_{01} - E_{12} - \cdots - E_{(n-1)n}. \tag{3.45}$$

If $A(\mathcal{P}^{nm})$ is fully constrained so that each path P_i starting on v_i^0 must end on a specific $v_{w_i}^n$, then the group dual constraint is β

$$\alpha_{01} * \alpha_{12} * \cdots * \alpha_{(n-1)n} = \beta, \tag{3.46}$$

where β is defined to be

$$\begin{pmatrix} \nu_0^0 & \cdots & \nu_{m-1}^n \\ \nu_{w_0}^0 & \cdots & \nu_{w_{m-1}}^n \end{pmatrix}. \tag{3.47}$$

If $A(\mathcal{P}^{nm})$ is unconstrained, then β is *free* and can be any element in $\mathbb{S}_m$. If $A(\mathcal{P}^{nm})$ partially constrained so that β is restricted to a subset of $\mathbb{S}_m$, then β is *restricted*. Otherwise, β is *bound*.

Fact 3.13 follows directly from Lemma 2.34 and the definition of constrained group duals.

Fact 3.13. *The group dual constraint β of any cycle stack assignment $A(C^{nm})$ is bound.*

We next introduce paths defined by transpositions.

Definition 3.14 (Transposition Paths)**.** Let $\alpha(\pi^n)$ be the group dual of some $A(\mathcal{P}^{nm})$

$$\alpha(\pi)^n = \alpha_{01} * \alpha_{12} * \cdots * \alpha_{(n-1)n}. \tag{3.48}$$

Each α_{pq} consists of m transpositions of the form $(v_i^p v_j^q)$, where v_i^p and v_j^q are vertex group duals. Let the succession of transpositions starting in α_{01} and ending in $\alpha_{(n-1)n}$ taking v_i^0 to some $v_{w_n}^n$

$$(v_i^0 v_{w_1}^1)(v_{w_1}^1 v_{w_2}^2) \dots (v_{w_{n-1}}^{n-1}, v_{w_n}^n) \tag{3.49}$$

where $w_k = 0, 1, \dots, m-1$, be the *i^{th} transposition path π_i of* $\alpha(\pi^n)$. π_i is the group dual of P_i, the i^{th} disjoint path in $A(\mathcal{P}^{nm})$ (Theorem 2.18).

Similarly, if $\alpha(\delta^n)$ is the group dual of some $A(C^{nm})$

$$\alpha(\delta)^n = \alpha_{01} * \alpha_{12} * \cdots * \alpha_{(n-1)0}, \tag{3.50}$$

the *i^{th} transposition cycle δ_i of* $\alpha(\delta^n)$ is

$$(v_i^0 v_{w_1}^1)(v_{w_1}^1 v_{w_2}^2) \dots (v_{w_{n-1}}^{n-1} v_{w_0}^0). \tag{3.51}$$

δ_i is the group dual of C_i, the i^{th} disjoint cycle in $A(\mathcal{P}^{nm})$ (Theorem 2.23).

As π^n and δ^n are compositions of elements, whatever those elements may be, transposition paths are likewise defined on them.

The dualities between π_i and P_i, δ_i and C_i, are intuitively clear and should not require proof. However, the follow lemma is included to further emphasize the relationship between the two.

Lemma 3.15. *For any π^n or $\alpha(\pi^n)$ in $\mathbb{S}_m$, there are exactly m transposition paths which form a disjoint cover of their vertex group duals. That is, every v_i^j in π^n or $\alpha(\pi^n)$ is in exactly one of m transposition path π_k or $\alpha(\pi_k)$.*

Proof. We prove this in the case of π^n. The case of $\alpha(\pi^n)$ is identical. Let

$$\pi^n = \epsilon_{01} * \epsilon_{12} * \cdots * \epsilon_{(n-1)n}. \tag{3.52}$$

Every ϵ_{pq} is an element in some $\mathbb{S}_m$, and therefore a one-to-one, onto mapping exists from the vertex group duals v_i^p, $i = 0, 1, \dots, m-1$, to v_j^q, $j = 0, 1, \dots, m-1$.

This can be written as a collection of transpositions of the form $(v_i^p v_j^q)$, where each vertex group dual belongs to a single transposition. If $n = 1$, then the transposition paths are of length one and form a cover of π^n. If $n \geq 2$, then π^n is a succession of these one-to-one mappings. Since each transposition path consists of adjacent transpositions of the form

$$(v_i^{r-1} v_j^r)(v_j^r v_k^{r+1}) \tag{3.53}$$

or, more generally, mappings of the form

$$v_i^{r-1} \to v_j^r \to v_k^{r+1} \tag{3.54}$$

it follows that every vertex group dual belongs to exactly one transposition path. That there are m transition paths follows directly from the bijective nature of the mapping. □

Lemma 3.15 is the group dual interpretation of Theorem 2.18. The group dual constraints for cyclic compositions are special.

Theorem 3.16 (Cyclic Compositions and Constraints)**.** *For every constrained cyclic composition $\delta^{n+1} = \beta$, $\beta = \iota$ for all $n \geq 0$.*

Proof. Let

$$\delta^{n+1} := \epsilon_{01} * \epsilon_{12} * \cdots * \epsilon_{n0} \tag{3.55}$$

where the i^{th} transposition cycle δ_i is

$$(v_i^0 v_{w_1}^1)(v_{w_1}^1 v_{w_2}^2) \ldots (v_{w_n}^n v_{w_0}^0). \tag{3.56}$$

As δ^{n+1} has $n + 1$ equivalent, cyclic forms

$$\begin{array}{lll}
\epsilon_{01} * \epsilon_{12} & * \cdots * & \epsilon_{(n-1)n}\epsilon_{n0} \\
\epsilon_{12} * \epsilon_{23} & * \cdots * & \epsilon_{n0} * \epsilon_{01} \\
\epsilon_{23} * \epsilon_{34} & * \cdots * & \epsilon_{01} * \epsilon_{12} \\
 & \vdots & \\
\epsilon_{n0} * \epsilon_{12} & * \cdots * & \epsilon_{(n-2)(n-1)} * \epsilon_{(n-1)n},
\end{array} \tag{3.57}$$

δ_i also has $n + 1$ equivalent forms, namely

$$\begin{array}{lll}
(v_i^0 v_{w_1}^1)(v_{w_1}^1 v_{w_2}^2) & \ldots & (v_{w_{n-1}}^{n-1} v_{w_n}^n)(v_{w_n}^n v_{w_0}^0) \\
(v_{w_1}^1 v_{w_2}^2)(v_{w_2}^2 v_{w_3}^3) & \ldots & (v_{w_n}^n v_{w_0}^0)(v_i^0 v_{w_1}^1) \\
(v_{w_2}^2 v_{w_3}^3)(v_{w_3}^3 v_{w_4}^4) & \ldots & (v_i^0 v_{w_1}^1)(v_{w_1}^1 v_{w_2}^2) \\
 & \vdots & \\
(v_{w_n}^n v_{w_0}^0)(v_i^0 v_{w_1}^1) & \ldots & (v_{w_{n-2}}^{n-2} v_{w_{n-1}}^{n-1})(v_{w_{n-1}}^{n-1} v_{w_n}^n).
\end{array} \tag{3.58}$$

However, the forms of δ_i collapse, respectively, into

$$\begin{gathered}(\nu_i^0 \nu_{w_0}^0) \\ (\nu_{w_1}^1 \nu_{w_0}^0)(\nu_i^0 \nu_{w_1}^1) \\ (\nu_{w_2}^2 \nu_{w_0}^0)(\nu_i^0 \nu_{w_2}^2) \\ \vdots \\ (\nu_{w_n}^n \nu_{w_0}^0)(\nu_i^0 \nu_{w_n}^n).\end{gathered} \tag{3.59}$$

Due to the subscript constraints placed on group duals by their stacked counterparts (Figure 3.4), it must be the case that $w_0 = i$. Thus the forms further collapse into the identity transpositions

$$\begin{gathered}(\nu_{w_1}^1 \nu_{w_1}^1) \\ (\nu_{w_2}^2 \nu_{w_2}^2) \\ \vdots \\ (\nu_{w_n}^n \nu_{w_n}^n).\end{gathered} \tag{3.60}$$

This applies to all transposition paths, implying that

$$\delta^{n+1} = \epsilon_{01} * \epsilon_{12} * \cdots * \epsilon_{n0} = \beta = \iota \tag{3.61}$$

and that the group dual constraint of each cyclic form of δ^{n+1} is also ι. Therefore $\beta = \iota$. □

This leads to the following important theorem relating assignment feasibility and group duals.

Theorem 3.17 (Feasible Cycle Stack Assignments and Group Duals)**.** *Let δ^{n+1} be the group dual of $C^{(n+1)m}$. Then $\alpha(\delta^{n+1})$ is cyclic if and only if its stacked dual $A(C^{(n+1)m})$ is feasible.*

Proof. For the forward component, a feasible $A(C^{(n+1)m})$ consists of m disjoint cycles C_i, each with a group dual of the form

$$(\nu_i^0 \nu_{w_1}^1)(\nu_{w_1}^1 \nu_{w_2}^2) \dots (\nu_{w_n}^n \nu_i^0) \tag{3.62}$$

By reasoning similar to that of the forward component of the proof for Theorem 3.16, we can show that

$$\alpha(\delta^{n+1}) = \beta = \iota \tag{3.63}$$

and that $\beta = \iota$ for every cyclic forms of $\alpha(\delta^{n+1})$. Therefore, by Theorem 3.16, $\alpha(\delta^{n+1})$ is cyclic.

For the backward component, by the proof of Theorem 3.16, each transposition path δ_i of $\alpha(\delta^{n+1})$ must take the form

$$\delta_i = (v_i^0 v_{w_1}^1)(v_{w_1}^1 v_{w_2}^2)\dots(v_{w_n}^n v_i^0) \tag{3.64}$$

so that the graph dual of δ_i is some cycle C_i in $A(C^{(n+1)m})$ of length $n+1$

$$C_i = e_{iw_1}^{01} - e_{w_1w_2}^{12} - \dots - e_{w_ni}^{n0}. \tag{3.65}$$

Therefore, as there are m transposition paths of length $n+1$ (Lemma 3.15), and so m cycles of length $n+1$, by Theorem 2.23 $A(C^{(n+1)m})$ is feasible. □

This is the primary benefit of working with group duals: feasibility of cycle stack assignments are maintained by requiring that their group dual constraints be the identity element.

Assignment duals are easily extended to entire stacked graphs.

Definition 3.18 (Group Dual of $A(\mathbb{G})$)**.** Let $\mathbb{G}$ be asingular and defined by a collection of k asingular path and cycle stacks $\mathcal{C}_i$ and $\mathcal{P}_j$, of which each may be of different length, such that the union of their edge stack sets is $\mathcal{E}$ of $\mathbb{G}$, and let γ be its group dual. Then, if A is an assignment on $\mathbb{G}$, the vector

$$\alpha(\gamma) :=< \alpha(\gamma_0), \alpha(\gamma_1), \dots, \alpha(\gamma_k) > \tag{3.66}$$

is the (symmetric) group dual of $A(\mathbb{G})$, the system of compositions $\alpha(\gamma_i)$, for all γ_i in γ. If $\alpha(\gamma)$ is the group duals of $A(\mathbb{G})$, then $A(\mathbb{G})$ is the (stacked) graph dual of $\alpha(\gamma)$.

The constrained group dual of $A(\mathbb{G})$ is the vector

$$\beta :=< \beta_0, \beta_1, \dots, \beta_k >, \tag{3.67}$$

with β_i the group dual constraint of $\alpha(\gamma_i)$, such that

$$\alpha(\gamma) = \beta. \tag{3.68}$$

Similarly, if $\mathbb{G}$ is not asingular, then $\alpha(\mathring{\mathbb{G}})$ and $\alpha(\mathring{\gamma})$ are duals of each other with $\mathring{\beta}$ the group dual constraint.

Included in the constrained group dual are all fully and partially constrained edge stack group duals α_{ij}, so that

$$\alpha_{ij} \in \beta_{ij}, \tag{3.69}$$

where β_{ij} is the subset of $\mathbb{S}_m$ elements that α_{ij} is restricted to. In the presence of non-free compositions, the notation $\alpha(\gamma) = \beta$ is one of convenience since the usage of '=' is not strictly correct.

The following two theorems and single definition reformulate the key findings regarding assignment feasibility in terms of graph duals.

Theorem 3.19 (Feasible Assignments and Group Duals)**.** *Let $\mathbb{G}$ be asingular and A an assignment on it, and γ and $\alpha(\gamma)$ their respective group duals, where γ includes the graph dual of every cycle stack in $\mathbb{G}$. Then, $A(\mathbb{G})$ is feasible if and only if, for every δ in γ, $\alpha(\delta)$ is cyclic.*

Proof. By Theorem 3.17 every cycle composition $\alpha(\delta)$ in $\alpha(\gamma)$ has a feasible graph dual in $A(\mathbb{G})$ and every feasible cycle stack assignment $A(C^{nm})$ in $A(\mathbb{G})$ has cyclic group dual $\alpha(\delta)$. By Theorem 2.25, $A(\mathbb{G})$ is therefore feasible and every $\alpha(\delta)$ is cyclic. □

Definition 3.20 (Fundamental Forms of Group Duals)**.** Let $\mathbb{G}$ be asingular, A a feasible assignment on it, and $\mathbb{T}$ a spanning tree stack of $\mathbb{G}$. Then the group dual γ of $\mathbb{G}$ is in *fundamental form* if its path compositions π_i are formed solely on edge stacks in $\tilde{\mathbb{G}}$ while its cyclic compositions δ_i are formed solely by the fundamental cycle stacks in $\mathbb{T}_{\mathcal{C}}$.

If $\mathbb{G}$ is not asingular, then fundamental forms are based on the fringe and fundamental cycles of $\mathring{\mathbb{G}}$.

Theorem 3.21 (Feasible Assignments and Fundamental Forms)**.** *Let γ be the group dual of some $\mathbb{G}$ in fundamental form. Then an assignment A on $\mathbb{G}$ is feasible if and only if, for every δ_i in γ, the group dual $\alpha(\delta_i)$ is cyclic, or, equivalently by Theorem 3.16, if $\alpha(\delta_i) = \iota$.*

Proof. The proof follows directly from Theorems 2.31 and 3.19. □

Theorems 3.19 and 3.21 have rather profound implication. Group duals, in particular those in a fundamental form, provide a means of representing the structure of stacked graphs while preserving assignment feasibility in an elegant form. We are no longer forced to deal with edges, vertices, and network flow values, only group elements and transposition. We also now have access to group theoretic tools with which to study feasibility, and which may provide more efficient optimization techniques than the LP formulation described in Chapter 4 to find the *best* feasible assignment for a given $\mathbb{G}$. Conversely, graph duals introduce a method of modelling at least one class of group theoretic problems in terms of graphs and network flows.

Most forms of problem modelling using stacked graphs will likely involve weighted edges, edge stacks, and assignments.

Definition 3.22 (Weighted Stacked Graphs, Assignments, and Group Duals)**.** For any stacked graph $\mathbb{G}$, a *weighting of* $\mathbb{G}$ assigns, to each $e_{ij}^{pq} \in \mathbb{G}_{\mathcal{G}}$, a real-valued *weight* w_{ij}^{pq}. For any assignment A on E_{pq}, the weight of A_{pq} is

$$w(A_{pq}) := \sum_{e_{ij}^{pq} \in E_{pq}} w_{ij}^{pq} x_{ij}^{pq} \tag{3.70}$$

where x_{ij}^{pq} is the flow value of e_{ij}^{pq} under A_{pq} (Definition 2.15). The weight of an assignment A on $\mathbb{G}$, feasible or otherwise, is

$$w(A(\mathbb{G})) := \sum_{A_{pq} \in A(\mathbb{G})} w(A_{pq}) = \sum_{E_{pq} \in \mathbb{G}} \sum_{e_{ij}^{pq} \in E_{pq}} w_{ij}^{pq} x_{ij}^{pq}. \tag{3.71}$$

If α_{pq} is the group dual of some A_{pq}, the weight of transposition (ij) is

$$w((ij)) := w_{ij}^{pq} \tag{3.72}$$

while the weight of α_{pq} is

$$w(\alpha_{pq}) := \sum_{(ij) \in \alpha_{pq}} w((ij)) = w(A_{pq}) \tag{3.73}$$

3.3 Duality and Non-Simple Stacked Graphs

In Section 2.4 we looked at assignment feasibility for loop and multi-edge stacks, concluding that the only feasible assignment for the former is the identity assignment A^* (Lemma 2.37). For the latter, all feasible assignments on some $E_{pq} - E_{qp}$ are of the form $A_{pq} - A_{qp}$ (Lemma 2.39). The graph duals of both non-simple structures satisfy similar constraints.

Lemma 3.23. *The group dual to any feasible assignment A on a loop stack E_{pp} is ι.*

Proof. See Fact 3.3. □

Lemma 3.24. *Let E_i^{pq} and E_j^{pq} be any two edge stacks between vertex stacks V_p and V_q, so that they form a multi-edge stack. The group duals to any feasible assignments on E_i^{pq} and E_j^{pq} are*

$$a_i^{pq} = a_j^{pq} \; \forall \; i, j. \tag{3.74}$$

That is, if assignments on multi edge stacks are feasible, then each has identical assignments.

Proof. This follows directly from Lemma 2.39. □

Group duals of non-simple stacked graphs degenerate to group duals of simple stacked graphs.

Theorem 3.25. *Let $\mathbb{G}$ be an asingular, non-simple stacked graph with group dual γ. In addition, let $\mathbb{G}'$ be the sub-graph of $\mathbb{G}$ formed by removing all loop stacks and all but one edge stack in every multi-edge stack, and let γ' be it's group dual. Then the group dual $\alpha'(\gamma')$ of any feasible assignment of A' on $\mathbb{G}'$ generates a unique group dual $\alpha(\gamma)$.*

Proof. The proof is similar to that of Theorem 2.40: the only group dual of feasible assignments on loop stacks is $\imath$ (Lemma 3.23) while the group dual of a single edge stack α_k^{pq} in multi-edge stack generates the (unique) group dual of each other α_i^{pq} (Lemma 2.39). □

Chapter 4

Linear Programming: Communication, Capacity, and Contiguity

In this chapter, we construct a linear program (LP) which satisfies the constraints of Theorem 2.31 and whose optimal solution is a minimally weighted feasible assignment. A basic knowledge of linear programming and network flows is assumed.

Let $\mathbb{G}$ be a weighted stacked graph. An assignment A on $\mathbb{G}$ is optimal if it provides a solution to the following objective function

$$\min \sum_{A_{pq} \in A(\mathbb{G})} \sum_{e_{ij}^{pq} \in E_{pq}} w_{ij}^{pq} x_{ij}^{pq} \tag{4.1}$$

with $x_{ij}^{pq} \in \{0, 1\}$, which will be shorthanded to

$$\min \sum w_{ij}^{pq} x_{ij}^{pq}. \tag{4.2}$$

Before we discuss the linear constraints that apply Equation 4.1, we should note that there are different 'scales' of 'levels' of optimization that can be conducted. Given the size of $\mathbb{G}$ and its relative complexity in terms of number of layers, regions, and proportion of singular to asingular edge stacks, any one of three general optimization approaches can be used: *local*, *regional*, or *global*.

4.1 Local, Regional, and Global Optimization

Local optimization involves finding minimal weightings to each singular and asingular edge stack assignments A_{pq} independent of all others. It is equivalent to solving a collection of bipartite matching problems, for which there exist network-flow algorithms far more efficient than a standard LP formulation [AMO93]. As each A_{pq} is independent, assignments can be solved in parallel. The difficulty is that

the set of edge stack assignments, taken as the definition for a single assignment on $\mathbb{G}$, may not be feasible.

Regional optimization refers to optimizing over the regions $\mathbb{G}_{\mathcal{R}}$ of $\mathbb{G}$. Regional assignments $A(\mathcal{R}_i)$ for each $\mathcal{R}_i \in \mathbb{G}_{\mathcal{R}}$ are solved for independently, as with local optimization. The key difference is that assignment feasibility needs to be considered: at the local level, assignments are on path stacks of length one and so all assignments are feasible (Corollary 2.19), where this is rarely the case at the regional level. Assignments on asingular edge stacks are solved for individually and independently, as at the local level.

The last level of optimization, *global*, solves the entirety of $A(\mathbb{G})$ at once - every A_{pq} is solved for simultaneously, including asingular edge stacks, and any non-basic singular constraints are enforced (Definition 2.41).

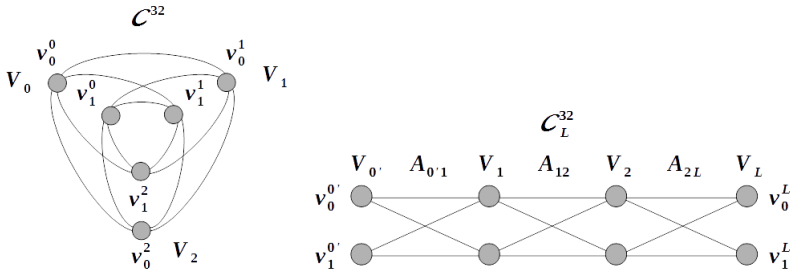
If singularity constraints are basic and either regional or global optimization is used, the process can be reduced to three distinct, independent stages: *local singular*, *local fringe*, and *core*. Local singular applies local optimization to each $E_{pq} \in \mathring{\mathbb{G}}$, local fringe does the same for $\tilde{\mathbb{G}}$, while core optimizes over the fundamental cycles in $\mathring{\mathbb{G}}$. The partitions of $\mathcal{E}$ for any $\mathbb{G}$ (Lemma 2.14) thus partition the optimal feasible assignment problem.

In general, only global optimization is guaranteed to yield an optimal solution to Equation 4.1. However, local and regional approaches are expected to be faster and, depending on the structure of $\mathbb{G}$ and underlying problem, produce result that, if not optimal, may near enough to optimality to be useful. In the LP constructs below, local core optimization is conducted first, followed by the ULA non-basic constraint.

4.2 The 3C Constraints of Assignment Feasibility

There are three kinds of linear constraints required to ensure assignment feasibility on asingular stacked graphs, such as $\mathring{\mathbb{G}}$: *communication*, *capacity*, and *contiguity*. Communication refers to where the disjoint paths of a feasible assignment on path and cycle stacks start and stop, also known as asingular constraints (Definition 2.33). As $A(C^{nm})$ is fully constrained for every C^{nm} (Lemma 2.34), every path in $A(C^{nm})$ must start and end on the same vertex. In effect, any start vertex 'communicates' with itself. For any path stack $\mathcal{P}^{nm}$, the start and end vertices of the connected components of $A(\mathcal{P}^{nm})$ may or may not be different. In our LP construction, we shall assume that path stacks are unconstrained, as constraints are not required for feasibility.

The necessity that connected components of feasible assignments are disjoint requires that each edge in an edge stack has a 'capacity' of 1 [AMO93]. Path and cy-



(a) (b)

Figure 4.1: C^{32} as (a) a standard cycle stack and (b) as a linear cycle stacks C_L^{32}.

cle stacks that are joined (Theorem 2.27) are required to have identical assignments where they are 'contiguous'.

The linear constraints for communication, capacity, and contiguity are described below. In their constructions, we treat cycle stacks as a variation of a path stack. These *linear* cycle stacks are easier to read from a diagrammatic perspective and illustrate how communication constraints can be put on path stacks.

Definition 4.1 (Linear Cycle Stacks)**.** For any asingular cycle stack

$$C^{nm} = E_{ab} - E_{bc} - \cdots - E_{yz} - E_{za} \tag{4.3}$$

split V_a into two vertex stacks $V_{a'}$ and V_L, so that E_{ab} becomes $E_{a'b}$ and E_{za} becomes E_{zL}. This converts C^{nm} into

$$C_L^{nm} := E_{a'b} - E_{bc} - \cdots - E_{yz} - E_{zL}, \tag{4.4}$$

where C_L^{nm} is a *linear cycle stack*, specifically, the *linearisation of* C^{nm}. If A is an assignment on C_L^{nm}, then $A(C_L^{nm})$ is defined to be feasible if $v_i^{a'}$ is path connected to v_j^L for all $i = j$ and disconnected for all $i \neq j$.

The weight of $e_{ij}^{a'b}$ and e_{ij}^{zL} are

$$w_{ij}^{a'b} = w_{ij}^{ab} \text{ and } w_{ij}^{zL} = w_{ij}^{za}. \tag{4.5}$$

Figures 3.3 and 4.1 demonstrate linear cycle stacks.

The definition of feasibility for linear cycle stacks satisfies Theorem 2.23 and is a special case of Theorem 2.19. Assignments on linear cycle stacks are an example of fully constrained path stack assignment (Definition 2.33).

In what follows, we'll assume that a spanning tree stack $\mathbb{T}$ has been found for $\mathring{\mathbb{G}}$ as well as a cycle stack basis $\mathbb{T}_C$, and that each cycle stack is in $\mathbb{T}_C$. Though discussed in the context of a single cycle stack C^{nm}, the constraints below apply to every cycle stack in $\mathbb{T}_C$.

4.2.1 Communication

For $A(C_L^{nm})$ to be feasible, a path must exist between $v_i^{a'}$ and v_i^L. As we're looking for minimally weighted paths, the condition can be modelled as a minimal path length LP problem, treating $v_0^{a'}$ as the source vertex and v_0^L as the sink vertex [AMO93]

$$\sum_{j=0}^{m-1} x_{ij}^{pq} - \sum_{j=0}^{m-1} x_{ji}^{qp} = \begin{cases} 1, & \text{if } i = 0 \text{ and } p = a' \\ -1, & \text{if } i = 0 \text{ and } q = L \\ 0, & \text{otherwise} \end{cases}, \tag{4.6}$$

$$\forall i \in 0, \ldots, m-1,\ \forall E_{pq} \in C_L^{nm}.$$

The same is true for the remaining $m-1$ paths. Modelling distinct source-sink pairs on a single network is problematic: if paths are not disjoint, then information concerning where each path starts and stops is lost (Figure 4.2).

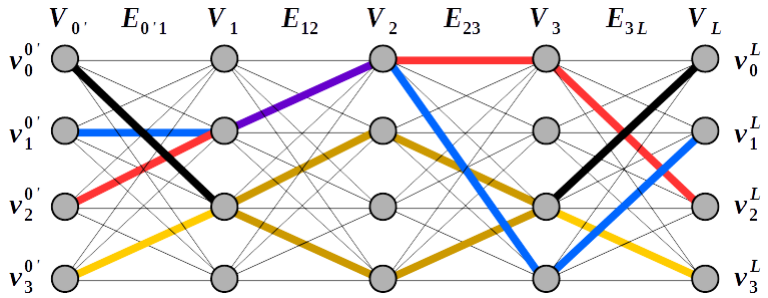


Figure 4.2: The four shortest paths for some C^{44}. Although the blue and red paths share edge e_{10}^{12}, by backtracking from v_1^L and v_2^L it's possible to extract both paths. In the case of path components $e_{21}^{12} - e_{12}^{23}$ and $e_{23}^{12} - e_{32}^{23}$, this is not possible: which is black and which yellow?

Path information can be maintained by treating paths as distinct commodities in a multi-commodity flow problem [AMO93]. To keep things simple and uni-commodity, we'll instead create m identical instances of C^{nm} (and therefore of C_L^{nm}), one for each path, and model a shortest path algorithm on each.

$$\sum_{j=0}^{m-1} {}^{k}x_{ij}^{pq} - \sum_{j=0}^{m-1} {}^{k}x_{ji}^{qp} = \begin{cases} 1, & \text{if } i = k \text{ and } p = a' \\ -1, & \text{if } i = k \text{ and } q = L \\ 0, & \text{otherwise} \end{cases},$$
$$\forall i \in 0, \ldots, m-1,\ \forall E_{pq} \in C_L^{nm},\ \forall k = 0, \ldots, m-1. \tag{4.7}$$

The k^{th} path is defined to begin on vertex k.

Each of the m copies of C_L^{nm} have identical edge weights. The optimal solution for the shortest path problem for each of the m paths will therefore be the same as if they were solved on a single C_L^{nm}. However, the third index k allows us to maintain communication information (Figure 4.3).

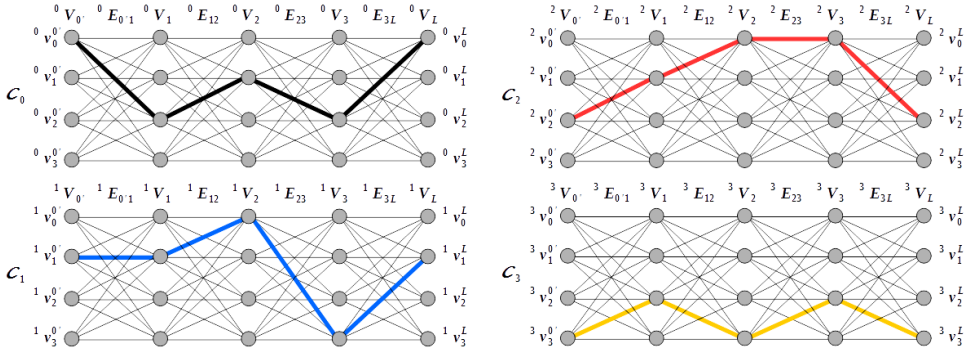


Figure 4.3: The same solution as Figure 4.2, but modelled on four copies of C^{44}: C_0, C_1, C_2, and C_3. All paths are well-defined.

4.2.2 Capacity

Though the m copies of C_L^{nm} technically give us disjoint paths, they are not disjoint in the original C_L^{nm} (Figure 4.2). To ensure disjointedness, we introduce the capacity constraint, which limits each edge to be used at most once throughout all m copies of C_L^{nm}

$$\sum_{k=0}^{m-1} {}^{k}x_{ij}^{pq} \leq 1,\ \forall i, j \in \{0, \ldots, m-1\},\ \forall E_{pq} \in C_L^{nm}. \tag{4.8}$$

4.2.3 Contiguity

Finally, we generalize ${}^{k}x_{ij}^{pq}$ to ${}_{r}^{k}x_{ij}^{pq}$ so as to tie together all joined cycle stacks in $\mathbb{T}_C$. Let C_r and C_s be any two cycle stacks in $\mathbb{T}_C$ which share common set $\mathcal{E}$ of edge stacks[28]. Then the shared edge flows (assignments) are the same if

$$ {}_{r}^{k}x_{ij}^{pq} = {}_{s}^{k}x_{ij}^{pq} = x_{ij}^{pq},\ \forall i,j,k \in \{0,1,\dots,|E_{pq}|\},\ \forall E_{pq} \in \mathcal{E}. \tag{4.9} $$

4.2.4 Asinglular LP Problem Statement

Combining the 3C constraints yields the following general LP for finding the minimally weighted feasible assignment for any $\mathring{\mathbb{G}}$

$$ \min \sum w_{ij}^{pq} x_{ij}^{pq} \tag{4.10} $$

s.t.

$$ \sum_{j=0}^{|E_{pq}-1|} {}_{r}^{k}x_{ij}^{pq} - \sum_{j=0}^{|E_{pq}-1|} {}_{r}^{k}x_{ji}^{qp} = \begin{cases} 1, & \text{if } i=k \text{ and } p=a' \\ -1, & \text{if } i=k \text{ and } q=L\ , \\ 0, & \text{otherwise} \end{cases} \tag{4.11} $$

$$ \forall\ i,k \in \{0,1,\dots,|E_{pq}|\},\ \forall\ E_{pq} \in C_r, \quad \forall C_r \in \mathbb{T}_C \tag{4.12} $$

$$ \sum_{k=0}^{|E_{pq}-1|} {}_{r}^{k}x_{ij}^{pq} \leq 1, \quad \forall\ i,j \in \{0,1,\dots,|E_{pq}|\},\ \forall\ E_{pq} \in C_r, \quad \forall C_r \in \mathbb{T}_C \tag{4.13} $$

$$ {}_{r}^{k}x_{ij}^{pq} = {}_{s}^{k}x_{ij}^{pq} = x_{ij}^{pq}, \quad \forall\ i,j,k \in \{0,1,\dots,|E_{pq}|\},\ \forall\ C_r, C_s \in \mathbb{T}_C, \quad \forall E_{pq} \in C_r \text{ and } C_s \tag{4.14} $$

$$ x_{ij}^{pq} \in \{0,1\}. \tag{4.15} $$

The LP does not account for fully or partially constrained path stacks.

The 3C constraints are sufficient conditions to ensure that any feasible solution to the LP is also an assignment on $\mathbb{G}$.

Theorem 4.2 (3*C* Constraints Produce Feasible $A(\mathring{\mathbb{G}})$)**.** *The* 3*C constraints define a system of linear constraints which produce feasible assignments on any* $\mathring{\mathbb{G}}$.

[28] i.e. They're 'joined' in the sense of Lemmas 2.26 and 2.27.

Proof. The communication constraint produces m paths on C_L^{nm}, where each path i starts on $v_i^{a'}$ and ends on v_i^L. The capacity constraint ensures that the paths are disjoint. That an assignment is produced on each E_{pq} is clear as $|E_{pq}| = m$, there are m edges in E_{pq} with non-zero flow values, and those edges are all disjoint. This implies a one-to-one, onto mapping from the vertices V_p to those of V_q. This is the definition of an asingular edge stack assignment (Definition 2.15). Thus, communication and capacity produce assignments on C_L^{nm}. By the construction of linear cycle stacks (Definition 4.1), this implies that each path is a cycle in C^{nm}. By Corollary 2.20 and the construction of C_L^{nm}, the disjoint paths in C_L^{nm} cycles form a disjoint cover cycle of C^{nm}. As the disjoint cover cycle is also an assignment, the assignment is feasible by Theorem 2.23.

The contiguity constraint ensures that joined asingular cycle stacks have the same assignments on shared edge stacks. By Lemmas 2.26 and 2.27, joinedness preserves feasibility. Therefore, the 3C constraints define linear constraints which produce feasible assignments on any $\mathring{\mathbb{G}}$. □

The above LP for $A(\mathring{\mathbb{G}})$ is a form of restricted regional optimization, applying only to cyclic edge stacks (Definition 2.12). There is no need to apply it to the entirety of $\mathring{\mathbb{G}}$ as the same optimal feasible assignment can be found by restricting it to each $\mathcal{R}_i \in \mathbb{G}_{\mathcal{R}}$.

4.2.5 Size Complexity of Asingular LP

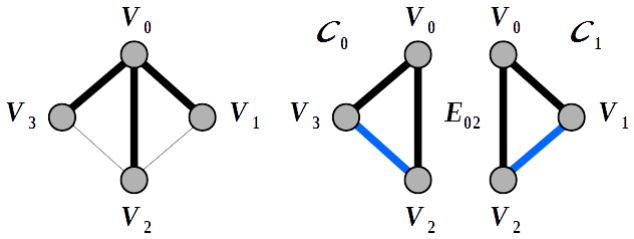
Our LP introduces a large number of additional edges. Linearisation of cycle stacks adds at most $|\mathbb{T}_C|$ edge stacks while communication constraints require at most

$$m = \max\{|C||C \in \mathbb{T}_C\} \tag{4.16}$$

copies of $\mathcal{E}$, yielding a total of

$$m(|\mathcal{E}| + |\mathbb{T}_C|) \tag{4.17}$$

edge stacks. Capacity constraints do not add additional edge stacks, though contiguity constraints introduce at most $|\mathbb{T}_C|$ additional copies of $\mathcal{E}$, since it's possible for an edge stack $E_{pq} \in \mathcal{E}$ to occur in every $C_i \in \mathbb{T}_C$ (Figure 4.4). Finally, the largest edge stack in $\mathring{\mathbb{G}}$ is also m. The size complexity V_{LP}, in terms of number of vertices, required to create the graph necessary to model our LP for some $\mathring{\mathbb{G}}$ is



(a) (b)

Figure 4.4: (a) Spanning tree $\mathbb{T}_{\mathbb{G}}$ produces two fundamental cycle stacks (b) $\mathcal{C}_0$ and $\mathcal{C}_1$, both containing E_{02}.

therefore bounded above by

$$\begin{aligned} V_{LP} &\leq \mathcal{O}\left(m^2(|\mathbb{T}_C| + m)(|\mathcal{E}| + |\mathbb{T}_C|)\right) && (4.18)\\ &= \mathcal{O}\left(m^2(|\mathbb{T}_C||\mathcal{E}| + |\mathbb{T}_C|^2 + m|\mathcal{E}| + m|\mathbb{T}_C|)\right) && (4.19)\\ &\leq \mathcal{O}\left(m^2(|\mathcal{E}|^2 + |\mathcal{E}|^2 + m|\mathcal{E}| + m|\mathcal{E}|)\right) && (4.20)\\ &= \mathcal{O}\left(m^2|\mathcal{E}|^2 + m^3|\mathcal{E}|\right) && (4.21)\\ &\leq \mathcal{O}\left(m^2|\mathcal{V}|^4 + m^3|\mathcal{V}|^2\right) && (4.22)\\ &\leq \mathcal{O}\left(m^3|\mathcal{V}|^4\right). && (4.23) \end{aligned}$$

The replacement of $|\mathbb{T}_C|$ with $|\mathcal{E}|$ in Equation 4.20 is made as at most every edge stack is restricted[29]. The replacement in Equation 4.22 follows from the fact that the number of edge stacks are bound by $|\mathcal{N}|^2$, just as the number of edges in a simple graph are bound by the square of the number of vertices[30].

We can therefore model the assignment problem for $\mathring{\mathbb{G}}$ with size complexity of $\mathcal{O}\left(m^3|\mathcal{V}|^4\right)$. In the case of nucleic acid phosphate backbones, we'll see that $m \leq 16$ or infinite, so that the constant factor is at most 256. As stacked graphs have only been defined for the finite case, infinite values of m are ignored.

[29]Every edge stack is restricted if every edge stack forms a loop stack. In this case, however, there is no optimization problem in light of Lemma 2.37

[30]There are at most $\binom{N}{2}$ edges in a simple graph consisting of N vertices, with

$$\mathcal{O}\left(\binom{N}{2}\right) = \mathcal{O}\left(\frac{N(N-2)}{2}\right) = \mathcal{O}\left(N^2\right). \quad (4.24)$$

4.3 Modelling Singular Constraints: ULA

The only non-basic constraint we've considered is ULA (Definition 2.42). It's a fortunate consequence of our construction of the contiguity constraint that the index k, which indicates the copy number of a cycle stack $\mathcal{C}^{nm}$, also represents the *layer* number (Theorem 2.32) for the region to which $\mathcal{C}^{nm}$ belongs. Equation 4.9 forces shared edge stacks for two joined cycle stacks to have the same flow value *for the same copy* k; contiguity is defined on *copies* of cycle stacks, not the stacks themselves. Thus, all connected edges in $A(\mathring{\mathbb{G}})$ have the same copy index k if they belong to the same region. Since the flow value is the same for k for a given region regardless of the cycle stack number s, the index can be dropped to produce

$$^{k}x_{ij}^{pq}.$$

Once $A(\mathring{\mathbb{G}})$ has been found, we no longer need the cycle stack copies created to for the communication constraint.

Modelling ULA constraints as an LP requires us to formulate constraints as binary logical ones. Casting techniques are from Discrete Optimization course notes.

If a vertex is in layer k of $\mathcal{R}_r$, then it's incident on one or two edges, otherwise it's incident on none[31].

$$\sum_{j=0}^{|\mathcal{R}_r|-1} {}^{k}x_{ij}^{pq} + \sum_{j=0}^{|\mathcal{R}_r|-1} {}^{k}x_{ji}^{qp} \in \{0,1,2\},\ \forall k = 0,\dots,|\mathcal{R}_r|-1, \forall E_{pq} \in \mathcal{R}_r \tag{4.25}$$

We next cast Equation 4.25 into binary form so that its solution is either 0 or 1. The constraints for the *layer indicator value* ${}^{k}y_i^p$ *of* $v_i^p \in \mathcal{H}_k^r$ are

$$0 \le {}^{k}y_i^p \le 1 \tag{4.26}$$

$${}^{k}y_i^p \le \sum_{j=0}^{|\mathcal{R}_r|-1} {}^{k}x_{ij}^{pq} + \sum_{j=0}^{|\mathcal{R}_r|-1} {}^{k}x_{ji}^{qp} \tag{4.27}$$

$$\sum_{j=0}^{|\mathcal{R}_r|-1} {}^{k}x_{ij}^{pq} + \sum_{j=0}^{|\mathcal{R}_r|-1} {}^{k}x_{ji}^{qp} \le 2 \cdot {}^{k}y_i^p \tag{4.28}$$

where the summation conditions are those of Equation 4.25. Replacing the summations with 1 or 2 forces ${}^{k}y_i^p = 1$ while a value of 0 forces ${}^{k}y_i^p = 0$.

We also want to know whether two layers $\mathcal{H}_a^r$ and $\mathcal{H}_b^t$ from adjacent regions $\mathcal{R}_r$ and $\mathcal{R}_t$ (Definition 2.11) are themselves adjacent through A_{pq} for a given $E_{pq} \in \mathring{\mathcal{E}}_{rt}$.

[31] Recall that cycle stacks are in *linear form*, hence the existence of vertices incident on only single edges in a path.

First, we create a new variable that indicates when ${}^{a}y_i^p$ and ${}^{b}y_j^q$ are simultaneously true

$$ {}^{ab}y_{ij}^{pq} := {}^{a}y_i^p \wedge {}^{b}y_j^q \tag{4.29} $$

with constraints

$$ -{}^{a}y_i^p + {}^{ab}y_{ij}^{pq} \leq 0 \tag{4.30} $$

$$ -{}^{b}y_j^q + {}^{ab}y_{ij}^{pq} \leq 0 \tag{4.31} $$

$$ {}^{a}y_i^p + {}^{b}y_j^q - {}^{ab}y_{ij}^{pq} \leq 1. \tag{4.32} $$

When both ${}^{a}y_i^p = {}^{b}y_j^q = 1$, ${}^{ab}y_{ij}^{pq} = 1$. Otherwise, ${}^{ab}y_{ij}^{pq} = 0$. Next, we note that

$$ {}^{ab}y_{ij}^{pq} \cdot x_{ij}^{pq} = 1 \tag{4.33} $$

precisely when v_i^p is incident on v_j^q under A_{pq}, $\mathrm{h}(v_i^p) = a$, and $\mathrm{h}(v_j^q) = b$. Equation 4.33 therefore indicates whether $\mathcal{H}_a^r$ is incident on $\mathcal{H}_b^t$ through A_{pq}.

The definition of ULA requires that *all* vertices in $\mathcal{H}_a^r$ that are adjacent on $\mathcal{R}_t$ are all adjacent on the same $\mathcal{H}_b^t$. Since there are $|\hat{\mathcal{E}}_{rt}^{\cdot}|$ edge stacks between $\mathcal{R}_r$ and $\mathcal{R}_t$, it follows that

$$ \sum_{E_{pq} \in \hat{\mathcal{E}}_{rt}^{\cdot}} \sum_{i=0}^{|V_p|-1} \sum_{j=0}^{|V_q|-1} {}^{ab}y_{ij}^{pq} \cdot x_{ij}^{pq} \in \{0, |\hat{\mathcal{E}}_{rt}^{\cdot}|\}, \forall a \in 0, \ldots, |\mathcal{R}_r| - 1, b \in 0, \ldots, |\mathcal{R}_t| - 1. \tag{4.34} $$

As the equation is non-liner, each product

$$ {}^{ab}y_{ij}^{pq} \cdot x_{ij}^{pq} $$

must be replaced with an indicator value similar to Equation 4.29. Let

$$ {}^{ab}h_{ij}^{pq} := {}^{ab}y_{ij}^{pq} \wedge x_{ij}^{pq} \tag{4.35} $$

with constraints

$$ -{}^{ab}y_{ij}^{pq} + {}^{ab}h_{ij}^{pq} \leq 0 \tag{4.36} $$

$$ -x_{ij} + {}^{ab}h_{ij}^{pq} \leq 0 \tag{4.37} $$

$$ {}^{ab}y_{ij}^{pq} + x_{ij}^{pq} - {}^{ab}h_{ij}^{pq} \leq 1 \tag{4.38} $$

so that Equation 4.34 can be written as

$$ \sum_{E_{pq} \in \hat{\mathcal{E}}_{rt}^{\cdot}} \sum_{i=0}^{|V_p|-1} \sum_{j=0}^{|V_q|-1} {}^{ab}h_{ij}^{pq} \in \{0, |\hat{\mathcal{E}}_{rt}^{\cdot}|\}, \forall a \in 0, \ldots, |\mathcal{R}_r| - 1, b \in 0, \ldots, |\mathcal{R}_t| - 1. \tag{4.39} $$

As $|\hat{\mathcal{E}}_{rt}|$ is fixed, Equation 4.39 can be normalized to

$$\sum_{E_{pq} \in \hat{\mathcal{E}}_{rt}} \sum_{i=0}^{|V_p|-1} \sum_{j=0}^{|V_q|-1} \frac{1}{|\hat{\mathcal{E}}_{rt}|} {}^{ab}h_{ij}^{pq} \in \{0,1\},\ \forall a \in 0,\dots,|\mathcal{R}_r|-1, b \in 0,\dots,|\mathcal{R}_t|-1. \tag{4.40}$$

Let h_{ab}^{rt} be the summation in Equation 4.40 for given a and b so that

$$h_{ab}^{rt} = \{0,1\}. \tag{4.41}$$

The variable h_{ab}^{rt} is the *layer adjacency indicator*, with $h_{ab}^{rt} = 1$ if $\mathcal{H}_a^r$ and $\mathcal{H}_b^s$ are mutually uniformly adjacent, and $w_{ab}^{rt} = 0$ if $\mathcal{H}_a^r$ and $\mathcal{H}_b^t$ are completely and mutually non-adjacent.

Equations 4.25 to 4.41 define the ULA non-basic singularity constraint for every pair of regions $\mathcal{R}_r$ and $\mathcal{R}_t$ with a non-empty boundary $\hat{\mathcal{E}}_{rt}$ in $A(\mathbb{G})$, where h_{ab}^{rt} indicates when two layers $\mathcal{H}_a^r$ and $\mathcal{H}_b^t$ are uniformly adjacent to each other. We shall not conduct an analysis ULA's space complexity.

It's possible to reduce the number of ULA constraints. Equation 4.34 becomes non-linear if $A(\mathring{\mathbb{G}})$ is solved for first prior to enforcing ULA on $A(\hat{\mathbb{G}})$. The term ${}^{ab}y_{ij}^{pq}$ becomes a fixed binary value as the layers to which $v_i^p \in \mathcal{R}_r$ and $v_j^q \in \mathcal{R}_s$ are are explicitly solved for in $A(\mathring{\mathbb{G}})$. Equations 4.35 through 4.38 can be removed. In general, however, we cannot solve for $A(\tilde{\mathbb{G}})$, $A(\mathring{\mathbb{G}})$, and $A(\hat{\mathbb{G}})$ independently and assume the resulting global solution $A(\mathbb{G})$ is optimal. Figure 4.5 illustrates this with a simple mixed $\mathbb{G}$. Whether or not optimal $A(\mathbb{G})$ is solved for globally or through local iterations (Section 4.1) will depend on the number of vertices and edges of $G_{\mathcal{G}}$ as well as whether $A(\mathbb{G})$ must be optimal or only reasonably so.

This ends our discussion of stacked graphs. In the following chapters we turn to the structure of RNA and the process of writing the conformation problem as an optimal feasible assignment on stacked graphs.

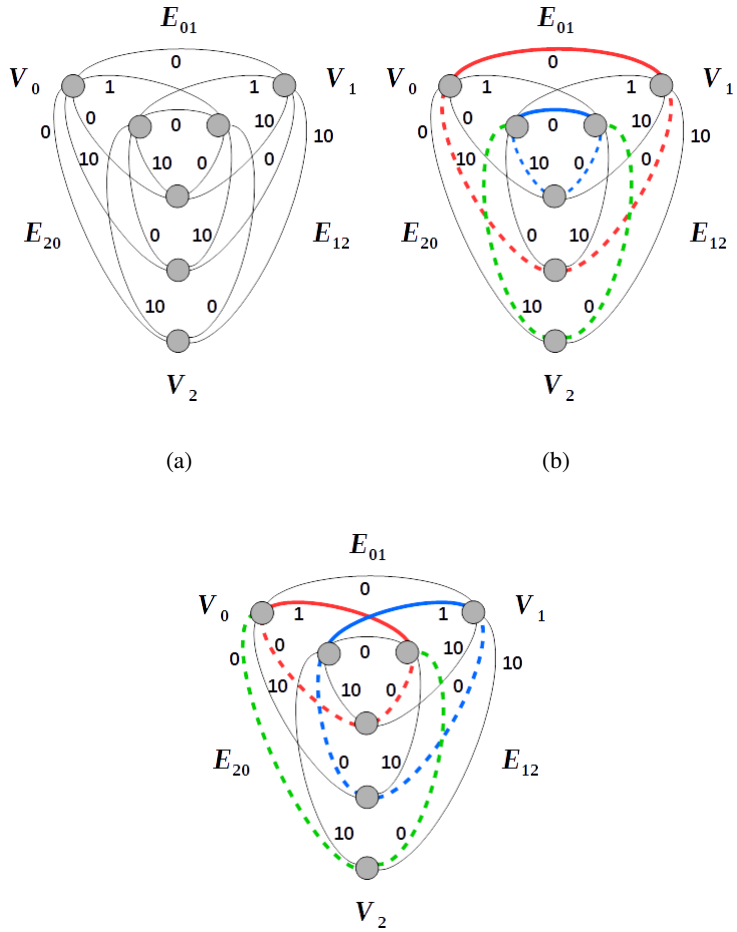


Figure 4.5: (a) Weighted $\mathbb{G}$ defined by mixed $\mathcal{C}^3$ with asingular E_{01} and singular E_{12} and E_{20}. (b) The optimal assignment on E_{01} (solid lines) has a weight of 0. However, all assignments on E_{12} and E_{20} must consist of forms similar to those in dashed lines in order to satisfy ULA. In each case, the total weight of A_{12} and A_{20} is always 30. In (c), however, A_{01} has a weight of 2 while A_{12} and A_{20} total to 0. Thus optimization over $A(\tilde{\mathbb{G}})$ first yields a different solution than optimizing over $A(\hat{\mathbb{G}})$ first.

Chapter 5

Kinematic Chains

Kinematic chains are mathematical models of rigid objects interconnected by joints, or 'lower pairs', that allow for a variety of relative motions. Their primary application is in mechanical engineering, particularly robotics. They can, however, be used to model chemical molecules by treating bonds as rigid structures and atomic rotation about bonds as rotational joints.

The literature on kinematic chains is vast and deep. The aim of this chapter is simply to introduce their general structure and demonstrate how RNA can be modelled as one, including only those concepts relevant to our project.

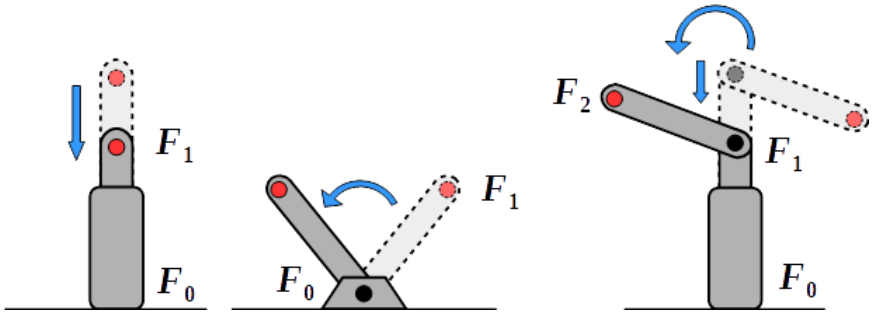
The following discussion is based heavily on [Ang07] and [McC90]. The term *chain* will be used as a shorthand for 'kinematic chain' throughout the remainder of this work.

5.1 Denavit-Hartenberg Parameters

Kinematic chains are defined primarily by the details of their joints. A *prismatic pair*, P, produces translational motion between successive reference frames F_i and F_{i+1} (Figure 5.1(a)) while a *rotational pair*, R, provides a rotational transformation (Figure 5.1(b)). Composition of pairs produces both translational and rotational transformations between frames F_i and F_{i+2} (Figure 5.1(c)). The rigid components being modelled are implied by the relative location and orientation of adjacent reference frames. The composite chain in Figure 5.1(c) is termed a PR chain while its *sub-chains* are P and R chains, respectively. In general, chains are named after ordered, serial sequence of their joints, with repeated sub-sequences replaced by a numeric value. Thus, a chain formed by three prismatic joints followed by a single rotational joint is a $PPPR$ chain, or a $3PR$ chain. The last frame of reference in a chain is commonly termed the *end-effector* and will, in this work, be denoted by a red circle.

There are a number of other joint types, such as helical and spherical, however all can be reduced to combinations of prismatic and rotational [Ang07]. In addition, non-serial or *parallel* chains are possible (Figure 5.2).

Another categorization relates to the space in which the chain exists. *Planar*



(a) Prismatic Pair (b) Rotational Pair (c) PR Chain

Figure 5.1: The two fundamental kinematic pairs: (a) prismatic, P, and (b) rotational, R. P pairs produce translational motion while R pairs produce rotational motion, with each providing a single degree of freedom. Their composition (c) results in a PR chain, which has two degrees of freedom. The end-effector is indicated in red.

chains are where the entire range of motion for the end-effector exists in a plane, while for *spherical* chains it exists on a sphere. *Spatial* chains have no such restrictions. Finally, *open* chains are those wherein the end-effector is free to move, while in *closed* chains the final reference frame is fixed [McC90].

In our study of RNA, we will concern ourselves solely with $6R$, serial, spatial open chains. However, we will have recourse to $2R$ and $3R$ chains to illustrate certain concepts due to the ease in visualizing their motions.

There are a variety of mathematical treatments for chains, including the use of quaternions and screws [McC90]. The most common approach involves standard matrix transforms, which is what we shall use here as the particular notation is of little consequence to us and matrices are more widely known.

Matrix representations of lower pairs are constructed using *Denavit-Hartenberg* parameters. In the context of purely rotational chains, the parameters are defined as follows [Ang07]:

- a_i, the distance between the primary axis, Z_i, of joints i and $(i+1)$
- α_i, the angle or *twist* between the primary axis of joints i and $(i+1)$
- b_i, the Z_i-coordinate or *offset* between the intersection of Z_i and X_{i+1}

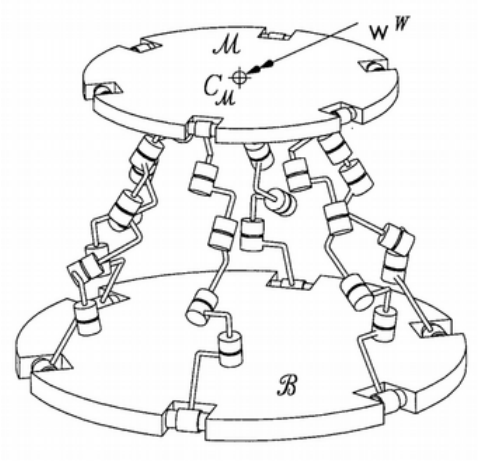


Figure 5.2: An example of a complex parallel kinematic chain with six 'legs' [Ang07].

– θ_i, variable torsion angle about the x-axis of rotational joint i, with respect to positive Z_i.

Figure 5.3 illustrates the parameters for a $6R$ spatial chain.

The matrix formation of the i^{th} rotational pair is

$$Q_i(\theta_i) = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & b_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.1}$$

Each set of parameter values defines a unique pair. Composition of pairs is accomplished via the production of transformation matrices. In the case of a $6R$ chain, we have

$$Q := Q_1 Q_2 Q_3 Q_4 Q_5 Q_6, \tag{5.2}$$

where Q encodes the position and orientation of the final reference frame F_6 (i.e. the end-effector) with respect to the base frame, ${F_0}^{32}$. Q, being the product of six single-variable functions, can be treated as a single six-variable function

$$Q(\theta) := Q(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6). \tag{5.3}$$

[32]This explains why variable indexing begins at 1 and not 0.

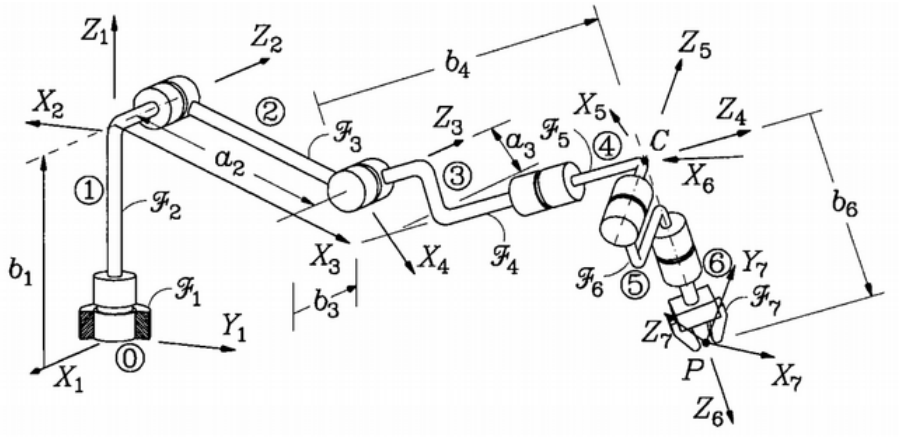


Figure 5.3: The Denavit-Hartenberg parameters associated with a $6R$ chain. Missing are the twist angles. In the case of α_3, this would be the angle between rotational axes Z_3 and Z_4. From [Ang07].

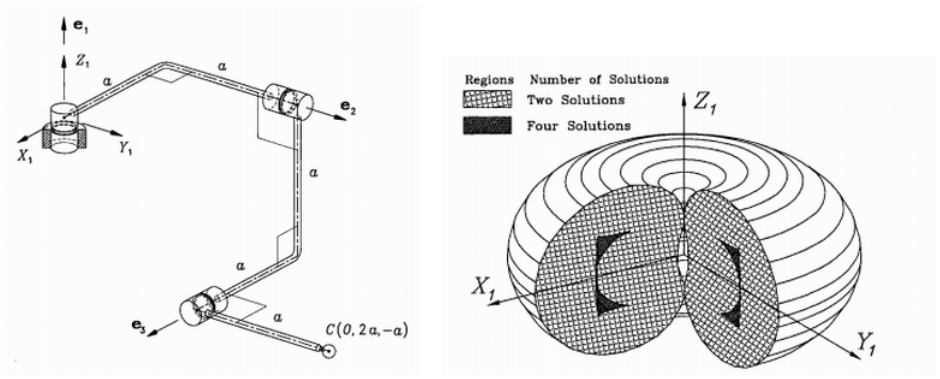
Every chain has an associated *workspace* which describes the locations and orientations reachable by its end-effector [Ang07]. Like a human arm, 'mathematical arms' have limited mobility. As we have difficulties touching the point of our backs between the shoulder blades, kinematic chains have difficulties touching portions of space around them. The $3R$ in Figure 5.4(a), for example, cannot place its end-effector immediately above or below the base reference fame, indicated by the origin.

However, like human arms, some locations of space are easier to reach than others. There are multiple sets of angles for shoulder, elbow, and wrist joints that place our hands in front of our face with a given orientation. So too with kinematic chains. The workplace of a chain is divided into distinct *regions*. All end-effector positions, which are encoded in Q, that lay within a given region have the same number of torsion solutions

$$\theta := \langle \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 \rangle \tag{5.4}$$

that produce their respective Q.

For example, the $3R$ chain workspace in Figure 5.4(b) is divided into 2 regions. There exist two sets of torsion angles $\theta = \langle \theta_1, \theta_2, \theta_3 \rangle$ and $\theta' = \left\langle \theta'_1, \theta'_2, \theta'_3 \right\rangle$ that would place the end-effector anywhere within the outer region, while four exist for the central region.



(a) (b)

Figure 5.4: (a) A $3R$ chain and (b) its workspace. The shaded sections of the workplace indicate different regions. From [Ang07].

5.2 Forward and Inverse Kinematics

There are two general problems associated with kinematic chains, termed the *forward* and *inverse kinematic* problems, or FK and IK. In the context of $6R$ chains, FK asks: 'Given a set of torsion angles $\theta =< \theta_1, \ldots, \theta_6 >$, what is Q?'[33]. The solution is trivial - evaluate Q at θ.

The inverse problem asks: 'Given Q, what is the *set of all* θ that produce Q?'. This question, like other inverse problems, has an exceptionally non-trivial answer.

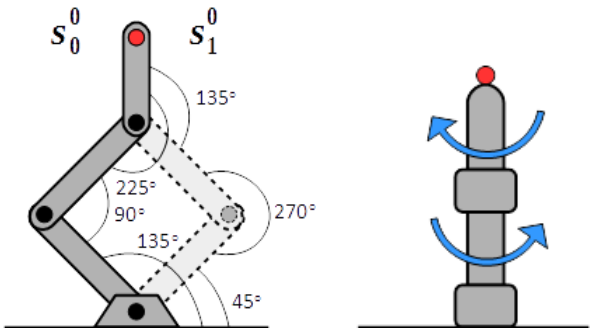
Fortunately, for our $6R$ case, general solutions do exist. For $6R$ chains, there exist either a) at most 16 solutions or b) infinite solutions to the IK problem for a given Q [Ang07]. Figures 5.5(a) and 5.5(b) give an example of finite and infinite solutions, respectively.

The workspace of any chain is a subspace of $\mathbb{R}^3 \times [0°, 360°)^3$. The real components $\mathbb{R}$ reflect the three Cartesian coordinates of the end-effector while the angular components $[0°, 360°)$ represent the three Euler angles associated with the end-effector's orientation. These will be referred to as the *cartesian* and *eulerian* components of a workspace, respectively, and collectively as the workspace.

The *torsional space* of a $6R$ chain is $[0°, 360°)^6$, and is the range of values that θ can take. Q is a many-to-one function from torsion space to workspace

$$Q : [0°, 360°)^6 \rightarrow \mathbb{R}^3 \times [0°, 360°)^3. \tag{5.5}$$

[33] i.e. 'What is the location and orientation of F_6 with respect to F_0?'



(a) (b)

Figure 5.5: There are two solutions to the IK problem for (a) the $3R$ chain at Q (red circle): s_0^0 with $\theta = \langle 135°, 90°, 135° \rangle$ and s_1^0 with $\theta' = \langle 45°, 270°, 135° \rangle$. Infinite solutions exist for (b) the $2R$ chain at Q if both joints rotate at equal rates in opposite directions. In contrast with (a) where joint rotation axes are perpendicular to the page, the joints of (b) are vertically parallel to the page.

The FK problem can therefore be stated as finding the image of $x \in [0°, 360°)^6$ under Q, while the IK problem involves finding the pre-image of $y \in \mathbb{R}^3 \times [0°, 360°)^3$ under Q. The image of the entirety of $[0°, 360°)^6$ under Q is the *conformation space* associated with Q, or a particular parametrization of a $6R$ chain. There exist fast algorithms to calculate all $6R$ chain IK solutions for any single $y \in \mathbb{R}^3 \times [0°, 360°)^3$ [MC94], a fact we'll use in the next chapter. However, no such algorithms exist to find the entire pre-image of a conformation space, which is what we wish to approximate (see Chapter 1).

Conformation spaces encapsulate how motion in $[0°, 360°)^6$ is reflected, under Q, in $\mathbb{R}^3 \times [0°, 360°)^3$. It is *not* a synonym for workspace, as workspaces are essentially volumes of space in $\mathbb{R}^3 \times [0°, 360°)^3$ and it's possible for two different chains to have the same workspace.

5.3 Kinematic Nucleic Acids

Converting a phosphate backbone into a kinematic chain is straightforward. Let the atoms, bond angles, bond lengths, and torsion - or dihedral - angles of the backbone be, respectively,

$$
\begin{aligned}
Atom &:= \langle P5, O5, C5, C4, C3, O3, P3 \rangle \\
Angle &:= \langle 180°, 180°, P5-O5-C5, O5-C5-C4, C5-C4-C3, \\
&\quad C4-C3-O3, C3-O3-P \rangle \\
Bond &:= \langle 0, P5-O5, O5-C5, C5-C4, C4-C3, C3-O3, O3-P3 \rangle \\
Tors &:= \langle 0, \alpha, \beta, \gamma, \sigma, \epsilon, \zeta \rangle
\end{aligned} \tag{5.6}
$$

where each element of $Angle$ and $Bond$ is the angle between three successive atoms about their interior bonds and distance between two successive bonds, respectively (Figure 5.6).

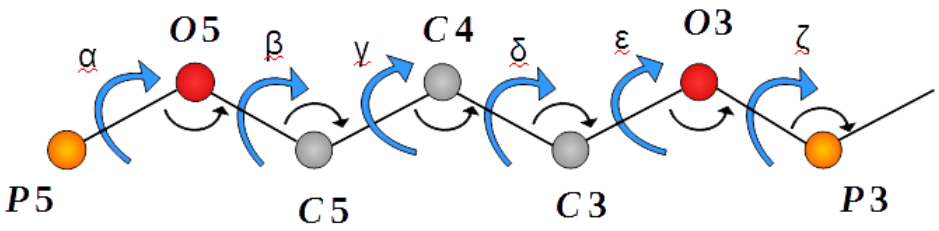


Figure 5.6: Atoms of the phosphate backbone. Atomic bond angles are indicated by thin black arrows and torsion angles are indicated by thick blue arrows. Note that there is no offset between successive atoms. Hydrogen atoms are excluded, but can be incorporated with some modifications.

Accordingly, frame F_i about atom $Atom[i]$ is defined by

$$
\begin{aligned}
a_i &= Bond[i] \\
\alpha_i &= Angle[i] - 180° \\
\theta_i &= Tors[i] \\
b_i &= 0,
\end{aligned} \tag{5.7}
$$

providing a parametrization of Equation (5.1). Index 0 orients the base frame about the $P5$ atom and is technically not needed (see Equation 5.2). It is included for completeness' sake. For our RNA chain, Q defines the relative position and orientation of the two phosphate atoms, $P5$ and $P3$, in the backbone.

5.4 Outline of Converting IK Solutions and Stacked Graphs

The following chapter details the technique we'll use to sample the conformation space for kinematic chains, in general, and single phosphate backbones, in the specific. This sample set will, in Chapter 7, be converted into a stacked graph $\mathbb{G}$, feasible assignments on which are point-wise approximation of the space's structure, as described in Section 2.1. Figure 5.7 illustrates the general process.

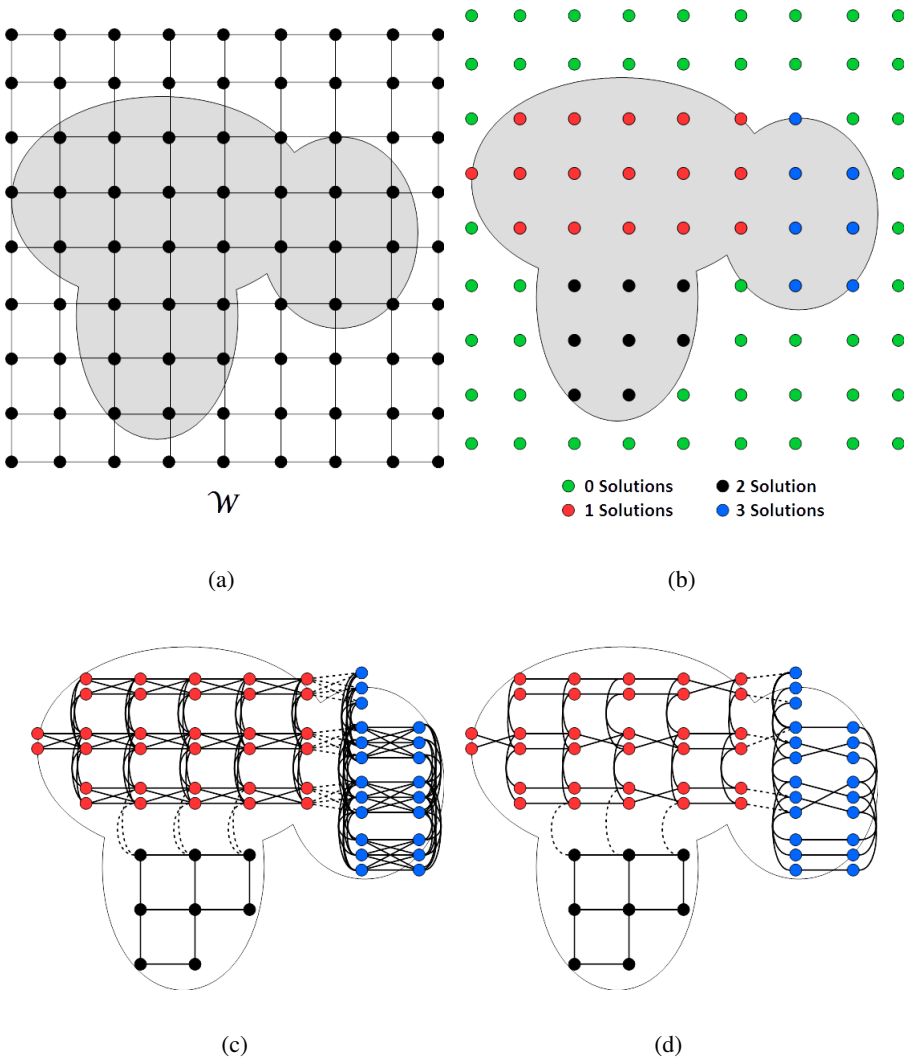


Figure 5.7: (a) A lattice is projected onto a chain workspace $\mathcal{W}$, after which (b) IK solutions are found at each lattice vertex. (c) Vertex stacks are formed from the solutions and edge stacks created between samples adjacent on the lattice with the number of solutions at each location defining regions and boundaries (dotted lines) for a stacked graph $\mathbb{G}$. Finally, (d) an optimal feasible assignment A is found for $\mathbb{G}$ using the LP described in Chapter 4.

Chapter 6

Conformation Space Sampling

In this chapter we describe the process by which chain conformation spaces are sampled, including a limited analysis of the number of samples that will be needed for the case of phosphate backbones. The first section introduces the technique while the second formally defines terms.

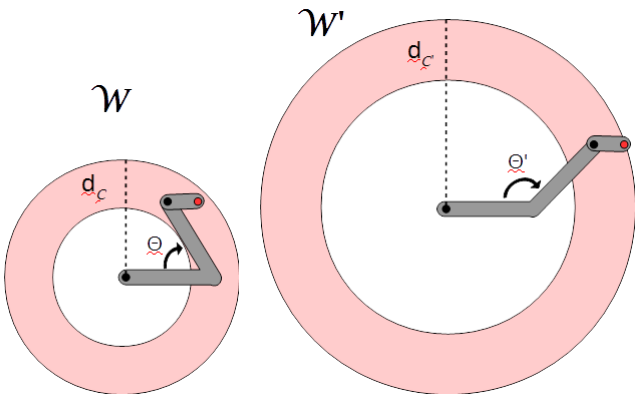
6.1 Sampling Technique

As noted in Section 5.2, every 6R chain workspace is a subspace of $\mathbb{R}^3 \times [0, 360°)^3$. The extent of the $\mathbb{R}^3$ (cartesian) components, or the maximum distance that can be had between the $P5$ and $P3$ atoms, are determined by the bond lengths a_i and angles α_i of the chain and are therefore dependent on specific parametrizations. Large values of a_i and values of α_i close to 180° tend to produce subspaces of $\mathbb{R}^3$ with a greater maximal extent than smaller a_i and more acute α_i (Figure 6.1). This is a basic geometric result. Eulerian component extent is affected by chain parametrization to a lesser degree, given that the super-space $[0, 360°)^3$ is of finite extent.

Ideally, a sampling lattice bounds the workspace as tightly as possible as we wish to sample the workplace exterior as infrequently as we can (i.e. the 0 solutions in Figure 5.7(b)). However, we would also like a lattice that can be used irrespective of chain parametrization, and therefore of workspace shape. To this end, we normalize all workspaces by dividing each a_i lengths by the maximal extent of the workspace. This produces normalized workspaces with maximal extents of 1, fitting nicely into the cubic space $[-1, 1]^3$ when our base frame F_0 (i.e. atom $P5$) is located at the origin $< 0, 0, 0 >$ (Figure 6.2).

Determining maximal workspace extents will not be covered here. For pure rotational chains with no offset (see Section 5.1), such as the phosphate backbone $6R$ chain, maximal extents are derivable by trigonometric arguments. Figure 5.6 illustrates a backbone conformation with maximal extent.

Sampling itself is done via bisection, repeatedly halving the $[-1, 1]$ and $[0, 360°)$ lattice components and taking samples on the resultant grid. We denote the number of bisections for the i^{th} cartesian interval to be n_{c_i}. The number of sample points



(a) (b)

Figure 6.1: Planar $2R$ example of how small changes in parameter values ($\theta \rightarrow \theta'$) can impact maximum workspace extent ($d_C \rightarrow d_{C'}$). Only the cartesian components of the workspaces (pink annuli) are shown.

on the i^{th} interval (including the ± 1 ends of the interval) is $2^{n_{c_i}} + 1$, so that the total number for the entire cartesian component is

$$(2^{n_{c_1}} + 1)(2^{n_{c_2}} + 1)(2^{n_{c_3}} + 1). \tag{6.1}$$

Similarly, the number of eulerian samplings, defined by the number of bisections n_{e_i}, is

$$(2^{n_{e_1}})(2^{n_{e_2}})(2^{n_{e_3}}) \tag{6.2}$$

with the $+1$ dropped due to the the circularity of $[0, 360°)$.

While each n_{c_i} and n_{e_j} are allowed to be different, to simplify matters, we'll assume that each n_{c_i} take on the same value, as well as each n_{e_j}. This allow us to refer to n_c and n_e without the additional subscripts. The total number of samples for a lattice becomes

$$(2^{n_c} + 1)^3 (2^{n_e})^3 \approx 2^{3(n_c+n_e)} = 8^{n_c+n_e}. \tag{6.3}$$

For small values, the number of samples is rather large: at $n_c = n_e = 4$, we have 16.7 million samples, and at $n_c = n_e = 5$, a little over a billion. These figures are for *samples*, not *solutions*, the later of which are bound by a constant

$$k 8^{n_c+n_e} \tag{6.4}$$

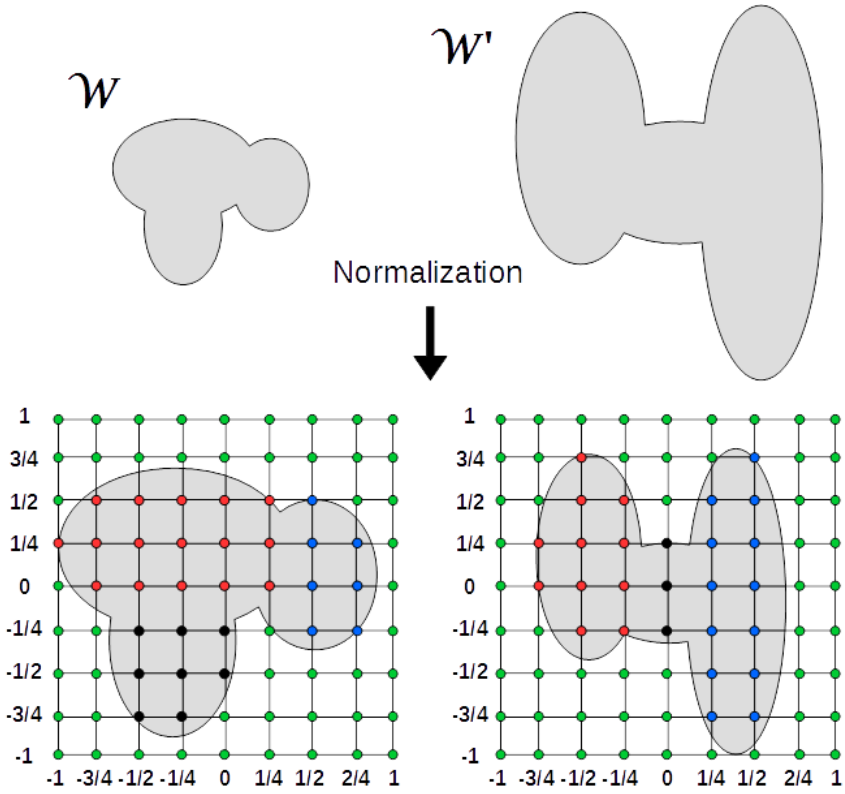


Figure 6.2: Normalizing workspaces A and B allows the same sampling lattice to be used.

with $k <= 16$. Recall from Section 5.2 that $6R$ chains have at most 16 IK solutions in the finite case.

But how accurately *do* four bisections approximate the underlying space? What of five bisections? Can we get away with 'only' a billion samples? Fortunately, there is a practical upper limit on the values of n_c and n_e resulting from the accuracy of current RNA crystallography data.

The Nucleic Acid Database (NDB) is a primary repository for nucleic acid information. The standard structural geometries of nucleic acids used by the NDB [Dat16] - including bond lengths and angles - are those of [GSC+96]. Table 6.1 displays the standard bond lengths, from which we can determine that the maximal extent of any unnormalized kinematic RNA workspace is, at most, 9.141 Å, a gross overestimation given that it supposes all bond angles are 180°.

Minimum bond length *error* is approximately 0.001Å. Any sampling lattice

associated with the workspace of maximal possible extent with an (unnormalized) grid size *less than* 0.001Å would thus provide a sampling resolution below the error bounds. This in turn provides us a practical upper limit on the value of n_c.

Table 6.1: RNA phosphate backbone mean bond lengths ($\bar{x}$) in Angstroms (Å), including their standard deviations (esd), number of samples (N), standard errors (se), and upper bounds of the mean lengths within two standard deviations. Total lengths indicate maximum possible extent of workspace, where all bond angles are assumed 180°. * from [GSC+96].

Bond Lengths

Bond	$\bar{x}$*	esd*	N*	se	$\bar{x} + 2$se
P-O5	1.593	0.010	15	0.003	1.599
O5-C5	1.440	0.016	14	0.004	1.448
C5-C4	1.510	0.013	80	0.001	1.512
C4-C3	1.524	0.011	80	0.001	1.526
C3-O3	1.433	0.019	13	0.005	1.443
P-O3	1.607	0.012	13	0.003	1.613
Total Length	9.107	-	-	-	9.141

The value of n_e is easier to restrict, since it's parameter - and therefore workspace - independent. From Table 6.2, we see that the smallest standard error for phosphate backbone bond angles is roughly 0.2°.

Table 6.2: RNA phosphate backbone mean bond angles ($\bar{x}$) in degrees, including their standard deviations (esd), number of samples (N), and standard errors (se). * from 4.5[GSC+96].

Bond Angles

Angle	$\bar{x}$*	esd*	N*	se
P-O5-C5	120.9	1.6	15	0.4
O5-C5-C4	109.4	0.8	14	0.2
C5-C4-C3	115.5	1.5	80	0.2
C4-C3-O3	110.6	2.6	80	0.3
C3-O3-P	119.7	1.2	13	0.3

The effect of bisections on grid lengths - including the maximal and normalized cartesian lengths - is described in Table 6.3. At $n_c = 14$ and $n_e = 12$ bisections, sampling resolutions is below 0.001Å and 0.2°, respectively.

Table 6.3: Effect of the number of bisections on the mean, maximal, and normalized cartesian grid lengths (Å), as well as the angular eulerian grid lengths (degrees). Bold values indicate cut-off points where lengths fall below standard error of existing experimental data.

# Bisects	Mean	Maximal	Normalized	Angular
0	9.107	9.141	1	360
1	4.554	4.570	0.5	180
2	2.277	2.285	0.25	90
3	1.138	1.143	0.125	45
4	0.569	0.571	0.0625	22.5
5	0.284	0.286	0.03125	11.3
6	0.142	0.143	0.01563	5.6
7	0.071	0.071	0.00781	2.8
8	0.035	0.036	0.00391	1.4
9	0.018	0.018	0.00195	0.7
10	0.009	0.009	0.00098	0.4
11	0.004	0.004	0.00049	0.2
12	0.002	0.002	0.00024	**<0.1**
13	0.001	0.001	0.00012	< 0.1
14	**<0.001**	**<0.001**	**0.00006**	< 0.1

It is important to note that the distances associated with these values of n_c and n_e are *not* indicative of actual crystallographic resolutions, but the errors associated with means. Some of the best existing data has a resolution of 0.48Å with values < 1Å being considered highly accurate [BHA15]. $n_c = 5$ therefore provides a higher resolution than most, if not all, existing data.

Calculating a more realistic limit for n_e is difficult. For our purposes, we'll take $n_e = 7$, equivalent to 2.8°, to be a reasonable upper limit.

This equates to $8^{5+7} \approx 69$ billion sample points. Even in light of the existence of high-speed IK algorithms, sampling the space is somewhat problematic: at a solving rate of (at least) 10 ms per sample [MC94], a total of 7,986 days - or 22 years - of CPU time is needed. Samples, however, are independent, so that the solving process is trivially parallelizable and therefore a candidate for distributed, cloud, or supercomputing.

Fortunately, we may need significantly fewer than 70 billion samples. The semi-rigid nature of backbone chains means that the accuracy of a conformation space approximated by any single chain parametrization may be quite low, depending on the value distributions on the parameters. It is therefore not necessary to obtain a large number of samples. What is important is how conformation spaces change under a distribution of parametrizations, changes which may manifest at low sampling resolutions (i.e. small values of n_c and n_e). Chapter 8 covers techniques for analysing such changes.

6.2 Formal Definitions

To be consistent across RNA chain parametrizations as well as n_c and n_e values, sample points require unambiguous labels. Using lattice coordinates alone is insufficient. For one, there are two lattices: the normalized lattice, with rational cartesian coordinates in $[-1, 1]$, and the lattice corresponding to the actual, unnormalized workspace, with real cartesian coordinates. For another, coordinates with real or rational values are awkward to work with in algorithmic contexts, where iterating over integers is common place.

There is also the issue of comprehension. If normalized coordinates are treated as canonical sample names, what, for instance, is the relationship between samples S_a and S_b below?

$$S_a : \left\langle -\frac{1}{2}, \frac{1}{2}.\frac{3}{4}, 45°, 270°, 180° \right\rangle \tag{6.5}$$

$$S_b : \left\langle -\frac{3}{4}, \frac{1}{4}, 0, 180°, 90°, 225° \right\rangle \tag{6.6}$$

Are they close to each other on the lattice relative to their sampling resolutions,

or quite distant? Do they even exist in the same *set* of resolutions, or is each vector component from different n_c and n_e values[34]? This isn't clear from the vectors alone, and knowing whether samples are adjacent to each other on a lattice is important.

We address these problems by introducing a formal sample naming scheme, followed by a set of functions that allow us to move between normal and non-normal lattices and alternative labelling techniques which will prove useful. First, sample naming.

We'll begin with the one dimensional case, where each sample is either on $[-1, 1]$ or $[0, 360°)$, then generalize to samples on $[-1, 1]^3 \times [0, 360°)^3$.

At $n_c = 0$, there are only two samples, located at -1 and $+1$ (Figure 6.3). We define these as S_0 and S_1, respectively. Increasing to $n_c = 1$ yields a third sample, S_2. At $n_c = 2$, S_3 and S_4 appear. The process continues, labelling new samples at successive midpoints, starting from the -1 end and working towards $+1$, naming them by increasing integer subscripts.

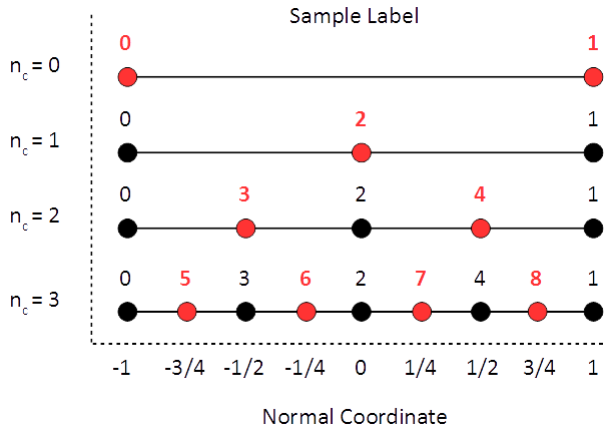


Figure 6.3: Sample labelling on cartesian components up to $n_c = 3$ bisections. Successive lines indicate sample labels introduced at particular n_c values (in red) as well as previously generated samples (in black). The bottom axis indicates the coordinate value of each sample in the normalized sample space.

Definition 6.1 (Bisectors)**.** The i^{th} *cartesian* and j^{th} *eulerian bisectors* are n_{c_i} and n_{e_j}, respectively. The *integers generated by* n_{c_i} *and* n_{e_j} fall in distinct ranges (Table 6.4).

[34]For example, $S_a[1] = -\frac{1}{2}$ and $S_b[1] = -\frac{3}{4}$ are produced by $n_c = 2$ and 3, respectively.

The values in Table 6.4 result from the following theorem.

Theorem 6.2 (Resolution)**.** *Let $k > 1$ and $k' > 1$ be generated by n_c and n_e, respectively. Then*

$$k \in [2^{n_c-1} + 1, 2^{n_c}] \tag{6.7}$$
$$k' \in [2^{n_e-1}, 2^{n_e} - 1] \tag{6.8}$$

and

$$n_c = \lceil \log_2 k \rceil \tag{6.9}$$
$$n_e = \lfloor \log_2 k' + 1 \rfloor. \tag{6.10}$$

In this context, bisectors n_c and n_e are the resolutions of k and k'*, respectively.*

Proof. The first part can be proved via induction on Figure 6.3 for n_c and a similar figure for n_e. For the second part, from Equation 6.9 we have

$$\begin{aligned} 2^{n_c-1} + 1 \leq \quad & k \quad & \leq 2^{n_c} \\ 2^{n_c-1} \leq \quad & k \quad & \leq 2^{n_c} \\ n_c - 1 \leq \quad & \log_2 k \quad & \leq n_c \\ \log_2 k \in \quad & [n_c - 1, n_c]. \end{aligned} \tag{6.11}$$

However, the upper-bound for the range generated by n_c is always a power of 2 so that k can be such that $\log_2 k = n_c$. Thus

$$n_c = \lceil \log_2 k \rceil. \tag{6.12}$$

Similarly, from Equation 6.8 we get $\log_2 k' \in [n_e - 1, n_e]$, though this time k' can be such that $\log_2 k = n_e - 1$ so that

$$n_e = \lfloor \log_2 k + 1 \rfloor. \tag{6.13}$$

□

Values generated by multiple n_{c_i} and n_{e_j} are used to name conformation space samples.

Definition 6.3 (Sample Name and Resolution)**.** Let S_k be a sample in $\mathbb{R}^3 \times [0, 360°)^3$. The *name* of S_k is the vector

$$S_k := \langle k_0, k_1, k_2, k_3, k_4, k_5 \rangle, \tag{6.14}$$

with $k_i \geq 0 \; \forall \; i$. The *resolution* of S_k is the vector returned by Res(·)

$$\text{Res}(S_k) := \left\langle n_{c_0}, n_{c_1}, n_{c_2}, n_{e_0}, n_{e_1}, n_{e_2} \right\rangle \tag{6.15}$$

where

$$\begin{aligned} n_{c_0} &= \lceil \log_2 k_0 \rceil, \; n_{e_0} &&= \lfloor \log_2 k_3 + 1 \rfloor \\ n_{c_1} &= \lceil \log_2 k_1 \rceil, \; n_{e_1} &&= \lfloor \log_2 k_4 + 1 \rfloor \\ n_{c_2} &= \lceil \log_2 k_2 \rceil, \; n_{e_2} &&= \lfloor \log_2 k_5 + 1 \rfloor. \end{aligned} \tag{6.16}$$

The terms 'sample' and 'name' are synonymous.

Table 6.4: The ranges of k generated by n_c/n_e.

n_c/n_e	n_c Range	n_e Range
0	[0,1]	[0]
1	[2]	[1]
2	[3,4]	[2,3]
3	[5,8]	[4,7]
4	[9,16]	[8,15]
5	[17,32]	[16,31]
6	[33,64]	[32,63]
7	[65,128]	[64,127]
8	[129,256]	[128,255]
9	[257,512]	[256,511]
10	[513,1024]	[512,1023]
11	[1025,2048]	[1024,2047]
12	[2049,4096]	[2048,4098]
13	[4097,8192]	[4096,8191]
14	[8193,16384]	[8192,16383]

The notation 'S_k' references a name, not the subscript k. It's possible to derive a unique value for k from a name (Appendix A), however it isn't as useful to work with. As such, subscripts will be used solely to differentiate samples, such as S_3 and S_9.

Definition 6.4 (Normal Coordinates)**.** The *normal coordinates* of S_k is the vector returned by Norm(·)

$$\text{Norm}(S_k) := \langle w_0, w_1, w_2, w_3, w_4, w_5 \rangle \tag{6.17}$$

where $w_0, w_1, w_2 \in [-1.1]$ and $w_3, w_4, w_5 \in [0°, 360°)$. Norm$(S_k)$ are the coordinates of S_k in $[-1, 1]^3 \times [0, 360°)^3$.

Construction of w_i is described in Appendix B.

Definition 6.5 (Absolute Coordinates)**.** Let d be the maximum linear distance between base frame F_0 and frame F_6 for a given parametrization of a $6R$ chain[35]. The *absolute coordinates* of S_k is the vector returned by Abs$(\cdot)$

$$\begin{aligned} \text{Abs}(S_k) &:= \langle a_0, a_1, a_2, a_3, a_4, a_5 \rangle && (6.18) \\ &= \langle d \cdot w_0, d \cdot w_1, d \cdot w_2, w_3, w_4, w_5 \rangle && (6.19) \end{aligned}$$

Abs(S_k) are the coordinates of S_k in the original chain workspace $\mathbb{R}^3 \times [0°, 360°)^3$.

Definition 6.6 (Sample Adjacency)**.** Let $r = \langle n_0, n_1, n_2, n_3, n_4, n_5 \rangle$ be a sample resolution and let S_u and S_v be two samples such that

$$\text{Res}(S_u)[i] \leq r[i] \quad (6.20)$$
$$\text{Res}(S_v)[i] \leq r[i], \quad (6.21)$$

with $0 \leq i \leq 5$. In other words, S_p and S_q *exist* at resolution r. Then, S_p and S_q are *adjacent at resolution r* if there exist no S_w, $S_w \neq S_p$ or S_q, that satisfies the following two conditions

$$1. \quad \text{Res}(S_w)[i] \leq r[i] \quad (6.22)$$

and either of

$$2a. \quad \text{Norm}(S_p)[i] \leq \text{Norm}(S_w)[i] \leq \text{Norm}(S_q)[i] \quad (6.23)$$
$$2b. \quad \text{Norm}(S_q)[i] \leq \text{Norm}(S_w)[i] \leq \text{Norm}(S_p)[i] \quad (6.24)$$

for all $i \in 0, \ldots, 5$.

The conditions of Definition 6.6 define vertex adjacency on a lattice graph with varying scales across axes. Consider, for example, the planar graph in Figure 6.4. The nodes $(0, 0)$, $(\frac{1}{2}, 0)$, and $(0, 1)$ are all adjacent, but $S_p = (0, 0)$ and $S_q = (\frac{1}{2}, 1)$ are not because $S_w = (\frac{1}{2}, 1)$ satisfies Condition 2a.

For a full example, consider the two samples S_1 and S_2

$$\begin{aligned} S_1 &= \langle 1, 5, 4, 7, 3, 9 \rangle && (6.25) \\ S_2 &= \langle 6, 10, 1, 11, 2, 8 \rangle && (6.26) \end{aligned}$$

[35] In our case, d is the maximum linear distance that atom $P5$ can be from $P3$ for a given set of bond length and angle values.

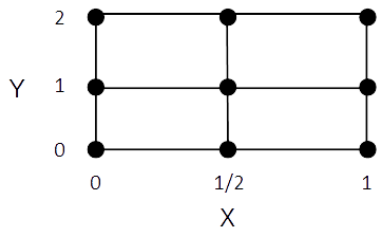


Figure 6.4: A planar lattice graph whose x-axis components are half the scale of the y-axis components.

which have normal coordinates

$$\text{Norm}(S_1) = \left\langle -1, \frac{3}{4}, -\frac{1}{2}, 315°, 270°, 67.5° \right\rangle \tag{6.27}$$

$$\text{Norm}(S_2) = \left\langle \frac{1}{4}, \frac{5}{8}, -1, 157.5°, 90°, 22.5° \right\rangle \tag{6.28}$$

and resolutions

$$\text{Res}(S_1) = \left\langle 0, 3, 2, 3, 2, 4 \right\rangle \tag{6.29}$$

$$\text{Res}(S_2) = \left\langle 3, 4, 0, 4, 2, 4 \right\rangle. \tag{6.30}$$

In addition, let the sample resolution be $r = \left\langle 3, 4, 2, 4, 3, 4 \right\rangle$. At r, S_1 and S_2 are adjacent. Figure 6.5 illustrates why.

Adjacency is defined at specific resolutions. If $r[1]$ and $r[2]$ were both any larger, then S_1 and S_2 would no long be adjacent.

That brings us to solutions, which also require unambiguous names.

Definition 6.7 (Sample Solutions)**.** Every sample S_k consists of a set of solutions s_i^k of the form

$$s_i^k := \left\langle \theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5 \right\rangle \tag{6.31}$$

with $\theta_j \in [0°, 360°)\ \forall j = 0, \ldots, 5$ being a torsion angle. Each s_i^k is a unique IK solution to a kinematic chain sampled at coordinates $\text{Abs}(S_k)$.

Unlike samples, solutions don't require explicit, vectorized names.

There is no inherent ordering to solutions. However, numbering must be consistent. While any ordering will do, one system will prove particularly useful when it comes time to optimize. The ordering is based on the absolute value of solution torsion angles.

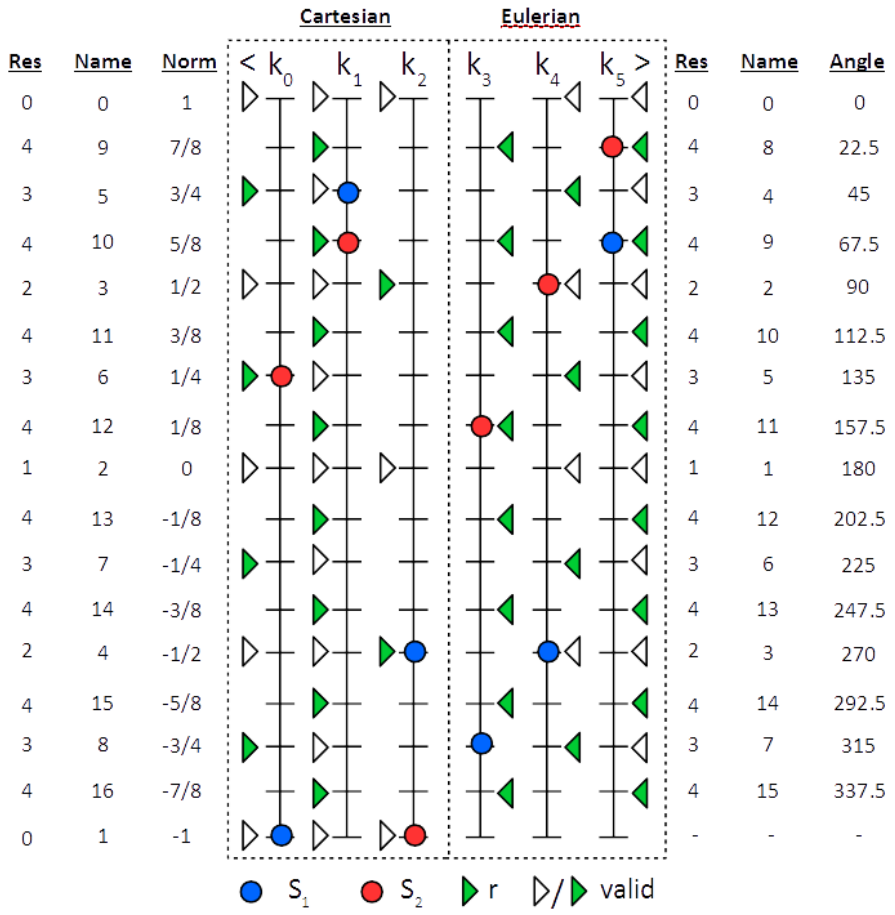


Figure 6.5: A vertical diagram for determining sample adjacency. The left hand side describes the (Res)olution, Name, and (Norm)al labels for the cartesian components of a sample, while the right hand side describes the eulerian, replacing normal distances with angular ones. The central component shows the values generated by $r = \langle 3,4,2,4,3,4 \rangle$ (green triangles), S_1 and S_2 (blue and red circles, respectively), and valid locations for any sample at resolution r (white and green triangles). To be adjacent, there must be at least one k_i where there are no valid locations between S_1 and S_2 (i.e. no triangles between red and blue circles). Both k_1 and k_2 satisfy this condition. Thus S_1 and S_2 are adjacent.

Definition 6.8 (Absolute Torsion Angle)**.** The absolute value of a torsion angle θ_i is

$$|\theta_i| := \begin{cases} \theta_i, & \text{if } \theta_i \leq 180^{\text{o}} \\ 360^{\text{o}} - \theta_i, & \text{otherwise.} \end{cases} \tag{6.32}$$

Solutions are ordered by increasing values of $|\theta_0|$, then $|\theta_1|$, etc. In the event of ties, the smallest *non*-absolute value is used. For example, let s_0^1, s_1^1, and s_2^1 be the three solutions of some $\boldsymbol{S}_1$

$$s_0^1 = \left\langle 33°, 167°, 347°, 12°, 298°, 101° \right\rangle \tag{6.33}$$

$$s_1^1 = \left\langle 92°, 2°, 311°, 321°, 134°, 296° \right\rangle \tag{6.34}$$

$$s_2^1 = \left\langle 33°, 193°, 55°, 277°, 176°, 4° \right\rangle. \tag{6.35}$$

As

$$|s_1^1[0]| > |s_0^1[0]| = |s_2^1[0]|$$

s_1^1 is last in the ordering. Since $|s_0^1[0]| = |s_2^1[0]|$, we need to look at the next index to determine ordering for s_0^1 and s_2^1. While

$$|s_0^1[1]| = |s_2^1[1]|,$$

the non-absolute angles yield

$$s_0^1[1] < s_2^1[1].$$

This implies that the ordering should be

$$s_0^1 < s_2^1 < s_1^1.$$

Definition 6.9 (Absolute Torsion Angle Ordering)**.** Let $\boldsymbol{S}_k$ be a sample with solutions s_i^k. Then, the *absolute (torsion angle) ordering of* $\boldsymbol{S}_k$ is the ordering of all s_i^k by the smallest absolute value $|s_i^k[j]|$, for increasing indices j beginning at $j = 0$. Ties are broken by the smallest non-absolute value, $s_i^k[j]$.

For the remainder of this work, an absolute ordering of solutions will be assumed in all cases.

It should be noted that absolute coordinates are defined as a scaling of normal coordinates, not the other way around. This is done to emphasis the independence of our modelling technique from any specific $6\boldsymbol{R}$ chain parametrization. Parametrizations are treated as special cases of a generalized normal form.

The notion of *distance* between two arbitrary solutions s_u^p and s_v^q is an important one.

Definition 6.10 (Angular Distance Between Solutions)**.** The *angular distance* between two angles $\theta, \phi \in [0°, 360°)$ is

$$\mathrm{d}(\theta, \phi) := |\theta - \phi|. \tag{6.36}$$

The angular distance between two vectors of angles of equal length $\theta = \langle \theta_0, \theta_1, \ldots, \theta_m \rangle$ and $\phi = \langle \phi_0, \phi_1, \ldots, \phi_m \rangle$ is

$$\mathrm{d}(\theta, \phi) := \sum_{i=0}^{m} \mathrm{d}(\theta_i, \phi_i). \tag{6.37}$$

The angular distance between two solutions s_u^p and s_v^q can therefore be written as

$$\mathrm{d}(s_p^u, s_q^v). \tag{6.38}$$

Angular distances arise from modelling changes in solution torsion angles resulting from small changes in the location of Q in its workspace. These small change in Q correspond to moving between adjacent samples on the normalized sampling lattice. We can thus drop all reference to Q and deal strictly with (the normal coordinates of) sample points. We'll demonstrate how with an example using the $3R$ planar chain of Figure 5.5(a).

Consider the two solutions of S_0 in Figure 6.6,

$$s_0^0 = \left\langle 135°, 90°, 225° \right\rangle \tag{6.39}$$
$$s_1^0 = \left\langle 45°, 270°, 135° \right\rangle. \tag{6.40}$$

At sufficiently high resolutions, an adjacent solution S_1 will have the same number of solutions as it will be in the same region[36], whose torsion angles are similar to those of S_0, say

$$s_0^1 = \left\langle 120°, 120°, 210° \right\rangle \tag{6.41}$$
$$s_1^1 = \left\langle 60°, 240°, 150° \right\rangle. \tag{6.42}$$

The angular distances between the first solutions and the second solutions of S_0 and S_1

$$\mathrm{d}(s_0^0, s_0^1) = 60°$$
$$\mathrm{d}(s_1^0, s_1^1) = 60°$$

[36] See Theorem 7.2.

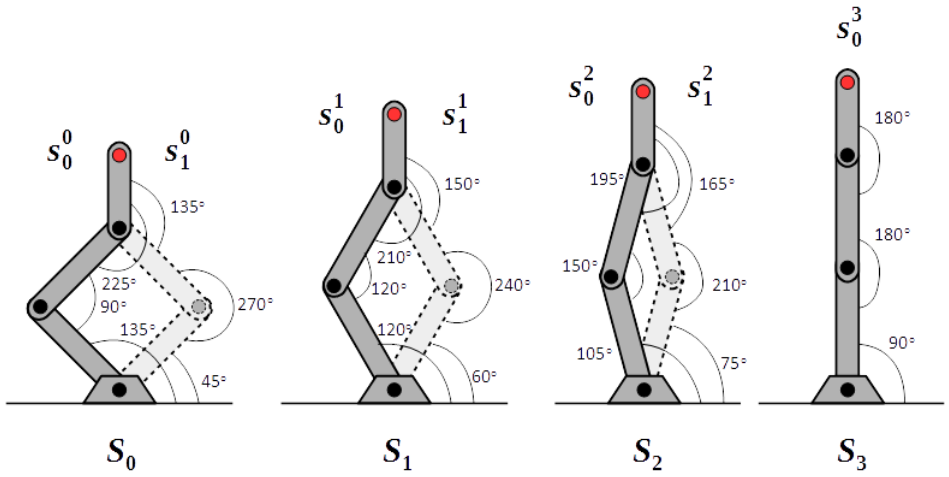


Figure 6.6: An example of conformational paths for a planar $R3$ chain. At each sample point (red circle) S_i, there are two distinct solutions, s_0^i and s_1^i. If successive S_i are assumed to be adjacent at some resolution, then the sequences $s_0^0 - s_0^1 - s_0^2$ and $s_1^0 - s_1^1 - s_1^2$ represent two disjoint conformational pathways from S_1 to S_3. With the incorporation of S_4, however, the paths are no longer disjoint: $s_0^0 - s_0^1 - s_0^2 - s_0^3 - s_1^2 - s_1^1 - s_1^0$ is a conformational path.

are much less than between s_0^0 and s_1^1, and s_1^0 and s_0^1

$$\begin{aligned} \mathrm{d}(s_0^0, s_1^1) &= 300° \\ \mathrm{d}(s_1^0, s_0^1) &= 300°. \end{aligned}$$

Assume a new sample S_2 adjacent to S_1 (Figure 6.6) with solutions

$$s_0^2 = \langle 105°, 150°, 195° \rangle \tag{6.43}$$
$$s_1^2 = \langle 75°, 210°, 165° \rangle. \tag{6.44}$$

Then similar results exist between the solutions of S_1 and S_2

$$\begin{aligned} \mathrm{d}(s_0^1, s_0^2) &= 60° \\ \mathrm{d}(s_1^1, s_1^2) &= 60° \\ \mathrm{d}(s_0^1, s_1^2) &= 180° \\ \mathrm{d}(s_1^1, s_0^2) &= 180°. \end{aligned}$$

There is a kind of natural 'path' between s_0^0, s_0^1, and s_0^2, as well as between s_1^0, s_1^1, and s_1^2. These are the *conformational pathways* of the sample sequence $S_0 - S_1 - S_2$. If we were to continuously 'pull' the end-effector from its position at S_0 to S_2 in Figure 6.6, it's clear that s_0^0 would transform into s_0^1, then into s_0^2. Similarly with s_1^0, s_1^1, and s_1^2. The pathways are discreet approximations of an otherwise smooth transformation.

A problem occurs when the number of solutions at a sample changes. In Figure 6.6, S_3 only has one solution

$$s_0^3 = \left\langle 90°, 180°, 180° \right\rangle. \tag{6.45}$$

Both the solutions at S_2 are equidistant from s_0^3

$$\begin{aligned} \mathrm{d}(s_0^2, s_0^3) &= 60 \\ \mathrm{d}(s_1^2, s_1^3) &= 60, \end{aligned}$$

violating what was previous bijection between solutions in adjacent samples. This 'collapsing' of solutions (or 'expansion', if going from fewer to more) is termed a *singularity* [37].

Definition 6.11 (Singularity)**.** A *singularity* is a pair of adjacent samples S_u and S_v such that $|S_u| \neq |S_v|$. An edge in a sampling lattice is a *singular edge* if it is adjacent on a singularity.

It should be clear that the angular distance between any two solutions of the same sample, s_i^k and s_j^k, is always non-zero. If $\mathrm{d}(s_i^k, s_j^k)$ were zero, they would cease to have distinct torsion angles, implying that $s_i^k = s_j^k$. This yields an interesting property: no conformational pathway exists between s_i^k and s_j^k in the absence of singularities. In order to for a conformational pathway to exist between s_i^k and s_j^k, passage through a singularity must occur. This is apparent for s_0^0 and s_1^0 in Figure 6.6 and is proved for general kinematic chain conformation spaces in Theorem 7.2.

Definition 6.12 (Conformational Pathways)**.** Let S be a path on a sampling lattice

$$S := S_0 - S_1 - \cdots - S_n \tag{6.46}$$

[37] Singularities also occur in the proper study of kinematic chains, representing losses of linear independence between certain axes in lower pairs, often producing sample points with infinite IK solutions. Our usage of the term is strictly to indicate changes in the number of solutions, or regional boundaries. We won't be working with 'proper' singularities, so there should be no difficulty in using the terminology.

and let $m = \max\{|S_0|, |S_1|, \ldots, |S_n|\}$. In addition, let P be an ordered set of m vectors p_i of samples solutions

$$p_i := \left\langle s^0_{a_0}, s^1_{a_1}, \ldots, s^n_{a_n} \right\rangle \tag{6.47}$$

on S, denoted by $P(S)$. Then each p_i is a *conformational pathway*, or *conpaths*, and $P(S)$ is a *bundle of conpaths*. Let the *weight* of any conpath p_k be

$$\mathrm{w}(p_k) = \sum_{i=0}^{n-1} d\left(s^i_{a_i}, s^{i+1}_{a_{i+1}}\right). \tag{6.48}$$

$P(S)$ is a bundle of *potential* conpaths if every solution s^p_u of every sample S_p is an element of at least one conpath p_k[38].

The *weight* of a bundle $P(S)$ is

$$\mathrm{w}(P(S)) := \sum_{p_k \in P} \mathrm{w}(p_k). \tag{6.49}$$

$P(S)$ is a bundle of *likely* conpaths if $P(S)$ has the smallest weight of all possible bundle of potential conpaths on S.

The difference between potential and likely conpaths is one of weighting. In the next chapter, we'll see that bundles of conpaths with smaller weightings will, at sufficiently high sampling resolutions, better reflect path connectedness in the actual conformation space being approximated.

Conformational pathways, can be extended into include entire sampling lattices.

Definition 6.13 (Conformational Pathways of a Sampling Lattice)**.** Let $\mathcal{G}$ be a sampling lattice and $\mathcal{P}$ be a set of bundles of potential conpaths on $\mathcal{G}$ consisting of exactly one bundle P_{pq} on every pair of adjacent samples S_p and S_q in $\mathcal{G}$. $\mathcal{P}$ is called a *conpath bundle set*.

Two solutions s^p_u and s^q_v are considered path connected in $\mathcal{P}(\mathcal{G})$ if there exists a sequence of solutions

$$s^p_u - s^{a_1}_{b_1} - s^{a_2}_{b_2} - \cdots - s^{a_m}_{b_n} - s^q_v \tag{6.50}$$

where every successive pair of solutions s^r_i and s^t_j form a conpath in bundle P_{rt}.

Let p^{pq}_{uv} be a path between s^p_u and s^q_v for a given $\mathcal{P}(\mathcal{G})$. Then $\mathcal{P}(\mathcal{G})$ is a *feasible* if the following conditions hold:

$$p = q \implies \exists\, s^r_i, s^t_j \in p^{pq}_{uv} \ni |S_r| \neq |S_t|. \tag{6.51}$$

[38] i.e. the conpaths of $P(S)$ cover the solution samples of S

In other words, there only exists a path between two solutions in a single sample if the path passes through a singularity. If $\mathcal{P}(\mathcal{G})$ is not feasible, it's *infeasible*.

Let $\mathbb{P}(\mathcal{G})$ be the set of all feasible bundle sets $\mathcal{P}_k$ for sampling lattice $\mathcal{G}$ and let the weight of any $\mathcal{P}_k$ be

$$\mathrm{w}(\mathcal{P}_k) := \sum_{P^k_{pq} \in \mathcal{P}_k} \mathrm{w}(P^k_{pq}). \tag{6.52}$$

The bundle set in $\mathbb{P}$ which has minimal weight is the *optimal conformational space* for $\mathcal{G}$.

The relationship between sampling lattices and conformational pathways with stacked graphs and assignments should be apparent. Every sampling lattice $\mathcal{G}$ can be represented by a stacked graph $\mathbb{G}$, every bundle set $\mathcal{P}$ on $\mathcal{G}$ corresponds to a unique assignment A on $\mathbb{G}$, and feasibility of $\mathcal{P}(\mathcal{G})$ is the same as feasibility on $A(\mathbb{G})$. As the size of edge stacks and the disjoint paths of their assignments define regions and layers in $\mathbb{G}$ and $A(\mathbb{G})$, respectively, so too do bundle sizes and conpaths define *conformational regions* and *conformational layers* in $\mathcal{G}$ and $\mathcal{P}(\mathcal{G})$, respectively.

Definition 6.14 (Conformational Regions and Layers)**.** Let $\mathcal{G}$ be a sampling lattice for a workspace $\mathcal{W}$ and $\mathcal{P}$ be a feasible bundle on $\mathcal{G}$. In addition, let $\mathcal{G}'$ be $\mathcal{G}$ less all singular edges. Then

$$\mathcal{R}_{\mathcal{G}} := \{ R_i | i = 0, 1, 2, \dots \} \tag{6.53}$$

is the set of path connected components, or *conformational regions*, R_i of $\mathcal{G}'$ while

$$\mathcal{L}_i := \{ L^i_j | j = 0, 1, 2, \dots \} \tag{6.54}$$

is the set of path connected components, or *conformational layers*, L^i_j of $\mathcal{P}(\mathcal{R}_i)$.

Chapter 7

Stacked Graphs of Conformation Spaces

In this chapter, we formally re-write sampling lattices in the language of stacked graphs. Prior to doing this, we must demonstrate that sampling lattices actually approximate conformation spaces as well as prove the path-disjointedness of any two sample solutions s_i^k and s_j^k in the absence of singularities. The process of doing so is reminiscent of Section 2.1, where we introduced the process of approximating an unknown function f using 'special pairs' of approaching images (i.e. sample solutions). In what follows below, approaches are defined in terms of shrinking open neighbourhoods and limits.

Kinematic chains are manifolds [McC90]. As sampling resolution increases, the distance between solutions for adjacent samples points on the sampling lattice decrease, with the distance between certain pairs of approaching 0° in the limit. These pairs are what defined conformational pathways in the previous chapter. We now prove that conpaths, as described in Definition 6.13, exist in connected manifolds. The proof assumes some knowledge of topology.

Definition 7.1 (Manifold)**.** [Mun00] An *m-manifold* is a Hausdorff space Y with a countable basis such that each point y of Y has a neighbourhood that is homeomorphic with an open subset of $\mathbb{R}^m$.

Theorem 7.2 (Layers and Regions of Manifolds)**.** *Let f be a continuous function on a connected metric space X*

$$f : X \to Y \tag{7.1}$$

such that Y is an m-manifold and f^{-1} is multi-valued on at least some $y_i \in Y$. Denote the set of pre-images $f^{-1}(y_i)$ as S_i

$$S_i := \{x_j^i | x_j \in f^{-1}(y_i)\}. \tag{7.2}$$

In addition, let K be a connected set in Y where $|S_i| = |S_j|$ for all $k_i, k_j \in K$, with

$$X_K := f^{-1}(K) = \bigcup_{k_i \in K} S_i. \tag{7.3}$$

Then for each $k_i \in K$, all $x^i_j \in S_i$ are mutually path disconnected in X_K. These components are the sub-layers $\underline{\mathrm{h}}^K_i$ *of X_K.*

The boundary of Y under f *is the set of* boundary points $\boldsymbol{B}$ *in Y, where $b \in Y$ is a boundary point if every neighbourhood of b contains at least two points y_r and y_t such that $|S_r| \neq |S_t|$. A* layer $\mathcal{H}_i$ of X under f *is a connected component in $f^{-1}(Y - \boldsymbol{B})$, with the set of all layers denoted as $\mathbb{H}_X$.*

The regions of Y under f *are the connected components of $Y - \boldsymbol{B}$.*

Proof. The proof for the path disconnectedness of all $x^i_j \in X_k$ for all j, for a given i, is by contradiction.

For any $x^p_a, x^p_b \in S_p$, y_p not a boundary point, the distance between x^p_a and x^p_b is

$$d(x^p_a, x^p_b) > 0 \tag{7.4}$$

when $a \neq b$. This follows immediately from the fact that x^p_a and x^p_b are distinct values in X. Next, let $\boldsymbol{B}_p$ be an open m-ball centred on y_p with radius e and which doesn't include a boundary point. Since Y is Hausdorff, such a neighbourhood always exists. For any $y_q \in \boldsymbol{B}_p$, the following two conditions hold:

1. for each $x^p_u \in S_p$ there is exactly one $x^q_v \in S_q$ such that

$$d(x^p_u, x^q_v) \to 0 \text{ as } e \to 0, \tag{7.5}$$

2. for each $x^q_v \in S_q$, there is exactly one $x^p_u \in S_p$ such that

$$d(x^p_u, x^q_v) \to 0 \text{ as } e \to 0. \tag{7.6}$$

Assume that these are not true. Then there are four cases, two for each condition. We treat the first condition only as the proofs for the second are similar.

In the first case, there exists an x^p_u such that there is no x^q_v where

$$d(x^p_u, x^q_v) \to 0 \text{ as } e \to 0. \tag{7.7}$$

In the limit of $e \to 0$, $\boldsymbol{B}_p \to y_p$[39]. Thus, $y_q \to y_p$ as $e \to 0$. However, this implies that $x^q_v \in S_p$ in the limit, meaning that

$$|S_p| \to |S_p| + 1 \text{ as } e \to 0. \tag{7.8}$$

This is impossible. Therefore there must be at least one $x^q_v \in S_p$ for every $x^p_u \in S_q$ such that

$$d(x^p_u, x^q_v) \to 0 \text{ as } e \to 0. \tag{7.9}$$

[39] i.e. the open ball contracts to point y_p at its centre

In the second case, there exist at least two elements in S_q such as Equation 7.5 holds. Let these be x_a^q and x_b^q. However, the existence of x_a^p and x_b^p imply that B_p contains a branch point, which, by definition, it doesn't. Specifically, in the limit of $e \to 0$, $|S_q| \to k$ with $k < |S_q|$: the size of S_q shrinks as $S_q \to S_p$.

This is a contradiction. Therefore there exists at most one $x_v^q \in S_q$ such that Equation 7.5 holds.

Therefore, for any S_k, every $x_i^k \in S_k$ is path disconnect in S. In addition, this implies that if x_i^k and x_j^k are path connected in $f^{-1}(Y)$, the path must contain a boundary point. □

Corollary 7.3. *Let X, Y, f, and B be as in Theorem 7.2. Then, for any $x_u^p, x_v^q \in f^{-1}(Y - B)$, x_u^p and x_v^q in different layers, every path connecting x_u^p and x_v^q contains at least one boundary point b.*

Proof. Since x_u^p and x_v^q belong to difference layers, they are path disconnected in $f^{-1}(Y - B)$. Therefore, if their path connected in $f^{-1}(Y)$, it must be through some element in B. □

Given that Equation 5.2 is continuous on a connected metric space, Theorem 7.2 applies to kinematic chains with the boundary points equivalent to singularities (Definition 6.11). The key difference is that in sampling lattices we treat infinite IK solutions as boundaries as well, the reason for which being that stacked graphs have not been generalized to infinite stack sizes.

Boundary points may be similar to branch points in complex functions. As I have not studied complex analysis - and to avoid digression into yet another mathematical field - the notion of boundary points are used instead.

It should be clear from Corollary 7.3 and the proof of Theorem 7.2 why it's the *minimally weighted* potential conpath bundles are what we're interested in: so long as our sampling lattice $\mathcal{G}$ gives good coverage of the workspace, and so long as a sufficiently high number of samples are used, the connected components in $\mathcal{P}(\mathcal{G})$ approximate the layers of Q^{-1}.

Re-writing a sampling lattice as a stacked graph is straightforward.

Definition 7.4 (Stacked Graphs of Sampling Lattices)**.** Let $\mathcal{G}$ be a sampling lattice consisting of samples S_i, each composed of a set of solutions s_j^i. Then the *stacked graph formed on* $\mathcal{G}$ is $\mathbb{G} = (\mathcal{V}, \mathcal{E})$, where $V_i \in \mathcal{V}$ if $S_i \in \mathcal{G}$, $E_{pq} \in \mathcal{E}$ if S_p and S_q are adjacent in $\mathcal{G}$, and $e_{uv}^{pq} \in E_{uv}$ if $s_u^p \in S_p$ and $s_v^q \in S_q$.

The weight of any e_{uv}^{pq} is

$$w_{uv}^{pq} := d\left(s_u^p, s_v^q\right). \tag{7.10}$$

An edge stack E_{pq} is singular if and only if $|S_p| \neq |S_q|$ and asingular otherwise.

Theorem 7.5. *Let $\mathcal{G}$ be a sampling lattice graph and $\mathbb{G}$ be the stacked graph formed on it. Then, for any path S in $\mathcal{G}$, every bundle P on S is isomorphic to a unique assignment A on $\mathcal{P}$ in $\mathbb{G}$, where $\mathcal{P}$ is the path stack associated with S. Thus, $\mathcal{P}(\mathcal{G})$ is feasible or optimal if and only if $A(\mathbb{G})$ is feasible or optimal, respectively.*

Proof. That every bundle is isomorphic to an assignment is obvious. The rest follows directly from the isomorphism. □

The structure of $\mathbb{G}$, the vertex stacks and their adjacency, is defined by the technique used to sample space X, with assignments A on $\mathbb{G}$ representing possible pre-images and connected components of Y and $Y - B$ under f. The weightings of each edge in $\mathbb{G}_{\mathcal{G}}$, the distances between point-wise pre-images $f^{-1}(y_i)$, are what guide the choice in which assignment best represents $f^{-1}(Y)$ or $f^{-1}(Y - B)$.

It should be emphasised that Theorem 7.2 and Corollary 7.3 apply to *any* continuous functions on a connected metric space that produce manifolds, not just those dealing with kinematic chains. Stacked graphs can therefore be used as a tool for study certain classes of manifolds.

7.1 Validating the ULA Constraint and Disconnecting Assignment Components

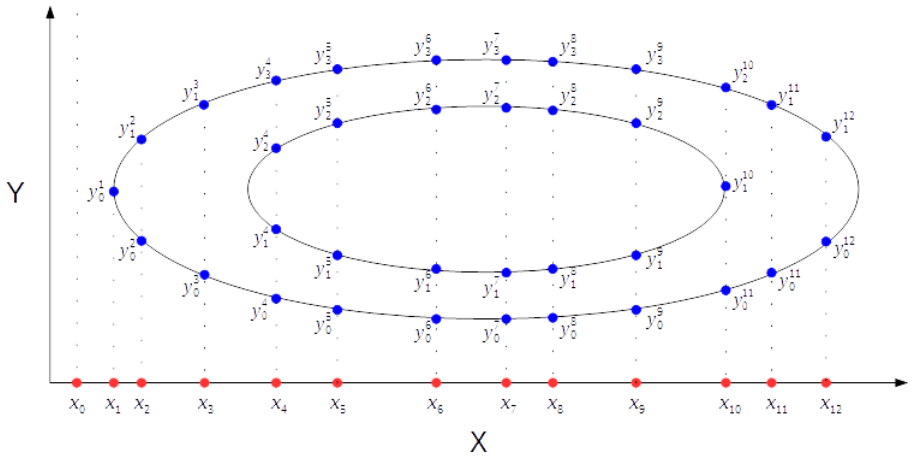
In Section 2.5 we introduced the non-basic Uniform Layer Adjacency (ULA) constraint for singular edge stacks. The rational given for it was so that the proportional and binary forms of the derived assignment $A(\mathbb{R}_{\mathbb{G}})$ would be the same (Lemma 2.43), which would assist in calculating metrics for how well optimal assignments approximate conformation manifolds (Chapter 8). Enforcing ULA using linear programming is, however, not straightforward (Section 4.3), and increasing the difficulty of one set of calculations to easy another set does not imply an overall reduction in complexity. There should be a inherent reason for using ULA. We provide such a rational in this section by demonstrating the conformation spaces exhibit the ULA property, or some variation of it.

Returning to the annulus boundary example of Section 2.1, with the primary image reproduced in Figure 7.1 below, we'll recall that singularities were introduced as size discrepancies between successive image sets $f(x_i)$ and $f(x_{i+1})$. In terms of stacked graphs, this correspond to asingular edge stacks. One such singularity is between the images of x_3 and x_4. What is visually quite clear is that *singularities to not uniformly impact the images they include*. The existence of an edge between y_1^3 and y_2^4 and another between y_0^3 and y_1^4 (Figure 7.1(b)) is questionable, however the edge between y_1^3 and y_3^4, and that between y_0^3 and y_0^4, are certainly not. Regardless of

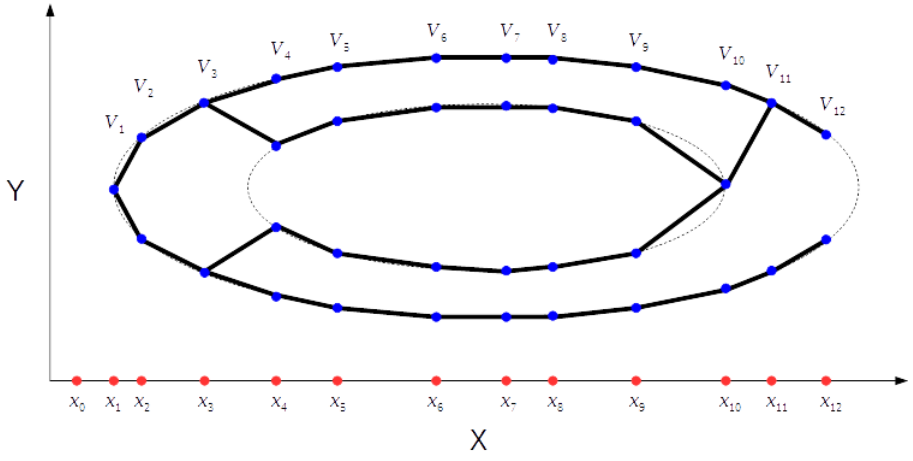
how points on the outer ring related to those of the inner ring, *intra*-ring adjacency is fixed.

In terms stacked graphs, for any asingular E_{pq}, $|V_p| > |V_q|$ if their exist a pair v_i^p and v_j^q such that the distance between them approaches zero in the sampling limit, then a singularity should have no impact on adjacency of the layers which v_i^p and v_j^q belong to. There is some degree of independence suggesting a degree of *uniform adjacency* between certain layers of adjacent regions. Whether this property is exactly ULA is an open question, but one which merits the use of ULA until the proper constraint(s) can be derived.

On the other hand, the existence of pairs v_i^p and v_j^q which *do not* approach each other in the limit gives a mechanism for identifying disjoint components in $f(X)$. Clearly, y_1^3 and y_2^4 do not approach each other, nor do y_0^3 and y_1^4. If the set S is enlarged to include samples between x_3 and x_4, and it's found that the same lack of convergence occurs, then this would be strong evidence for the existence of disjoint components or layers in $f(X)$. If an optimal assignment A on $\mathbb{G}$ is found and a lack of convergence is discovered, any edge e_{ij}^{pq} in $A(\mathbb{G})$ whose flow value x_{ij}^{pq} is 1, but whose *weight* w_{ij}^{pq} - which corresponds to angular distance - is not sufficiently close to zero can be be deleted from $A(\mathbb{G})$. Deleting all offending edge stacks could provide a more accurate approximation. Figure 7.2 does this for our annulus example.



(a)



(b)

Figure 7.1: (a) The annulus $f(X)$ of Section 2.1 and (b) and optimal linear approximation to it based on the sampling regime.

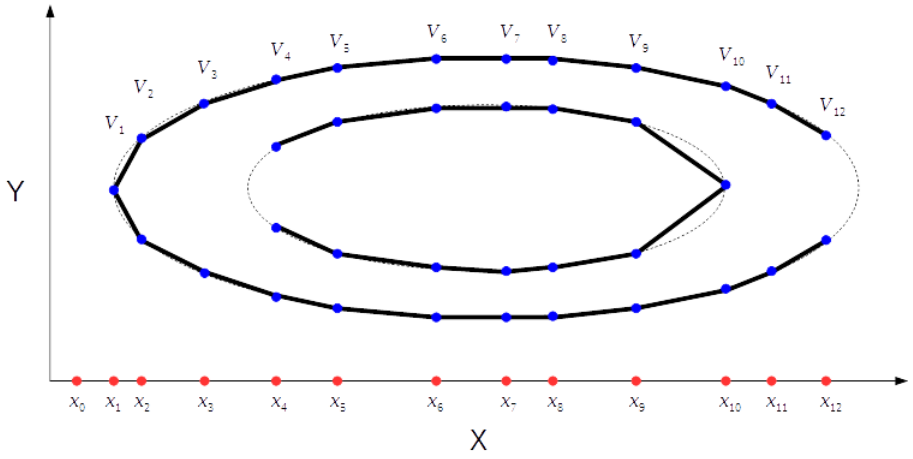


Figure 7.2: More accurate approximations of $f(X)$ can be had removing the edges between successive image elements y_a^i and y_b^{i+1} whose mutual distance are suspected to be non-diminishing in the sampling limit.

Chapter 8

Conformation Space Simulations

In Chapters 2 and 3 we explored the mathematics of stacked graphs and, in Chapters 5 through 7, approximated the conformation spaces of kinematic chains and nucleic acid phosphate backbones using these structure. The conformation space problem for was reduced, in Chapter 7, to finding an optimal feasible assignment, a problem that was solved in Chapter 4 through the construction of a linear program that satisfied the assignment feasibility conditions of Theorem 2.31. This chapter develops a collection metrics to study how well an optimal assignment approximates a conformation space. In addition, it generalizes what has thus far been the study of *rigid* chains to *semi-rigid* chains, allowing us to account for flexibility in chemical bond lengths and angles.

For this chapter, all stacked graphs $\mathbb{G}$ are assumed to be associated with normalized lattices generated by sampling a chain conformation space using the bisection technique described in Chapter 6 at some fixed resolution (Definition 7.4). Unless otherwise stated, all assignments A are assumed to be optimal. Graphs of the form $A(\mathbb{G})$ will henceforth be termed *conformation graphs*. When referring to normal coordinates and vertex stacks, Norm(V_k) will be used as a short hand for 'Norm(S_k), where S_k is the sample associated with V_k' (see Definition 7.4).

Definition 8.1 (Similar and Dissimilar Stacked Graphs)**.** Let $\mathbb{G}$ and $\mathbb{G}'$ be generated by two different chain parametrizations. If the parametrizations are approximately the same, then $\mathbb{G}$ and $\mathbb{G}'$ are *similar*, otherwise they are *dissimilar*.

Whether two parametrizations are 'approximately the same' is context dependant. For our purposes, parametrizations drawn from the same distributions will usually be similar.

8.1 Comparing Normalized Workspaces

This section discusses four possible congruence metrics for comparing stacked graphs: Ω_{A^*}, Ω_V, $\Omega_{\mathcal{R}}$, and $\Omega_{\mathcal{H}}$. Additional metrics are certainly possible.

8.1.1 A^* Congruence Metric Ω_{A^*}

In Chapter 2 we introduced the identity assignment A^* (Definition 2.38). It turns out that, besides being the graph dual of the identity element in $\mathbb{S}_m$ (Fact 3.3), A^* is potentially an optimal feasible assignment for asingular edge stacks at infinite sampling resolutions. This applies only in the case where an absolute torsion angle (Definition 6.9) or equivalent ordering is used. The result stems from the continuous nature of conformation spaces.

Recall from Theorem 7.2 and its proof that for pairs of adjacent samples S_p and S_q in the same region, there exists, for each sample s_u^p, precisely one sample s_v^q such that the angular distance between s_u^p and s_v^q approaches 0° as the distance between S_p and S_q approaches zero[40]. In the context of stacked graphs, this means that the weight for edge e_{uv}^{pq} is nearly zero at sufficiently high resolutions, with $u = v$ under absolute ordering. The absolute torsion angle ordering ensures that, at sufficiently high resolutions, two vertices having the same second index, belonging to adjacent vertex stacks, and existing in the same region, have nearly identical corresponding angular solution vectors (Definition 6.7).

The implication is that at high resolutions A^* is a nearly optimal (i.e. minimally weighted feasible) assignment for asingular E_{pq}, and by extension, for every region $\mathcal{R}_i \in \mathbb{G}_{\mathcal{R}}$. The graph dual interpretation is that $\alpha_{pq} \approx \imath$ for all p, q.

With this in mind, it's possible to define a single value that reflects how well a particular sampling regime approximates the finite IK components of a conformation space.

Definition 8.2 (A^* Convergence Metric Ω_{A^*})**.** Let A be any feasible, though not necessarily optimal, assignment on some $\mathbb{G}$. Then the A^* *convergence metric* for $A(\mathbb{G})$ is

$$\Omega_{A^*}(A(\mathbb{G})) := \frac{\sum\limits_{A_{pq} \in A(\mathring{\mathbb{G}})} \sum\limits_{i=0}^{|V_p|} x_{ii}^{pq}}{\sum\limits_{E_{pq} \in \mathring{\mathbb{G}}} |V_q|}. \tag{8.1}$$

Equivalently, Ω_{A^*} is the ratio between the number of identity transpositions (ii) in the graph dual γ of $\mathbb{G}$ and the total number of possible identity transpositions[41].

Lemma 8.3. *If* $x_{ij} \in \{0, 1\}$, $\Omega_{A^*}(A(\mathbb{G})) \to 1$ *as the sampling resolution of* $\mathbb{G}$ *approaches infinity.*

[40] i.e. as the radius of B_p goes to zero.

[41] Under the assumption that the transposition representation is minimal (Section 3.1).

One issue with using Ω_{A^*} is that the value can be misleading at low resolutions. It's entirely possible that $\Omega_{A^*}(A(\mathbb{G})) = 1$, with Ω_{A^*} decreasing dramatically before converging once more to 1 at high resolutions.

Another issue is that the convergence rate of Ω_{A^*} is both unknown and likely dependent on the details of the conformation space being approximated. Convergence is predicted to be non-uniform within a region as areas near singular boundaries are expected to converge slower then areas distant from them. This prediction is based on the kinematic singularities that haunt singular edge stacks, which can produces extremely high angular velocities [Ang07], manifesting as large angular distances between the solutions of adjacent samples even when resolutions are high. It may, however, be possible to approximate convergence rates by locally sampling conformation spaces at high resolutions and averaging the resulting convergence values. This approach is not covered here.

8.1.2 Vertex Stack Congruence Metric Ω_V

When comparing two similar stacked graphs $\mathbb{G}$ and $\mathbb{G}'$, it's not necessary that both exist at the same resolution. One can compare vertex stacks which exist in both $\mathbb{G}$ and $\mathbb{G}'$, corresponding to vertex stacks having the same normal coordinates (Definition 6.4).

Definition 8.4 (Shared Vertex Stacks $\cap_V$)**.** For any two $\mathbb{G}$ and $\mathbb{G}'$, the *shared vertex stacks of* $\mathbb{G}$ *and* $\mathbb{G}'$ are

$$\mathbb{G} \cap_V \mathbb{G}' := \{V_i | V_i \in \mathbb{G}, V_j' \in \mathbb{G}', \text{Norm}(V_i) = \text{Norm}(V_j')\}. \tag{8.2}$$

Definition 8.5 (Vertex Stack Congruence Metric Ω_V)**.** Let $\mathbb{G}$ and $\mathbb{G}'$ be two stacked graphs and

$$\tilde{\mathcal{V}} := \mathbb{G} \cap_V \mathbb{G}'. \tag{8.3}$$

Then the *vertex stack congruence metric of* $\mathbb{G}$ *with respect to* $\mathbb{G}'$ is

$$\Omega_V(\mathbb{G}, \mathbb{G}') := 1 - \frac{\sum_{\tilde{V}_i \in \tilde{\mathcal{V}}} ||V_i| - |V_i'||}{\sum_{\tilde{V}_i \in \tilde{\mathcal{V}}} |V_i|}. \tag{8.4}$$

For graph duals, the metric is a comparison between the size of the symmetric groups that a vertex dual belongs to in γ and γ'.

Though simplistic, Ω_V is straightforward to calculate, providing a quick and dirty comparison technique that doesn't require expensive assignment calculations. Intuitively, if $\mathbb{G}$ and $\mathbb{G}'$ are similar and $\Omega_V(\mathbb{G}, \mathbb{G}') \approx 1$, then the conformational spaces should be nearly identical.

Lemma 8.6. *For any two* $\mathbb{G}$ *and* $\mathbb{G}'$ *generated by the same chain parametrizations, but possibly at different sampling resolutions,* $\Omega_V(\mathbb{G}, \mathbb{G}') = 1$.

8.1.3 Regional Congruence Metric $\Omega_{\mathcal{R}}$

Higher level comparisons between similar stacked graphs begin with identifying regions between them.

Definition 8.7 (Regional Congruence Metric $\Omega_{\mathcal{R}}$)**.** Let $\mathbb{G}_a$ and $\mathbb{G}_b$ be similar stacked graphs at the same resolution. Let $\mathbb{G}_c$ be a stacked graph consisting of two vertex stacks V_a and V_b, where

$$V_a := \{v_i^a | \mathcal{R}_i \in \mathbb{G}_a\} \tag{8.5}$$

and

$$V_b := \{v_j^b | \mathcal{R}_j \in \mathbb{G}_b\}, \tag{8.6}$$

and a single edge stack E_{ab}. Assign to each e_{ij}^{ab} a positive weight w_{ij}^{ab}. Then, when $A(\mathbb{G}_c)$ is optimal.

$$\Omega_{\mathcal{R}}(\mathbb{G}_a, \mathbb{G}_b) := 1 - A(\mathbb{G}_c) \tag{8.7}$$

is a *regional congruence metric* if the weights w_{ij}^{ab} are defined such that

$$\Omega_{\mathcal{R}}(\mathbb{G}_a, \mathbb{G}_b) \to 1 \tag{8.8}$$

as the parametrizations of $\mathbb{G}_a$ and $\mathbb{G}_b$ approach each other.

If $\mathcal{R}_i^a \in \mathbb{G}_{\mathcal{R}}^a$ and $\mathcal{R}_j^b \in \mathbb{G}_{\mathcal{R}}^b$ are adjacent in $A(\mathbb{G}_c)$[42], then $\mathcal{R}_i^a$ and $\mathcal{R}_j^b$ *identify* with each other, denoted by $\mathcal{R}_i^a \equiv \mathcal{R}_j^b$. Otherwise, $\mathcal{R}_i^a$ and $\mathcal{R}_j^b$ *do not* identify with each other, or $\mathcal{R}_i^a \not\equiv \mathcal{R}_j^b$.

The optimal feasible assignment $A(\mathbb{G}_c)$ identifies, for each region in $\mathbb{G}_a$, the region in $\mathbb{G}_b$ that best correlates with it, based on the weighting metric chosen. If the parametrizations of $\mathbb{G}_a$ and $\mathbb{G}_b$ are sufficiently close, then E_{ab} is likely asingular and the total weight of $A(\mathbb{G}_c)$ should be nearly zero regardless of the weighting scheme used. One weighting scheme is based on *centres of mass*.

Definition 8.8 (Centre of Mass)**.** Let $\mathcal{V}$ be a set of vertex stacks V_i associated with some $\mathbb{G}$. Assign to each V_i a mass $\text{Mass}(V_i)$. Then $\text{Norm}(\mathcal{V})$ are the normal coordinates for the centre mass of the vertex stacks using normal coordinates. The centre of mass for $\mathbb{G}$ is $\text{Norm}(\mathbb{G})$.

[42] That is, if $x_{ij}^{ab} = 1$.

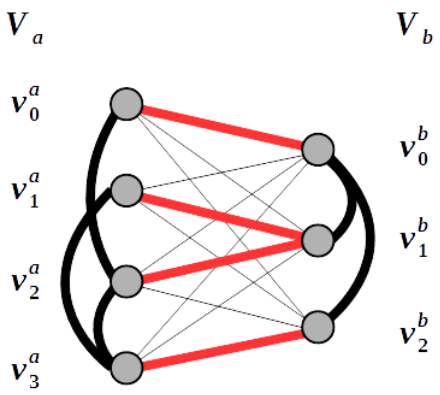


Figure 8.1: The regions of both $\mathbb{G}_a$ and $\mathbb{G}_b$ are treated as vertices in vertex stacks of a new $\mathbb{G}_c$. Edges weights (suppressed) between 'region vertices' are a measure of difference between the two regions, such as distance between centres of mass. The optimal feasible assignment A on $\mathbb{G}_c$ (red lines) identify the regions of $\mathbb{G}_a$ with those of $\mathbb{G}_b$. Bold black edges indicate adjacency between regions in the same stacked graph.

In most cases, the mass of each vertex stack will all be 1. Regions, being stacked graphs, have centres of mass. The weights w_{ij}^{ab} are simply the euclidean distance between Norm($\mathcal{R}_i^a$) and Norm($\mathcal{R}_j^b$).

One problem with $\Omega_{\mathcal{R}}$ is that it doesn't necessarily preserve regional adjacency. In Figure 8.1, $\mathcal{R}_0^a$ and $\mathcal{R}_2^a$ are both adjacent in $\mathbb{G}_a$, and are identified with the $\mathcal{R}_0^b$ and $\mathcal{R}_1^b$, respectively, which are likewise adjacent in $\mathbb{G}_b$. This is to be expected as $\mathbb{G}_a$ and $\mathbb{G}_b$ are assumed to be similar. However, there is a problem with the identifications $\mathcal{R}_0^a \equiv \mathcal{R}_0^b$ and $\mathcal{R}_1^a \equiv \mathcal{R}_1^b$ as $\mathcal{R}_0^a$ and $\mathcal{R}_1^a$ are adjacent but $\mathcal{R}_0^b$ and $\mathcal{R}_1^b$ are not. The identification in Figure 8.1 does not preserve regional adjacency.

Adjacency can be preserved with the following restrictions.

Definition 8.9 (Regional Adjacency Constraint)**.** Let $\mathcal{R}_i^a$ and $\mathcal{R}_j^a$ be regions in $\mathbb{G}_a$, $\mathcal{R}_x^b$ and $\mathcal{R}_y^b$ be regions in $\mathbb{G}_b$, with $\mathbb{G}_a$ and $\mathbb{G}_b$ similar. Then

$$\mathcal{R}_i^a \equiv \mathcal{R}_x^b \text{ and } \mathcal{R}_j^a \equiv \mathcal{R}_y^b \tag{8.9}$$

only if

$$\hat{\mathcal{E}}_{ij} = \emptyset \text{ and } \hat{\mathcal{E}}_{xy} = \emptyset \tag{8.10}$$

or

$$\hat{\mathcal{E}}_{ij} \neq \emptyset \text{ and } \hat{\mathcal{E}}_{xy} \neq \emptyset. \tag{8.11}$$

The regional adjacency constraint is similar to the non-basic ULA constraint in Section 2.5.

Note that Definition 8.9 says nothing about adjacency when two regions in $\mathbb{G}_a$ identify with the same region in $\mathbb{G}_b$, and vice versa. The identifications of $\mathcal{R}_1^a$ and $\mathcal{R}_2^a$ with $\mathcal{R}_1^b$ in Figure 8.1 are therefore valid.

Like Ω_V, $\Omega_\mathcal{R}$ can be performed in the absence of assignments.

8.1.4 Layer Congruence Metric $\Omega_\mathcal{H}$

When the regions of similar $\mathbb{G}_a$ and $\mathbb{G}_b$ have been identified and optimal assignments have been calculated for both, the next natural comparison is between the layers of both assignments. The process is identical to regional identification, save for being based on regional stacked graphs (Definition 2.35).

Definition 8.10 (Layer Congruence Metric $\Omega_\mathcal{H}$)**.** Let $\mathbb{G}_a$ and $\mathbb{G}_b$ be similar, have identified regions, and have optimal assignments A_a and A_b, respectively. If the regional stacked graphs of $\mathbb{G}_a$ and $\mathbb{G}_b$ are $\mathbb{R}_{\mathbb{G}_a}$ and $\mathbb{R}_{\mathbb{G}_b}$, respectively, denote their vertex stacks as V_i^a and V_j^b, respectively.

Create a new stacked graph $\mathbb{G}_c$ composed of the vertex stacks of both $\mathbb{R}_{\mathbb{G}_a}$ and $\mathbb{R}_{\mathbb{G}_b}$, where $E_{pq}^{ab} \in \mathbb{G}_c$ if $\mathcal{R}_p^a \equiv \mathcal{R}_q^b$.

Assign to each ${}^{ab}e_{ij}^{pq}$ a positive weight ${}^{ab}w_{ij}^{pq}$. Then

$$\Omega_\mathcal{H}(\mathbb{G}_a, \mathbb{G}_b) := 1 - A(\mathbb{G}_c), \tag{8.12}$$

where $A(\mathbb{G}_c)$ is optimal, is a *layer congruence metric* if the weights ${}^{ab}w_{ij}^{pq}$ are defined such that

$$\Omega_\mathcal{H}(\mathbb{G}_a, \mathbb{G}_b) \to 1 \tag{8.13}$$

as the parametrizations of $\mathbb{G}_a$ and $\mathbb{G}_b$ approach each other.

Layer ${}^a\mathcal{H}_i^p$ in $A_a(\mathbb{G}_a)$ is adjacent to layer ${}^b\mathcal{H}_j^q$ if ${}^{ab}x_{ij}^{pq} = 1$. If so, they identify, ${}^a\mathcal{H}_i^p \equiv {}^b\mathcal{H}_j^q$. Otherwise, they do not identify, ${}^a\mathcal{H}_i^p \not\equiv {}^b\mathcal{H}_j^q$.

Distances between centres of mass, this time for layers in $A_a(\mathbb{G}_a)$ and $A_b(\mathbb{G}_b)$, provide a weighting scheme that produces a layer congruence metric.

As with $\Omega_\mathcal{R}$, when the regions of $\mathbb{G}_a$ and $\mathbb{G}_b$ consist of more than one layer each, issues of layer adjacency preservation in $\Omega_\mathcal{H}$ can arise. For example, let $\mathbb{G}_a$ and $\mathbb{G}_b$ consist of two regions each: $\mathcal{R}_0^a, \mathcal{R}_1^a, \mathcal{R}_0^b$, and $\mathcal{R}_1^b$, where $|\mathcal{R}_0^a| = 4$, $|\mathcal{R}_1^a| = 3$, $|\mathcal{R}_0^b| = 3$ and $|\mathcal{R}_1^b| = 4$. To simplify matters, assume that the derived assignments $A_a(\mathbb{R}_{\mathbb{G}_a})$ and $A_b(\mathbb{R}_{\mathbb{G}_b})$ are binary (Definition 2.36), and that layer adjacency is that described by the assignment in Figure 8.2. We'll also assume that regional identification is $\mathcal{R}_0^a \equiv \mathcal{R}_0^b$ and $\mathcal{R}_1^a \equiv \mathcal{R}_1^b$. Finally, let $A_c(\mathbb{G}_c)$ be as illustrated in Figure 8.2.

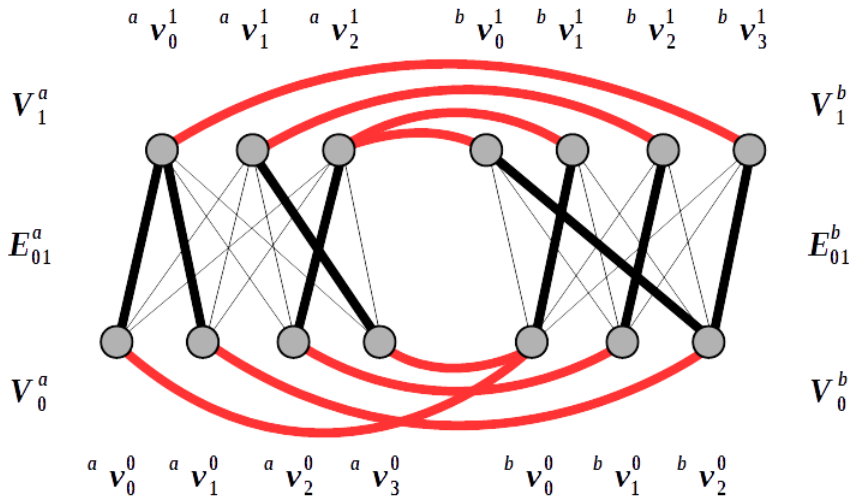


Figure 8.2: The assignments on E_{01}^a and E_{01}^b are the derived assignments $A_a(\mathbb{R}_{\mathbb{G}_a})$ and $A_b(\mathbb{R}_{\mathbb{G}_b})$, respectively, indicating layer adjacency. The red lines indicate the assignment $A_c(\mathbb{G}_c)$, which is also indicates layer identification.

Layer identities ${}^{a}\mathcal{H}_1^0 \equiv {}^{b}\mathcal{H}_2^0$ and ${}^{a}\mathcal{H}_0^1 \equiv {}^{b}\mathcal{H}_3^1$ preserve layer adjacency since

$$ {}^{aa}x_{10}^{01} = 1 \text{ in } A_a(\mathbb{R}_{\mathbb{G}_a}) \tag{8.14} $$

and

$$ {}^{bb}x_{23}^{01} = 1 \text{ in } A_b(\mathbb{R}_{\mathbb{G}_b}) \tag{8.15} $$

On the other hand, we have ${}^{a}\mathcal{H}_2^0 \equiv {}^{b}\mathcal{H}_1^0$ and ${}^{a}\mathcal{H}_2^1 \equiv {}^{b}\mathcal{H}_1^1$, which don't preserve adjacency as

$$ {}^{aa}x_{22}^{01} = 1 \text{ in } A_a(\mathbb{R}_{\mathbb{G}_a}) \tag{8.16} $$

but

$$ {}^{bb}x_{11}^{01} = 0 \text{ in } A_b\mathbb{R}_{\mathbb{G}_b}. \tag{8.17} $$

Visually, two red edges ${}^{ab}e_{ij}^{xy}$ and ${}^{ab}e_{kl}^{wz}$ preserve adjacency if both ${}^{aa}e_{ik}^{xw}$ and ${}^{bb}e_{jl}^{yz}$ are black in Figure 8.2.

As with regions in $\Omega_{\mathcal{R}}$, layer adjacency in $\Omega_{\mathcal{H}}$ can be enforced with an additional constraint. The constraint is exactly the Uniform Layer Adjacency constraint discussed in Section 2.5 (Definition 2.42).

The assumption of binary $A(\mathbb{R}_{\mathbb{G}_a})$ and $A(\mathbb{R}_{\mathbb{G}_b})$ is a large one. In its absence, assignments are proportional, resulting in layer identifications that are also proportional. 'Proportional adjacency' is something we wish to avoid at this stage of

research, which is one of the reasons we introduced binary derived assignments to begin with (Definition 2.36). We also have reason to suspect that conformation space are binary in nature, as noted in Section 7.1.

It's possible to calculate $\Omega_{\mathcal{H}}$ in the absence of $\Omega_{\mathcal{R}}$, in which case the layers of $\mathbb{G}_a$ and $\mathbb{G}_b$ each form a single vertex stack. If $\mathbb{G}_a$ and $\mathbb{G}_b$ are sufficiently similar, then the loss of regional data should not impact optimal identifications.

We noted at the end of Section 2.3 that regional stacked graphs give a broad-scale description of the conformational pathways between conformations. Full conformation graphs are large, complex structures whose size is exponential to their sampling resolution (Equation 6.4), but regional stacked graph sizes eventually stabilize since the number of regions and layers for conformation graphs are finite[43]. If the resolution is high enough to pick out each region, general connectivity between conformations can be studied via the connectivity of layers on a graph of reasonable size.

8.1.5 Metrics At Infinite Resolutions

In the previous sections, we assumed that $\mathbb{G}$ and $\mathbb{G}'$ were similar, but still born from distinct chain parametrizations. An alternative is to assume that $\mathbb{G}$ and $\mathbb{G}'$ follow from the same parametrization but exist at different resolutions. Applying metrics Ω_V, $\Omega_{\mathcal{R}}$, and $\Omega_{\mathcal{H}}$ to different resolutions of the same parametrizations allows us to estimate the impact that increased sampling resolution has on conformation space approximations. In fact, Ω_{A^*} is such an estimation, since it's essentially comparing some conformation graph $\mathbb{G}$, which exists at a finite resolution, with what it should be at infinite resolution.

Definition 8.11 (Reflexive Congruence Metrics)**.** Let $\mathbb{G}_i$ be generated at resolution $r_i =< i, i, i, i, i, i >$ for some chain parametrization. Then $\Omega_V(\mathbb{G}_i, \mathbb{G}_j)$, $\Omega_{\mathcal{R}}(\mathbb{G}_i, \mathbb{G}_j)$, and $\Omega_{\mathcal{H}}(\mathbb{G}_i, \mathbb{G}_j)$ are *reflexive* congruent metrics. If $j = i + 1$ or $i = j + 1$, then they are *step-wise* reflexive.

If conformation graphs are generated iteratively through increasing resolutions of $n_c = n_e$, we can produce, for any single parametrization, a vector of four values which measure how accurate the approximations are.

[43]This follows from the proof of Theorem 7.2. For every non-boundary point $y_p \in Y$, there is an open m-ball centered on y_p that doesn't contain boundary points. It follows then that every connected component in $Y - B$, or region in Y, has a non-trivial extent. As X is of finite size, being the product finite intervals $[0°, 360°)$, so too must Y be a finite m-manifold. Thus there can only be a finite number of regions. By assumption, stacked graphs only involve finite IK solutions, so that every region has finitely manly layers. Therefore both the number of regions and layers in any chain conformation space are finite.

Definition 8.12 (Reflexive Congruence Vector)**.** Let $\mathbb{G}_i$ and $\mathbb{G}_{i+1}$ be stacked lattice graphs associated with a single chain parametrization with resolution

$$r_i =< i, i, i, i, i, i > . \tag{8.18}$$

If the weightings of $\Omega_{\mathcal{R}}$ and $\Omega_{\mathcal{H}}$ are such that

$$\lim_{i\to\infty} \Omega_{\mathcal{R}}(\mathbb{G}_i, \mathbb{G}_{i+1}) = 1 \tag{8.19}$$

$$\lim_{i\to\infty} \Omega_{\mathcal{H}}(\mathbb{G}_i, \mathbb{G}_{i+1}) = 1, \tag{8.20}$$

then the *step-wise reflexive congruence vector* is

$$\Omega(\mathbb{G}_i) := \left\langle \Omega_{A^*}(\mathbb{G}_i), \Omega_{A^*}(\mathbb{G}_{i+1}), \Omega_{\mathcal{R}}(\mathbb{G}_i, \mathbb{G}_{i+1}), \Omega_{\mathcal{H}}(\mathbb{G}_i, \mathbb{G}_{i+1}) \right\rangle \tag{8.21}$$

and the *total step-wise reflexive congruence metric* is

$$\Omega_{\mathbb{G}_i} := \frac{\sum_{j=0}^{3} \Omega(\mathbb{G}_i)[j]}{4} \tag{8.22}$$

where $\Omega_{\mathbb{G}_i} \to 1$ as $i \to \infty$.

With the advent of additional reflexive congruence metrics, the definitions of $\Omega(\mathbb{G}_i)$ and $\Omega_{\mathbb{G}_i}$ can be extended to accommodate them.

8.1.6 Metrics on Groups of Conformation Graphs

Ω_{A^*},Ω_V, $\Omega_{\mathcal{R}}$, and $\Omega_{\mathcal{H}}$ can be applied pair-wise to collections of similar stacked graphs in order to determine how chain parametrizations impact conformation space geometry.

Definition 8.13 (Collective Congruence Metrics)**.** Let $\mathfrak{G}$ be a set of m similar stacked graphs $\mathbb{G}_i$. The *collective* A^*, vertex stack, regional, and layer congruence metrics $\Omega^C_{A^*}, \Omega^C_V, \Omega^C_{\mathcal{R}}$, and $\Omega^C_{\mathcal{H}}$, respectively, are defined as the mean value of applying Ω_{A^*} to every $\mathbb{G}_i \in \mathfrak{G}$, and $\Omega_V, \Omega_{\mathcal{R}}$, and $\Omega_{\mathcal{H}}$ to every pair $\mathbb{G}_i, \mathbb{G}_j \in \mathfrak{G}$, respectively. For example

$$\Omega^C_V(\mathfrak{G}) := \frac{\sum_{\mathbb{G}_i\in\mathfrak{G}} \sum_{\mathbb{G}_j\in\mathfrak{G}, i\neq j} \Omega_V(\mathbb{G}_i, \mathbb{G}_j)}{\binom{m}{2}}. \tag{8.23}$$

The *collective congruence vector* is

$$\Omega^C(\mathfrak{G}) := \left\langle \Omega^C_{A^*}(\mathfrak{G}), \Omega^C_V(\mathfrak{G}), \Omega^C_{\mathcal{R}}(\mathfrak{G}), \Omega^C_{\mathcal{H}}(\mathfrak{G}) \right\rangle \tag{8.24}$$

while the *total collective congruence metric* is

$$\Omega^C_{\mathfrak{G}} := \frac{\sum_{j=0}^{3} \Omega^C(\mathfrak{G})[j]}{4}. \tag{8.25}$$

In the case of RNA, the stacked graphs $\mathbb{G}_i \in \mathfrak{G}$ are parametrized by values pulled from the distribution of backbone bond length and angle values (Tables 6.1 and 6.2). As $|\mathfrak{G}| \to \infty$, we may find that the values of $\Omega^C_{A^*}$, Ω^C_V, $\Omega^C_{\mathcal{R}}$, and $\Omega^C_{\mathcal{H}}$ converge to particular values. These values would represent the degree to which chain flexibility impacts conformation space geometry. More flexible chains, defined by wider parameter distributions, should have lower values than less flexible chains, defined by narrower distributions.

A variant of collective congruence metrics are the *mean* congruence metrics, which measure how strongly conformation graphs generated by parameter distributions diverge from that generated by mean values.

Definition 8.14 (Mean Congruence Metrics)**.** Let $\mathfrak{G}$ be a set of n similar stacked graphs $\mathbb{G}_i$ generated from a set of chain parameter distributions, where $\mathbb{G}_0$ is generated from the mean values. The *mean* A^* congruence metric is

$$\Omega^M_{A^*}(\mathfrak{G}) := \frac{\sum_{i=1}^{n-1} |\Omega_{A^*}(\mathbb{G}_0) - \Omega_{A^*}(\mathbb{G}_i)|}{n-1} \tag{8.26}$$

while the *mean* vertex stack, regional, and layer congruence metrics Ω^M_V, $\Omega^M_{\mathcal{R}}$, and $\Omega^M_{\mathcal{H}}$, respectively, are of the form

$$\Omega^M_V(\mathfrak{G}) := \frac{\sum_{i=1}^{n-1} \Omega_V(\mathbb{G}_0, \mathbb{G}_i)}{n-1}. \tag{8.27}$$

The *mean* congruence vector is

$$\Omega^M(\mathfrak{G}) := \left\langle \Omega^M_{A^*}(\mathfrak{G}), \Omega^M_V(\mathfrak{G}), \Omega^M_{\mathcal{R}}(\mathfrak{G}), \Omega^M_{\mathcal{H}}(\mathfrak{G}) \right\rangle \tag{8.28}$$

while the *total mean* congruence metric is

$$\Omega^M_{\mathfrak{G}} := \frac{\sum_{j=0}^{3} \Omega^M(\mathfrak{G})[j]}{4}. \tag{8.29}$$

Variant congruence metrics based on median values are also possible.

Mean (and median) congruence metrics are likely to be those most useful for our purposes, being measures of how the conformation space of a kinematic chain is impacted by converting its *rigid* (mean-/median-valued) components into *semi-rigid* components.

8.2 Averaged Stacked Graphs

The previous section developed a number of metrics for comparing different stacked graphs and assignments on them. We now turn to *averaging* a collection of stacked graphs.

Definition 8.15 (Averaged Stacked Graphs)**.** Let $\mathfrak{G}$ be a set of similar stacked graphs $\mathbb{G}_i$ all at the same resolution r and generated by the same sampling technique. The *averaged stacked graph of* $\mathfrak{G}$, denoted by $\bar{\mathbb{G}}$, is composed of the vertex stack set $\bar{\mathcal{V}}$. Vertex stacks $\bar{V}_k \in \bar{\mathcal{V}}$ are the set of vertices $\bar{v}_j^k$ returned by a *representative function* f acting on the set

$$\{v_j^p | v_j^p \in V_p \in \mathbb{G}_i \in \mathfrak{G} \ni \text{Norm}(V_p) = \text{Norm}(V_k),\ \forall p, i\}, \tag{8.30}$$

or the set of all vertices v_j^p across all $\mathbb{G}_i$ whose containing vertex stacks have the same normal coordinates as V_k. Each element in $\bar{v}_j^k \in \bar{V}_k$ is a *representative vertex*.

A representative function is one that either a) selects from a set of vertices a sub-set that represents the distribution of sample torsion vectors s_j^k associated with the vertex set, or b) creates a new, smaller set of vertices with representative sample torsion vectors.

Edges stack $\bar{E}_{pq}$ exists if $\bar{V}_p$ and $\bar{V}_q$ are adjacent in some $\mathbb{G}_i \in \mathfrak{G}$.

A common class of representative functions would be clustering algorithms, which aggregate sub-sets of vertices into groups based on their associated sample torsion angles. A set of representative vertices would be formed by either a single vertex drawn from each group or a single vertex created from each group whose torsion angles are some statistical average of its group. Detailed discussion of possible representative functions is beyond the scope of this work.

In the context of kinematic chains and phosphate backbones, $\bar{\mathbb{G}}$ describes a conformation space for a rigid chain containing the key characteristics of a semi-rigid chain. This rigid chain is *not necessarily* the rigid chain generated by mean valued chain parameters, whose conformation graph is $\mathbb{G}_0$ (Definition 8.14). By subjecting $\mathbb{G}_0$ and $\bar{\mathbb{G}}$ to congruence analysis, we can determine how well a rigid, mean valued conformation space models, 'on average', a semi-rigid conformation space, for a given set of representative functions

The benefit of using averaged stacked graphs over a collection $\mathfrak{G}$ is that only two optimal assignments need to be found, one each for $\mathbb{G}_0$ and $\bar{\mathbb{G}}$. In the previous section, comparing a collection of stacked graphs $\mathfrak{G}$ required, depending on the metric used, finding feasible/optimal assignments A_i for *every* $\mathbb{G}_i \in \mathfrak{G}$, in addition to the already time-intensive inverse kinematic sampling process (Chapter 6). Whereas a complete analysis of even a small sized $\mathfrak{G}$ may not be practical, an average comparison almost certainly is.

Finally, it's likely possible to generalize averaged stacked graphs so that each $\mathbb{G}_i \in \mathfrak{G}$ need not be at the same resolution.

8.3 Energetics and Simulations

For any single conformation graph $\mathbb{G}$ with feasible assignment A, $A(\mathbb{G})$ is a non-stacked graph on which simulations can be carried out using random processes. These processes require some kind of decision making variable. For phosphate backbones, the variables are driven by atomic forces.

Molecular energetics, the attractive and repulsive forces between atoms, provides a means of converting our conformation graphs $A(\mathbb{G})$ into digraphs and Markov chains. As each $A(\mathbb{G})$ is an approximation of some conformation space, or, in the case of $\bar{A}(\bar{G})$, an averaged approximation, any energetics model we incorporate need only be reasonably accurate. We do *not* need highly accurate quantum mechanical descriptions of inter-atomic forces as the approximate nature of our graphs will likely override the energetic precision such descriptions would provide. We can satisfy ourselves with any model that quickly calculates the potential energy contained by a single phosphate backbone with torsion angles equal to some solution s_i^j. Such a model could simply be the sum total of the Lennard-Jones potential [LMS03] between every pair of atoms in the backbone.

Regardless of the specific energetics model used, simulation begins by assigning an energetics value q_i to each vertex $v_i \in \mathbb{G}_\mathcal{C}$[44]. Next, each edge $e_{ij} \in \mathbb{G}_\mathcal{C}$ is assigned a weight

$$w_{ij} := q_j - q_i. \tag{8.31}$$

All energies can be considered finite, with any infinite or undefined values resulting from overlapping atomic nuclei being replaced with sufficiently large numbers. In addition, we treat e_{ij} and e_{ji} as distinct, directed edges so that $w_{ij} = -w_{ji}$.

Definition 8.16 (Energized Conformation Graphs)**.** Let $A(\mathbb{G})$ be a directed conformation graph with edge weights representing energy difference between vertices, as described above. Then $A(\mathbb{G})$ is an *energized* conformation graph. $A(\mathbb{G})^+$ is the

[44] Vertex/sample solution energetics are independent of any assignment

positive sub-graph consisting of all edges with positive weights, while $A(\mathbb{G})^-$ is the *negative* sub-graph consisting of all edges with negative weights.

Lemma 8.17. *If $A(\mathbb{G})$ is an energized conformation graph, then $A(\mathbb{G})$ contains no directed cycles with all negative edge weights. $A(\mathbb{G})^-$ and $A(\mathbb{G})^+$ are either trees or forests.*

Proof. Every vertex has a single, finite, real-valued energy, implying that there is no infinite, repeating sequence of vertices

$$\cdots - v_0 - v_1 - v_2 - \cdots - v_n - v_0 - v_1 - \ldots \tag{8.32}$$

whose energies are monotonically decreasing. Therefore there are no directed cycles consisting entirely of negative edge weights. Since $A(\mathbb{G})^+$ is $A(\mathbb{G})^-$ with edge directions and weight signs reversed, and as $A(\mathbb{G})^-$ contains no cycles, neither does $A(\mathbb{G})^+$. Both must therefore be trees or forests. □

Lemma 8.17 provides a simple way of identifying stable conformations and conformational pathways.

Lemma 8.18 (Stable Conformations and Conformational Pathways in $A(\mathbb{G})^-$)**.** *Every leaf in $A(\mathbb{G})^-$, or vertex with no out-going edges, is a stable conformation and every directed path is a conformational pathway or conpath.*

Proof. A conformation is stable if there are no lower energy conformations it can move into. This is indicated by an outgoing, negative edge. Thus, every leaf in $A(\mathbb{G})^-$ is stable as it has no outgoing edges. □

The existence of multiple directed paths beginning at some vertex $v_i \in A(\mathbb{G})^-$ indicates that multiple conpaths are possible. It's reasonable to assume that the paths actual phosphate backbones will take are the most energetically favourable (i.e. minimally weighted) edges at each branching point. Still, it's possible that less than optimal, though still negative, paths are chosen, perhaps do to atomic forces external to the backbone interacting with it. Path-choice can be modelled probabilistically by replacing each edge weight w_{ij} with the proportional value

$$\tilde{w}_{ij} := \frac{q_j - q_i}{\sum\limits_{k\,:\,e_{ik} \in A(\mathbb{G})^-} (q_k - q_i)} \tag{8.33}$$

with $\tilde{w}_{ii} := 1$ when there are no outgoing edges from v_i.

Probabilistic forms of energized $A(\mathbb{G})^-$ have a useful property.

Theorem 8.19. *Probabilistic, energized $A(\mathbb{G})^-$ are absorbing Markov chains* [45].

Proof. The definition of $\tilde{\omega}_{ij}$ implies that $A(\mathbb{G})^-$ is a Markov chain as the sum of all edge weights leaving any v_i is 1. That $A(\mathbb{G})^-$ is absorbing is implied by Lemma 8.17: every conpath eventually ends in a stable conformation, which is an absorbing state. □

The calculation for determining the probability that any path starting at v_i will end at any particular absorbing state is well known for absorbing Markov chains [TK98]. Given the torsion angles for some initial RNA conformation and a probabilistic conformation graph $A(\mathbb{G})^-$, we can therefore predict which final conformation it is most likely to stabilize at, up to the resolution of $A(\mathbb{G})^-$. This applies equally to rigid, mean valued conformations graphs $A(\mathbb{G}_0)$, semi-rigid, averaged conformation graphs $A(\bar{\mathbb{G}})$, and collections of conformation graphs $\mathfrak{G}$ with their associated assignments. In the third case, the long-term behaviour of each $A(\mathbb{G}_i)$ Markov chain is calculated and compared.

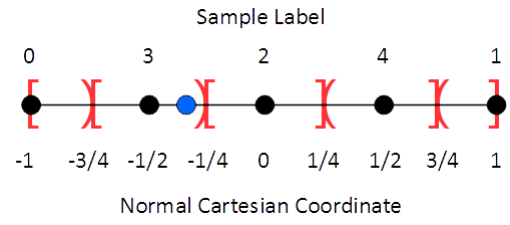
Additional analysis techniques can be used to identify which vertices contribute most to these probabilities. The implication is that these vertices are conformational bottlenecks, which are precisely those conformations that would be targeted for drug development (Chapter 1).

It's possible to study pathways on an energized conformation graph without removing $A(\mathbb{G})^+$. To account for energetic fluctuation in a backbones environment, a random value can be add or subtracted from each conformation's energy after each movement on the graph. While complicating the analysis, it may help to illuminate transient pathways, pathways which, on average, are only mildly unfavourable do to slightly positive edge weight values. Edges in $A(\mathbb{G})^+$ which are close to zero may provide occasional energetically favourable pathways which would circumvent the effects of conformation-targeting drugs.

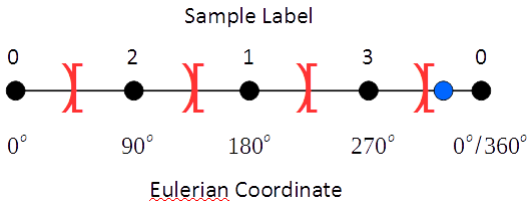
8.4 Model Validation and Real-World Data

One way to validate our model is to compare the predicted distributions of conformations using absorbing Markov chains against real-world data, such as the Nucleic Acid Database's massive collection of RNA crystallography data. At the time of this writing, the online database consists of over 1100 pure RNA structures, some composed of several hundred nucleotides, with each nucleotide representing

[45] A Markov chain is absorbing if there is at least one absorbing state (i.e. a state where the probability of leaving it is 0) and every state can reach an absorbing state [TK98].



(a)



(b)

Figure 8.3: Sample names of real-world data (blue circles) are assigned new names and coordinates to match the theoretical sample (black circles) contained by the half-open lattice interval the real-world data falls in. These intervals are based on the lattice's resolution. For the cartesian case (a) with $n_c = 2$, a real-world data point with normal coordinate -0.489 would be labelled 3 with coordinate $-\frac{1}{2}$. In the eulerian case (b) with $n_e = 2$, 348.9° would be labelled as sample 0 with coordinate 0°.

a stable conformation[46]. Unfortunately, NDB's RNA data does not come ready-packaged with normalized chain parametrizations. Each nucleotide needs to be processed. The process of converting raw NDB data into an appropriate chain format is a technical exercise we'll not undertake here. For our purposes we'll assume that any real-world conformation data we have access to is in such a format.

There is also the issue that real-world chains involve real-valued workplace sampling locations which do not fit nicely on the vertices of our sampling lattices (Chapter 6). This problem can be resolved by grouping data based on lattice resolution. Figure 8.3 illustrates one such grouping technique.

A small difficulty arises when we consider that there are two related sets of val-

[46]Though not necessarily energetically favourable. See below.

ues: euclidean sample names and torsion angles. Recall that every euclidean sample corresponds to a specific set of torsion angles, or inverse kinematic solutions (Sections 5.2 and 6.2). However we group real-world data on our sampling lattices, the correspondence needs to be consistent: if a data point is grouped with sample S_6, then its associated torsion angles need to be sufficiently close to some solution s_i^6. What constitutes 'sufficiently close' is debatable, but for averaged conformation graphs, within two standard deviations of the averaged solutions' distribution should suffice.

Assume, then, that we have both a technique for grouping data as well as a method for ensuring consistent correspondence between sample grouping and sample solutions. Comparing model predictions with real-world data amounts to counting the number of real-world data points existing in each lattice grouping, finding the proportion of all counts in each group, and comparing these proportions to the long term behaviour of our absorbing Markov chains. If our model is accurate, the proportions should be similar. If not, something may be amiss.

There is a hidden assumption we're making in this comparison which is that real-world data is energetically favourable. If we find that a large proportion of these nucleotides exist in conformations which we predict to be unstable, then it need not mean our approach is flawed. Recall that we're modelling a single nucleotide backbone in isolation of its nucleoside (Chapter 1). Additional energetics are involved with stacking and base pairing of nucleosides in polynucleotides which may force individual backbones into unstable conformations while maintaining an overall energetically favourable conformation for an entire strand.

A second validation technique is to compare the conformational pathways of energized conformation graphs with those produced by other, more sophisticated molecular modelling software, ones wherein detailed energetics calculations are used. Existing software is computationally expensive[47]. Should stacked graphs and assignments provide similar, though perhaps less accurate, results than other techniques at a fraction of the time, the former could be used to guided simulations on the latter. Kinematic simplification of transition structures may prove useful to force field algorithms by way of providing ranges of probable conformations.

[47]Based on discussions with Al Vasius

Chapter 9

Concluding Remarks and Future Research

The aim of this thesis has been to develop a technique for studying how nucleic acids move, in particular, ribonucleic acid. Three research goals were outlined in Chapter 1 which were formally met in Chapter 8

1. The approximation of conformation spaces for semi-rigid kinematic chains, in the form of optimal assignments on averaged stacked graphs;

2. The identification of energetically favourable backbone conformations, in the form of absorbing states in Markov chains;

3. The identification of energetically favourable conformational pathways, in the form of directed paths on negative energized conformation graphs $A(\mathbb{G})^{-}$.

It was in the chapters that preceded it, however, that the goals were substantively developed. The technique was developed conceptually, if not entirely chronologically, as follows. We began by restricting our attention to the phosphate backbone of a single nucleotide as it provides the lowest level restrictions on movement. The full range of motion for a single backbone was viewed as a pre-image problem, namely, finding an approximation to Q^{-1} where Q is continuous (Equation 5.3) and forms a connected manifold [McC90]. To do this, we took advantage of the existence of high-speed algorithms for finding the inverse kinematics of $6R$ kinematic chains [MC94], which a single backbone can be modelled as, to sample Q^{-1}. These samples were 'stitched together' to produce stacked graphs, on which optimal feasible assignments formed the approximations of Q^{-1}. Finally, we developed a number of metrics for investigating the accuracy of the approximation, generalized approximations to account for semi-rigid nature of phosphate backbones, and converted the approximations into absorbing Markov chains to find stable conformations and conformational pathways.

The end result is a graph $A(\mathbb{G})$ on which walks form a sequence of conformational changes in a single phosphate backbone; small, discrete changes in torsion angles that approximate continuous motion. Determining whether one backbone

conformation can be reached from another conformation is reduced to finding a low energy path between two vertices. While there nevertheless exist computational barriers to be overcome, namely that of finding large numbers of IK solutions and solving collections of linear programs, the computational burden is all upfront: once solutions and assignments have been found, the actual process of finding pathways is straightforward.

There remains, of course, the issue of validation. This refers not solely to whether the technique presented above performs as required - namely, whether approximations to the conformation problem can be produced with a reasonable degree of accuracy - but also whether those approximations can be had within a reasonable period of time. The computational requirements for producing high resolution maps of a single phosphate backbone conformation space are dependent on both the speed of inverse kinematic algorithms (Section 6.1) and on the formulation of feasibility algorithms, such as that outlined in Chapter 4. Fortunately, the nature of the technique is less restricted in terms of production than theory. Assuming that, practically, useful approximations to the conformation space can be had, the issue of timely production is of limited import. The nature of the technique allows for improvements to be had over a period of time. Samples can gradually be added and optimal assignments found quickly on local regions. While local optimization is may not yield global optimality (Section 4.1 and Figure 4.5), actual globally optimal assignments are not necessarily necessary, as we noted in the previous chapter. We *know* what an optimal feasible assignment looks like with a sufficiently large number of samples - the A^* congruence metric Ω_{A^*} (Section 8.1.1). What will likely be of most interest in conformation spaces are the areas near singular boundaries, specifically, the connectivity between layers of distinct regions. Regions and layers are themselves dull structures, uniform regions of space that may very well be represented by single vertex stacks[48]. Once singular boundaries have been delineated at a particular sampling resolution, further approximation improvements will almost certainly take the form of increased sampling along these boundaries to refine them. Rather then improving global resolution, restricted, local improvements should greatly reduce overall computational burdens.

With regards to future research beyond that of model validation, there exists at least one other application for stacked graphs. During its development, I discovered that it can also be used for modelling certain kinds of object tracking problems, which could be explored. There is also the relationship between stacked graphs and symmetric groups which is in need of additional investigation, having only been briefly considered here. It is unknown whether there exist other symmetric group problems which can benefit from a graph dual interpretation. The issue of

[48]Hence regional stacked graphs $\mathbb{R}_{\mathbb{G}}$ (Definition 2.35).

improving the computational complexity of solving the assignment feasibility problem could also be addressed.

Finally, I think it important to note that the approach we've taken to address the conformation problem appears to be a novel one and is, at least conceptually, fairly straightforward. While a diverse range of mathematical knowledge is required to understand and develop it, the technical depth required of each field is limited. The graph theory component (Chapter 2) is elementary, as is the group theory (Chapter 3). Linear programming (Chapter 4), while necessary to *find* (optimal) feasible assignments, it is not required to determine necessary and sufficient conditions for feasibility (Theorem 2.31). Kinematic chains are used casually. What we require from them is the boundedness of the number of inverse kinematic solutions for our particular chain [Ang07], the existence of high speed algorithms to find these solutions [MC94], and the knowledge that they are continuous functions that produce manifolds [McC90]. Even the topological connection between stacked graphs and the conformation space problem (Chapter 7) is fundamental.

This is said not to discourage the technique. At its core, the complex issue of how to approximate the conformation space of a nucleic acid backbone has been reduced to solving a collection of interrelated assignment problems, problems which are well-know and simple to understand. That it can be done with a suite of seemingly mundane mathematics is, I believe, worthy of attention. Nearly every mathematical course I've taken as both an undergraduate and graduate student has directly contributed to the production of this thesis.

There is a scene early in Dumas' 'The Count of Monte Cristo' [Dum45] wherein Edmond Dantès, a prisoner of the Château d'If, is conversing with abbé Faria, another prisoner, in the former's cell. Dantès, awed by Faria's breadth of reading, remarks that the abbé must know many languages, to which Faria responds that he knows several and that:

> *'... I can [also] understand modern Greek with the help of Ancient Greek, but I speak it poorly; I am studying it now.'*
> *'You are studying it?' Dantès exclaimed.*
> *'Yes, I have compiled a vocabulary of the words that I know and have arranged them, combined them, turned them one way, then the other, so as to make them sufficient to express my thoughts. I know about one thousand words, which is all I absolutely need, though I believe there are a hundred thousand in dictionaries. Of course I shall not be a polished speaker, but I shall make myself understood perfectly, which is good enough.'*

I would be a poor student if I claimed that I knew all I absolutely needed and I can only hope that this work is as clear to others as it is to myself.

Bibliography

[AMO93] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, New Jersey, 1993. → pages 9, 20, 21, 29, 30, 56, 57, 59

[Ang07] J. Angeles. *Fundamentals of Robot Mechanical Systems: Theory, Methods, and Algorithms, 3rd Edition*. Mechanical Engineering Series. Springer, 2007. → pages 2, 68, 69, 70, 71, 72, 104, 120

[BHA15] M.P. Blakely, S.S. Hasnain, and S.V. Antonyuk. Sub-atomic resolution x-ray crystallography and neutron crystallography: Promise, challenges and potential. *IUCrJ*, 2(4):464–474, 2015. → pages 82

[Bol79] B. Bollobas. *Graph Theory: An Introductory Course*. Springer-Verlag, 1979. → pages 13, 29

[BW97] A.S. Brodsky and J.R. Williamson. Solution structure of the hiv-2 tar-argininamide complex. *J. Mol. Biol.*, 267(3):624–639, 1997. → pages 5, 6

[Dat16] Nucleic Acid Database. Ideal geometries [online]. 2016 [cited February 27, 2016]. → pages 2, 5, 79

[Dum45] A. Dumas. *The Count of Monte Cristo*. Penguin Classics, 1844-45. → pages 120

[Fou92] L.R. Foulds. *Graph Theory Applications*. Springer, 1992. → pages 9, 40

[Fra03] J.B. Fraleigh. *A First Course in Abstract Algebra, 7th Edition*. Addison Wesley, 2003. → pages 39, 40, 42

[GSC+96] A. Gelbin, B. Schneider, L. Clowney, S. Hsieh, Olson W.K., and H.M. Berman. Geometric parameters in nucleic acids: Sugar and phosphate constituents. *J. Am. Chem. Soc.*, 118(3):519–529, 1996. → pages 79, 80

[KHP11] K. S. Keating, E.L. Humphris, and A.M. Pyle. A new way to see rna. *Quarterly Reviews of Biophysics*, 44(4):433–466, 2011. → pages 1, 2

[LMS03] K.J. Laidler, J.H. Meiser, and B.C. Sanctuary. *Physical Chemistry, Fourth Edition*. Houghton Mifflin, 2003. → pages 113

[LNC93] A.L. Lehninger, D.L. Nelson, and M.M. Cox. *Principles of Biochemistry, 2nd Edition*. Worth, 1993. → pages 1, 3

[Mak08] C.H. Mak. Rna conformational sampling. i. single-nucleotide loop closure. *J. Comput. Chem.*, 29(6):926–933, 2008. → pages 1, 2, 4

[MC94] D. Manocha and J.F. Canny. Efficient inverse kinematics for general 6r manipulators. *IEEE Transactions on Robotics and Automaton*, 10(5):648–657, 1994. → pages 4, 73, 82, 118, 120

[McC90] J.M. McCarthy. *An Introduction to Theoretical Kinematics*. The MIT Press, 1990. → pages 68, 69, 95, 118, 120

[MCM11] C.H. Mak, W.Y Chung, and N.D. Markovskiy. Rna conformational sampling. ii. arbitrary length multinucleotide loop closure. *J. Chem. Theory Comput.*, 7:1198–1207, 2011. → pages 2

[MLL08] R.J. Milgram, G. Liu, and J.C. Latombe. On the structure of the inverse kinematics map of a fragment of protein backbone. *J. Comput. Chem.*, 29(1):50–68, 2008. → pages 1, 4

[Mun00] J.R. Munkres. *Topology, 2nd Edition*. Prentice Hall, 2000. → pages 95

[PRT+07] J.M. Porta, L. Ros, F. Thomas, F. Corcho, J. Cantó, and J.J. Pérez. Complete maps of molecular-loop conformation spaces. *J. Comput. Chem.*, 28(13):2170–2189, 2007. → pages 2, 4

[Rom12] S. Roman. *Fundamentals of Group Theory: An Advanced Approach*. SpringerLink, 2012. → pages 45

[TK98] H.M. Taylor and S. Karlin. *An Introduction to Stochastic Modelling, 3rd Edition*. Academic Press, 1998. → pages 115

[WKM+08] X. Wang, G. Kapral, L. Murray, D. Richardson, J. Richardson, and J. Snoeyink. Rnabc: Forward kinematics to reduce all-atom steric clashes in rna backbone. *J. Math. Biol.*, 56:253–278, 2008. → pages 1, 2

Appendices

Appendix A

Constructing Unique Subscripts

In the one dimensional case, where the name of any sample S_r is $< k >$, then setting $r = k$ is sufficient to provide a unique sample subscript. In higher dimensions, unique integral subscripts depend on the number of dimensions or length, of S_r. On way to create integral subscripts is via 'spiralling'. Figure A.1 demonstrates the process in the two-dimensional cartesian case. Higher dimensions with eulerian components are similarly constructed.

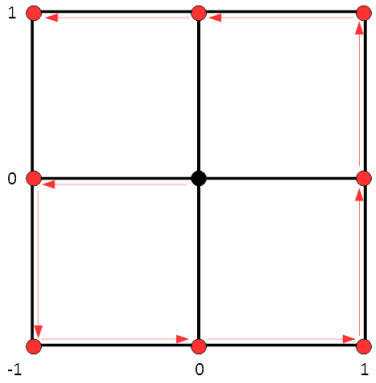
Table A.1 lists the first twenty-eight sample names generated from unique subscripts, in accordance with the process in Figure A.1.

Table A.1: The first 27 samples with unique subscripts generated for the two-dimensional cartesian case using a spiralling technique.

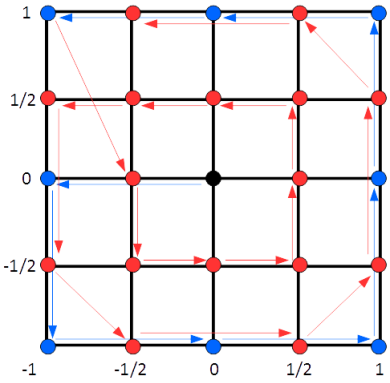
Sample	Name	Sample	Name
S_0	$< 0, 0 >$	S_{14}	$< \frac{1}{2}, \frac{1}{2} >$
S_1	$< -1, 0 >$	S_{15}	$< 0, \frac{1}{2} >$
S_2	$< -1, -1 >$	S_{16}	$< -\frac{1}{2}, \frac{1}{2} >$
S_3	$< 0, -1 >$	S_{17}	$< -1, \frac{1}{2} >$
S_4	$< 1, -1 >$	S_{18}	$< -1, -\frac{1}{2} >$
S_5	$< 1, 0 >$	S_{19}	$< -\frac{1}{2}, -1 >$
S_6	$< 1, 1 >$	S_{20}	$< \frac{1}{2}, -1 >$
S_7	$< 0, 1 >$	S_{21}	$< 1, -\frac{1}{2} >$
S_8	$< -1, 1 >$	S_{22}	$< 1, \frac{1}{2} >$
S_9	$< -\frac{1}{2}, 0 >$	S_{23}	$< \frac{1}{2}, 1 >$
S_{10}	$< -\frac{1}{2}, -\frac{1}{2} >$	S_{24}	$< -\frac{1}{2}, 1 >$
S_{11}	$< 0, -\frac{1}{2} >$	S_{25}	$< -\frac{1}{4}, 0 >$
S_{12}	$< \frac{1}{2}, -\frac{1}{2} >$	S_{26}	$< -\frac{1}{4}, -\frac{1}{4} >$
S_{13}	$< \frac{1}{2}, 0 >$	S_{27}	$< 0, -\frac{1}{4} >$

'Spiral integral' construction resembles the bisection technique for sample names (Figure 6.3). The pattern can spiral in or out, clockwise or counter-clockwise, and generates all subscripts for samples at and above a given resolution.

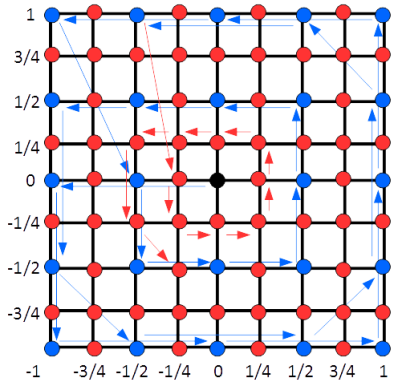
As this work does not define each sample S_r by unique subscripts, but instead by its name (Definition 6.3), and as there are a variety of spiralling patterns, an explicit algorithm for finding spiral integers based on sample names is not included.



(a) Lattice for $n_{c_0} = n_{c_1} = 0$ and 1



(b) Lattice for $n_{c_0} = n_{c_1} = 2$



(c) Lattice for $n_{c_0} = n_{c_1} = 3$

Figure A.1: One method for spiral integer construction on the square $[-1, 1]^2$ generated by n_{c_0} and n_{c_1}. Coordinates correspond to sample names (see Figure 6.3). (a) Beginning at the origin (black circle) and with counter-clockwise rotation, follow the red arrows to visit each sample generated by $n_{c_0} = n_{c_1} = 0$ (red vertices) and 1, labelling them by increasing integers with 0 for the origin (labels suppressed). Labels form the unique subscript values. (b) Add the samples generated by $n_{c_0} = n_{c_1} = 2$. The next label corresponds to the sample immediately left of the origin. Repeat step (a), labelling samples in a counter-clockwise direction, spiralling outwards. Vertices and arrows indicated in blue are those generated by previous values of n_{c_0} and n_{c_1}. (c) Repeat step (b) with $n_{c_0} = n_{c_1} = 3$. Only the first few red arrows are shown for readability.

Appendix B

Constructing Normal Coordinates

We begin with the cartesian case, followed by the eulerian case, both in one dimension. The higher dimensional cases are cartesian products of the single dimensional cases. The aim is to assign each sample a *rational sort value* $\frac{p}{q} \in [0, 1]$ in the cartesian case so that samples form a partially ordered set. The sort values are then transformed into normal coordinates in $[-1, 1]$. In the eulerian case, the rational sort value is in $[0, 1)$ with a normal coordinate value of $[0, 360°)$

Assume that sample S_r is generated by $n_c > 0$. The denominator q of its sort value is defined to be one less than the total number of samples generated by n_c bisections. From Table 6.4, $n_c = 0$ generates two samples, with each increment of n_c increases the number of solutions by a power of two. The total number of samples generated by n_c bisections is

$$\begin{aligned} q &= (2 + 2^0 + 2^1 + \cdots + 2^{n_c-1}) - 1 && \text{(B.1)} \\ &= 1 + \sum_{i=0}^{n_c-1} 2^i && \text{(B.2)} \\ &= 1 + \frac{1 - 2^{n_c}}{1 - 2} && \text{(B.3)} \\ &= 2^{n_c}. && \text{(B.4)} \end{aligned}$$

The numerator p of the sort value for S_r is the number of 'ticks' from the left-hand side that S_r is located at in Figure 6.3. For example, if $n_c = 2$, then $p = 1$ for sample 3 and $p = 3$ for sample 4.

According to Table 6.3, the *last* label generated by $(n_c - 1)$ is

$$2^{n_c-1}$$

so that if the sample name of S_r is k, then k is the

$$k - 2^{n_c-1} \qquad \text{(B.5)}$$

sample name generated by n_c bisections. Thus, label $2^{n_c-1} + 1$ should be at the first position, $2^{n_c-1} + 2$ should be at the second, etc. However, labels generated by n_c

have positions which *alternate* with those generated by $(n_c - 1)$, so that the former labels have *even* p values and the later have *odd* ones. This implies that

$$\begin{aligned} p &= 2(k - 2^{n_c - 1}) - 1 \\ &= 2k - 2^{n_c} - 1. \end{aligned} \tag{B.6}$$

The rational sort value $\frac{p}{q}$ for S_r is therefore

$$\begin{aligned} \frac{p}{q} &= \frac{2k - 2^{n_c} - 1}{2^{n_c}} \\ &= \frac{2k - 1}{2^{n_c}} - 1 \end{aligned} \tag{B.7}$$

when $n_c > 0$. For $n_c = 0$, there are only two samples, which are always at the end-points of the interval. We therefore define $p := 0$ and $p := 1$ for $k = 0$ and $k = 1$, respectively, giving rational sort values of 0 and 1 for S_r with names $k = 0$ and $k = 1$, respectively.

Conversion of rational sort values to normal coordinates is accomplished by transforming $[0, 1]$ into $[-1, 1]$

$$\begin{aligned} \frac{p}{q} * 2 - 1 &= \left(\frac{2k - 1}{2^{n_c}} - 1\right) * 2 - 1 \\ &= \frac{2k - 1}{2^{n_c - 1}} - 3. \end{aligned} \tag{B.8}$$

Rational sort values $\frac{p'}{q'}$ for eulerian components are generated in a manner similar to that above. If S_r has name k' and is generated by $n_e > 0$, then q' is

$$q' = \sum_{i=0}^{n_e - 1} 2^i \tag{B.9}$$

$$= \frac{1 - 2^{n_e}}{1 - 2} \tag{B.10}$$

$$= 2^{n_e} - 1 \tag{B.11}$$

as only one sample is generated by $n_e = 0$. The numerator remains the same at

$$p' = 2k' - 2^{n_e} - 1 \tag{B.12}$$

yielding a rational sort value of

$$\frac{p'}{q'} = \frac{2k' - 2^{n_e} - 1}{2^{n_c} - 1} \tag{B.13}$$

$$= \frac{2k'}{2^{n_e} - 1} - 1 \tag{B.14}$$

for $n_e > 0$. As with the cartesian components, for S_r with name $k' = 0$ generated by $n_e = 0$, the rational sort value is defined to be 0. This produces rational sort values in $[0, 1)$. Normal coordinates are formed by transforming $[0, 1)$ into $[0°, 360°)$

$$\frac{p'}{q'} * 360° = \left(\frac{2k'}{2^{n_e} - 1} - 1 \right) * 360° \tag{B.15}$$

In the general case of sample

$$S_k = \langle k_0, k_1, k_2, k_3, k_4, k_5 \rangle$$

with resolution

$$\text{Res}(S_k) = \left\langle n_{c_0}, n_{c_1}, n_{c_2}, n_{e_0}, n_{e_1}, n_{e_2} \right\rangle$$

we have the following construction for function Norm$(\cdot)$ (Definition 6.4)

$$\text{Norm}(S_k) = \langle w_0, w_1, w_2, w_3, w_4, w_5 \rangle$$

where

$$\begin{aligned} w_0 &= \tfrac{2k_0 - 1}{2^{n_{c_0}-1}} - 3, \; w_1 = \left(\frac{2k'_3}{2^{n_{e_0}} - 1} - 1 \right) * 360° \\ w_2 &= \tfrac{2k_1 - 1}{2^{n_{c_1}-1}} - 3, \; w_3 = \left(\frac{2k_4}{2^{n_{e_1}} - 1} - 1 \right) * 360° \\ w_4 &= \tfrac{2k_2 - 1}{2^{n_{c_2}-1}} - 3, \; w_5 = \left(\frac{2k_5}{2^{n_{e_2}} - 1} - 1 \right) * 360°. \end{aligned} \tag{B.16}$$