

# Milestone Report — Optimization of Subgraph Isomorphism Algorithm

by Nelson Batista, Max Inciong, Francesca Truncale

Senior Project II

Professor Jianting Zhang

Fall 2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Ullmann Algorithm</b>	<b>2</b>
2.1	Naive Approach . . . . .	2
2.2	“Refine M” Procedure . . . . .	2
2.3	Full Algorithm . . . . .	2
<b>3</b>	<b>Implementation of the Algorithm</b>	<b>2</b>
3.1	Python Implementation and Performance . . . . .	2
3.2	Initial C++ Implementation . . . . .	2
3.3	Improved C++ Implementation . . . . .	2
3.4	Optimizations through OpenMP . . . . .	2
<b>4</b>	<b>Bibliography</b>	<b>2</b>

# 1 Introduction

The goal of this project is to understand and optimize an algorithm for solving the NP-complete problem of subgraph isomorphism. An *isomorphism* between two graphs is a mapping from the vertices of one graph, say  $G$ , to another graph, say  $H$ , such that any two vertices which are adjacent in  $G$  are mapped to vertices which are adjacent in  $H$ . The subgraph isomorphism problem, in turn, is the problem of determining whether an isomorphism exists between a graph  $G$  and any of the subgraphs of a larger search graph,  $H$ . The issue that makes the subgraph isomorphism problem so difficult to solve for a particular pair of graphs is cycling through the main graph and expanding each node within the graph and checking if each of the generated subgraphs formed from each node is isomorphic to the control graph.

Fortunately, many attempts exist to solve the subgraph isomorphism problem. Among the earliest of these is the famous *Ullmann Algorithm*, proposed by Julian R. Ullmann in his 1976 paper, *An Algorithm for Subgraph Isomorphism*. He first describes a basic, naive approach to finding a mapping from the vertices of a graph to a subgraph of a larger graph. He goes on to define a “refine” procedure to dramatically reduce the number of possible mappings that must be checked. These will be covered in greater detail in the next section.

The goal of our project was to take this algorithm and improve its performance. Subgraph isomorphism is an expensive problem to solve, and finding multiple possible isomorphisms from one graph to subgraphs of another is even more expensive, so even slight improvements in the algorithm’s performance is likely to lead to large performance improvements in any large-scale program which needs to use it often.

## 2 Ullmann Algorithm

### 2.1 Naive Approach

### 2.2 “Refine M” Procedure

### 2.3 Full Algorithm

## 3 Implementation of the Algorithm

### 3.1 Python Implementation and Performance

### 3.2 Initial C++ Implementation

### 3.3 Improved C++ Implementation

### 3.4 Optimizations through OpenMP

## 4 Bibliography