CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

PROTEIN FOLDING: PLANAR CONFIGURATION SPACES OF DISC ARRANGEMENTS AND HINGED POLYGONS: *PROTEIN FOLDING IN FLATLAND*

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

by

Clinton Bowen

August 2014

The thesis of Clinton Bowen is approved:

Dr. Silvia Fernandez — Date

Dr. John Dye — Date

Dr. Csaba Tóth, Chair — Date

California State University, Northridge

Table of Contents

ABSTRACT

PROTEIN FOLDING: PLANAR CONFIGURATION SPACES OF DISC ARRANGEMENTS AND HINGED POLYGONS: *PROTEIN FOLDING IN FLATLAND*

By

Clinton Bowen

Master of Science in Computer Science

abstract goes here

Chapter 1

Background

In this section, we cover the background subjects needed to formally pose the problem and present solutions in this thesis. We start with two types of combinatorial structures, linkages and polygonal linkages. We then discuss the configuration spaces of linkages and polygonal linkages. We then look into an alternate representation of linkages, disk arrangements and state the disk arrangement theorem. We then look at satisfiability problems and then review a framework, the logic engine, which can encode a type of satisfiability problem. Finally, we cover the basic definitions of algorithm complexity for **P** and **NP**.

1.1 Graphs

A *graph* is an ordered pair $G = (V, E)$ comprising of a set V of vertices and a set E of edges or lines. Every edge $e \in E$, is an unordered pair of distinct vertices $u, v \in V$ (the edge represents their adjacency, $\{u, v\} \in E$). With this definition of a graph, there are no loops (self adjacent vertices) or multi-edges (several edges between the same pair of vertices).

A motivation for using graphs is modelling physical objects like molecules. This requires an embedding into the plane or $\mathbb{R}^3$. An *embedding* of the graph $G = (V, E)$ is an injective mapping $\Pi : V \mapsto \mathbb{R}^2$ (see Figure 1.1).

graphics/PetersonGraph.png

Figure 1.1: An embedding of the Peterson graph.

1.1.0.1 Edge Crossings

We define *plane embeddings* to be an embedding where the following degenerate configurations do not occur:

(i) the interiors of two or more edges intersect, or

(ii) an edge passing through a vertex

graphics/crossingType2.png

(a) The interior of the edges intersect.

graphics/crossingType3.png

(b) An edge passes through a vertex.

Figure 1.2: These figures exhibit the 4 types of edge crossings.

A *planar graph* is an abstract graph that admits a plane embedding.

1.1.1 Trees

A *simple cycle* of a graph is a sequence of distinct vertices that do not repeat with the exception of the starting (ending) vertex. The vertices of an embedded tree have a nomenclature. One vertex is designated to be the *root* where the edges are oriented away from the root. Adjacent vertices have a parent-child relationship. The *parent* is closer to the root, while the *child* is further way from the root.

graphics/PetersonGraphWithSimpleCyle.png

(a) A proper embedding of the Peterson graph with a simple cycle of (2,7,9,6,1,0,5,8,3,2).

graphics/RandomTree16Nodes.png

(b) A tree with 16 vertices

1.1.2 Ordered Trees

An *ordered tree* is a rooted tree in which the counter-clockwise ordering of children of a common parent matters. The orderedness of a tree is a property of the tree's embedding. A fortiori, embeddings of ordered trees are equivalent if for each parent, the counter-clockwise ordering of children are the same.

graphics/orderedTreeExample1.png

(a) An embedding of a tree.

graphics/orderedTreeExample2.png

(b) Another embedding of a tree.

Figure 1.4: A rooted tree with whose children have a different counter-clockwise ordering. Note that these ordered trees are not equivalent.

1.1.3 Graph Isomorphism

To determine when two graphs are equivalent, we need to define an isomorphism for graphs. Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, a graph isomorphism $f : V_1 \mapsto V_2$ such that for any two vertices $u, v \in V_1$ that are adjacent, i.e. $(u, v) \in E_1$, if and only if $(f(u), f(v)) \in E_2$.

Graph	Vertices	Edges
G_1	$\{a,b,c,d,e\}$	$\{(a,b),(b,c),(c,d),(d,e),(e,a)\}$
G_2	$\{1,2,3,4,5\}$	$\{(1,2),(2,3),(3,4),(4,5),(5,1)\}$

Table 1.1: Two graphs that are isomorphic with the alphabetical isomorphism $f(a) = 1$, $f(b) = 2$, $f(c) = 3$, $f(d) = 4$, $f(e) = 5$.

graphics/graphIsomorphismExample.pdf

Figure 1.5: This figure depicts the graph isomorphism shown in Table (**??**) between V_1 and V_2 in the plane.

1.1.4 Linkages

When graphs model physical objects, distances between adjacent vertices matter. The length assignment of a graph $G = (V, E)$ is $l : E \mapsto \mathbb{R}^+$. A *linkage* is a graph $G = (V, E)$ with a length assignment $l : E \mapsto \mathbb{R}^+$. We consider embeddings of a graph that respects the length assignment. A *realization* of a linkage, G and l, is an embedding fo a graph, Π, such that for every edge $\{u, v\} \in E$, $l(\{u, v\}) = |\Pi(u) - \Pi(v)|$.

graphics/crossingEdgeLinkage.pdf

Figure 1.6: A linkage where edges cross however it does not contain loops or multiple edges between vertices.