Bounding the Genus of a Graph: A Project Proposal

Taylor Cox
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba
Email: coxt3@myumanitoba.ca

***Abstract*—An embedding of a graph G on some surface Σ is a drawing of G on Σ such that the edges of G only intersect at their endpoints, namely the vertices. The genus of a surface Σ is the number of handles present on Σ. The genus of a graph is the minimum number of handles that must be added to Σ in order for an embedding of G to be possible. The graph genus problem is given as follows: Given a graph G and a natural number k, determine whether G has genus k or less. In 1989, Thomassen proved that the graph genus problem is NP-Complete. However, in 1996 Mohar et al. gave a linear-time algorithm to solve the graph genus problem for a fixed k. The goal of this project is to investigate these papers and others to evaluate the current state of the graph genus problem. Approaches will then be developed to bound the genus of a graph G to a sufficiently small interval such that repeated execution of Mohar 1996 will be feasible. Therefore, the expected result of this project is a set of one or more heuristics that will reduce the search space for determining the genus of a graph.**

***Index Terms*—Genus,Embedding,Surfaces**

I. Introduction

Graph Embedding Problems lie at the intersection between Graph Theory and Topology. The problem in graph embedding of particular interest to this project is the graph genus problem, defined as follows: Given a Graph G = (V,E), find the minimum genus of G.

The remainder of this proposal is structured as follows: Section II gives a detailed definition of the graph genus problem. Section III constitutes a survey of literature relevant to the GGP. Next, section IV will discuss specific points of investigation. Section V will propose contributions related to those points, and section VI will detail the proposed programming components of this project.

II. Problem Statement

III. Relevant Literature

IV. Areas of Investigation

V. Proposed Contributions

VI. Programming Components

VII. Conclusion

References

[1]

[2]

[3]

[4]