On Competition for Undergraduate Co-op Placements: A Graph Approach

by

Yuheng Jiang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2015

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The objective of this thesis is to improve the co-operative (co-op) education process by analyzing the relationships among academic programs in the context of the co-op job market. To do this, we propose and apply a novel graph-mining methodology. The input to our problem consists of student-job interview pairs, with each student labelled with his or her academic program. From this input, we build a weighted directed graph, which we refer to as a program graph, in which nodes correspond to academic programs and edge weights denote the percentage of jobs that interviewed at least one student from both programs. For example, a directed edge from the Computer Engineering program to the Electrical Engineering program with weight 0.36 means that of all the jobs that interviewed at least one Computer Engineering student, 36 percent of those jobs also interviewed at least one Electrical Engineering student. Thus, the larger the edge weight, the stronger the relationship and competition between particular programs. The output consists of various graph properties and analyses, particularly those which find nodes forming clusters or communities, nodes that are connected to few or many clusters, and nodes that are strongly connected to their immediate neighbours. As we will show, these properties have natural interpretations in terms of the relationships among academic programs and competition for co-op jobs.

We applied the proposed methodology on one term of co-op interview data from a large Canadian university. We obtained interesting new insights that have not been reported in prior work. These insights can be beneficial to students, employers and academic institutions. Characterizing closely connected programs can help employers broaden their search for qualified students and can help students select programs of study that better correspond to their desired career. Students seeking a multi-disciplinary education can choose programs that are connected to other programs from many different clusters. Additionally, institutions can attend to programs that are strongly connected to (and face competition from) other programs by attracting more employers offering jobs in this area.

Acknowledgements

I would like to thank my supervisor, Professor Lukasz Golab, for his guidance throughout my undergraduate and graduate studies. I would also like to thank Professor Ken McKay and Professor Mark Smucker for spending time reviewing my thesis.

I must also extend my gratitude to Grant O'Neill, who provided the data used in this thesis, and Dana Evans Laity, who introduced me to Grant. This thesis was not possible without their help and kindness.

Finally, I need to thank my friends who proofread my thesis. They provided valuable comments and suggestions to improve the quality of my thesis.

Dedication

I would like to dedicate this thesis to my family and friends. I would not be able to come this far without your support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

According to the World Association for Cooperative & Work-integrated Education, 275 institutions from 37 countries have implemented cooperative education (co-op) programs [51]. The co-op students can work as full-time workers at a company for a period of time during their undergraduate education. Co-op experiences are vital because they supplements students' classroom skills and help them to gain practical experiences at different workplaces. In addition, students have opportunities to enrich their resumes with work experiences that help them kick start their careers early on [80, 90, 111].

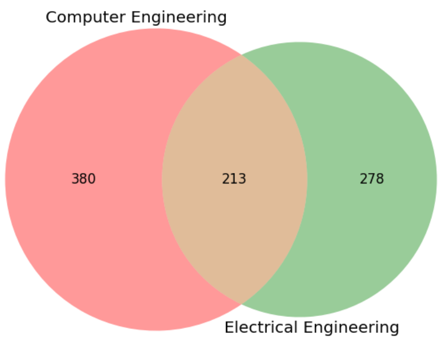
Many researchers have analyzed co-op education from various perspectives because of its growing popularity [51, 121]. Various papers identified factors and skills that are needed for a successful co-op term [30, 32, 48, 64, 100, 108, 122]. Another main area of research is on the effect of co-op education. Blair et al. [13] and Raelin et al. [98] studied the impact of co-op education on academic achievement and development of soft skills. Other researchers examined the design of co-op systems and recommended areas of improvements, such as communications and training [26, 25, 31, 62, 116, 101, 114].

The majority of related work qualitatively or statistically analyzed co-op education through survey data with fewer than 100 entries. Acquiring large datasets is challenging since the data are usually private and used exclusively for specific institutions or companies. To the best of our knowledge, the first research work that used large-scale data was our previous work [67]. We used a data-driven approach to analyze the satisfaction of co-op education using three years' worth of evaluation data gathered by a large Canadian educational institution. The data includes 36,615 evaluation pairs from 19,093 placements with 4,709 unique employers. The data was entered directly into the institution's co-op system by students and employers at the end of their work terms. We found that students received better evaluations in their senior years, but they rated their first employer the highest. We also found that senior students outperformed junior students in work placements abroad, and extended work terms at the same employer (spanning more than one academic term) did not increase student satisfaction.

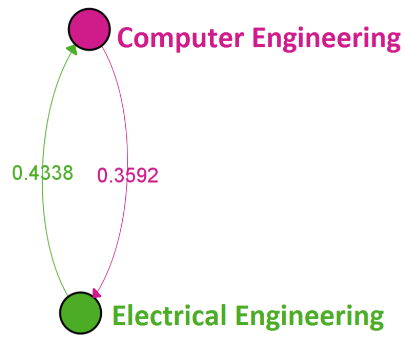
In this thesis, we make further contributions to the emerging research field of educational data mining and data-driven analysis of co-op education in particular. The objective of this thesis is to improve the co-op process by characterizing the relationships and extent of competition for co-op placements among students from various academic programs.

Our motivation for studying relationships among academic programs is threefold. First, with academic institutions introducing new programs in recent years [44, 118], it is often unclear how one program differs from another. This means that employers may not know exactly which programs to advertise their jobs to and students may not realize that they could be qualified for a job targeted to a different but related program (e.g., Computer Science and Software Engineering). Understanding similarities among programs can lead to more effective job and academic classification schemes and therefore can help match up job opportunities with qualified students. This analysis can also help students choose programs of study that better correspond to their desired careers. Second, data from the co-op system may be used to confirm which academic programs are multi-disciplinary and enable their students to obtain jobs from various categories. This issue is becoming increasingly important given the recent rise in popularity of multi-disciplinary and well-rounded education [9, 10, 18, 71, 119]. Third, academic departments often include examples of jobs secured by their graduates in promotional brochures and on departmental websites. Analyzing co-op job data can reveal job types that are exclusive to particular departments, and, conversely, departments whose students tend to compete for jobs with students from other departments. The university can then choose to attract more employers that offer jobs to programs facing strong competition. Thus, the problem we study in this thesis is critical to co-operative education from the student's, employer's and institution's perspective.

While some of the above questions have been raised in prior work on co-operative eduction, to the best of our knowledge this thesis is the first to propose techniques for answering them. The technical contribution of this thesis is a data-driven methodology for analyzing the relationships among academic programs using job interview data. The input to our problem consists of student-job interview pairs, with each student labelled with his or her academic program. The insight behind our methodology is to transform this input to a graph, which we refer to as a program graph, in which nodes correspond to academic programs and edge weights denote the percentage of jobs that interviewed at least one student from both programs. We provide an example in Figure 1.1. In this example, students from the Computer Engineering program interviewed for 593 distinct jobs, while students from the Electrical Engineering program interviewed for 491 distinct jobs. 213 of these jobs were common. As a result, the directed edge from Computer Engineering to Electrical Engineering has a weight of 0.36, derived from $213/593 = 0.36$. This weight



(a) Venn diagram representation



(b) Graph representation

Figure 1.1: A graph example

denotes the fact that of all the jobs that interviewed at least one Computer Engineering student, 36 percent of those jobs also interviewed at least one Electrical Engineering student. The same idea applies to the directed edge from Electrical Engineering to Computer Engineering, which is 0.43. Thus, the larger the edge weight, the stronger the relationship and competition between two programs.

After the above transformation, we proceed by performing standard graph analyses. As in other graph mining literature, we are interested in nodes forming clusters or communities, nodes that are connected to few or many such clusters, and nodes that are strongly connected to their immediate neighbours. This reveals exactly the information we need to understand the types of relationship occurring among academic programs in the co-op context. Graph clustering and community detection determine groups of related programs whose students interview for the same types of jobs; programs with connections to multiple clusters are likely to be multidisciplinary; and programs with high (weighted) in-degree and out-degree are the ones whose students face competition from students in other programs.

We applied the proposed methodology on one term of co-op interview data from a large Canadian university, consisting of 16,855 student-job interview pairs, with each student labelled with his or her academic program and each job labelled with its employer, job title, and targeted programs. We report several interesting findings enabled by our methodology.

First, the academic structure of the university does not always align well with the groups of closely-connected programs in our graph. For instance, some programs offered by the same department such as Electrical Engineering and Computer Engineering (both part of the Department of Electrical and Computer Engineering) were more weakly connected than programs from separate departments or even separate faculties such as Economics (Faculty of Arts) and Professional Risk Management (Faculty of Mathematics). On the other hand, the clusters we obtained naturally lead to intuitive job categories and academic specializations. Second, some programs that one expects to be focused, such as Psychology, turned out to be multi-disciplinary in the co-op context. These programs include students who obtained interviews for various types of jobs. On the other hand, joint programs were not necessarily multi-disciplinary: for instance, students from the Business & Math program interviewed mostly for business-oriented jobs and rarely for math or statistics jobs. Lastly, 29 out of 93 academic programs, among them Environmental Sciences/Ecology, had no jobs that interviewed only their students and no students from other programs. These programs may face the strongest competition.

We also created a separate program graph using only the interviews obtained by senior students. The clusters obtained from this graph were similar, but the nodes in this graph generally had lower fan-out than those in the graph including interviews of all students. This means that an employer offering a senior-level job tends to interview (senior) students from fewer distinct departments than a junior-level job, which makes sense since senior students are expected to possess specific skills that are not found in students from many other programs.

To recap, the two contributions of this thesis are 1) a novel graph-mining methodology for understanding relationships and competition among academic programs in the co-op job market, and 2) a case study using a large data set from a large Canadian university which showcases the utility of the proposed methodology. To the best of our knowledge, this is the first work to apply graph mining to co-op data and the first work to analyze the relationships among academic programs in the co-op context.

The remainder of this thesis is organized as follows. Chapter 2 discusses related work; Chapter 3 describes our data and methodology; Chapter 4 describes the experimental results; Chapter 5 concludes the thesis with directions for future work.

Chapter 2

Related Work

Since this thesis combines the fields of co-operative education and graph mining, we discuss related work from these two domains. We do not make any new algorithmic contributions in graph mining; instead, our contribution is to link graph properties and analyses with insight about the co-op system. In terms of co-op education, our methodology is the first to address the relationships among academic programs in the co-op context and enables new insight that, to the best of our knowledge, has not appeared before.

2.1 Co-op Education

Studies in the co-op education field can often be characterized by their primary intended audience. The studies are mainly geared toward students, employers, or educational institutions.

For students, many previous studies focused on identifying factors or skills that determine the success of the co-op experience through survey data [30, 32, 48, 64, 100, 108, 122]. For instance, leadership is found to be important to co-op education. Others studied the effect of co-op education on soft skills and academic achievements [13, 98] and found that co-op education have positive effects on academic marks. Furthermore, Gault et al. used survey data to validate that co-op plays a vital role in career success [54]. The importance of a multi-disciplinary background was also highlighted given the diverse responsibilities of professional roles such as engineers [18, 71]. Furthermore, it was found that a multi-disciplinary background can help students to deal with unknown and complex challenges in the future [9, 10, 119]. While the importance of multi-disciplinary education has been recognized, to the best of our knowledge our work is the first to propose a methodology for potentially verifying whether students from a particular academic program do indeed qualify for jobs from multiple disciplines.

Our previous work was the first to use a large dataset to assess the students' satisfaction of employers and employers' satisfaction of students [67]. The data included 36,615 evaluation pairs from 19,093 placements with 4,709 unique employers. We found that students received better evaluations in their senior years, but they rated their first employer the highest. We also found that senior students were more successful than junior students in work placements abroad, and extended work terms at the same employer did not increase student satisfaction. In this thesis, we target a different problem of understanding the relationships among academic programs in the co-op context.

From an employer's perspective, several studies analyzed how employers can benefit from having co-op students [20, 21, 22]. Molestane analyzed employers' expectations in the hospitality field and found that employers require more resources from educational institutions [84]. In terms of the employment process, many researchers analyzed how employers can attract more applications. Arachchige and Robertson identified determinants of employer branding based on a survey data of one academic faculty [5]. Leung analyzed 127 job descriptions and determined whether certain job description components attract a greater number of applicants [79]. Collins and other researchers found that early knowledge of employers' reputation and image are highly related to application intentions and decisions [24, 33, 34]. Furthermore, Hesketh found that employers have trouble advertising to specific academic programs and instead they advertise based on desired skillsets [63]. In this thesis, we will help employers to better advertise their jobs by finding clusters in the program graph that correspond to similar job categories that require similar skillsets.

Some studies whose primary intended audience are educational institutions [25, 26, 61, 62, 114, 116, 101] made recommendations to improve the co-op system and experience. These studies found that educational institutions should provide sufficient personnel and encourage employers to provide deep learning opportunities for students. They also recommended that recent graduate students should provide guidance to current co-op students. Some studies assessed the effectiveness of the design of co-op systems, such as the setup of coordinators [31, 41]. Additionally, it was suggested that academic programs can be evaluated based on their students' ability to obtain jobs [46, 117], which is a question that can be answered with the help of our methodology. Finally, Wilson and other researchers urged traditional academic disciplines to be updated so that the new disciplines can better reflect reality [44, 118]. As we will show, clustering our program graph can help determine groups of similar and/or popular disciplines based on the co-op job market.

2.2 Graph Analytics

Graph analytics, also known as network analysis, is a well-studied field. A graph can be generally described by the number of connected components, diameter, density, average shortest paths, and global clustering coefficient [16, 82]. A graph is *connected* if there is a path from any vertex to another vertex. A disconnected graph consists of several *connected components*, which are maximal connected subgraphs. *Diameter* describes the maximum length of all shortest paths, also called geodesics, between a node and any other node. *Density* in a graph compares the actual number of edges to the maximal possible number of edges. The average path length measures the average of all geodesics in a graph. This is the metric used in the famous phenomenon of "six degrees of separation" in many social graphs [6, 45, 58, 85, 115]. The global clustering coefficient is a measure of the degree to which nodes in a graph are inter-connected and based on triplets of nodes [94].

In addition to computing general graph properties, it is useful to identify important vertices (we will use the terms vertex and node interchangeably) [15, 17]. For this, there are four types of vertex *centrality* measurements. *Degree centrality* measures the number of ties that a node has. In-degree, also called fan-in, counts the number of edges directed to a node, while out-degree, also called fan-out, counts the number of outgoing edges from a node to its immediate neighbours. *Closeness centrality* of a node is defined by the reciprocal of the sum of its geodesic distances from all other nodes. *Betweenness centrality* quantifies the number of times a node acts as a bridge along the shortest path between two other nodes [53]. *Eigenvector centrality* measures the influence of a node in a graph by assigning scores to all nodes. A node gets a higher score if it is connected to high-scoring nodes, and vice versa. The PageRank algorithm is an example that uses this idea [95]. Centrality analysis has a wide range of applications such as identifying topics in Wikipedia [36] and popular researchers in scientific literature [58, 102]. In this thesis, we will use all of the above concepts except Eigenvector centrality.

Aggarwal and Wang [2] categorized common graph mining algorithms into pattern mining, classification, and clustering. Pattern mining algorithms are used to find frequently occurring subgraph patterns. We will not do subgraph pattern mining since we did not find any meaningful links between specific patterns and the questions we want to study. Classification algorithms can be used on graphs that contain missing information about nodes or edges. The algorithms learn from existing labels and predict labels for unknown nodes or edges. We will not do graph classification because our data are complete and we are not interesting in classifying new nodes based on their graph properties (rather, we are interested in describing the relationships among existing nodes). On the other hand, clustering is critical to our objective of finding related academic programs, i.e., those whose students interviewed for the same or similar jobs.

There are many graph clustering algorithms. Tang and Liu[109] categorize the algorithms into four types: node-centric, group-centric, network-centric, and hierarchy-centric.

- Node-centric algorithms enforces nodes within a cluster to satisfy certain properties such as mutuality, reachability or degree. Finding cliques and near-cliques is an example of this approach where the clusters must have a desired density [1, 50].
- For group-centric clustering algorithms, individual nodes are permitted to have low connectivity as long as the cluster as a whole satisfies a constraint such as high density [1, 42].
- Network-centric clustering algorithms allocate vertices into a number of disjoint sets. Max-flow min-cut algorithms assign nodes into clusters in a manner that minimizes the number of inter-cluster edges and maximizes the number of intra-cluster edges [4, 7, 49]. Vertex algorithms group nodes based on their structure. For example, Hopcroft et al. use Cosine Similarity to cluster nodes in the citation database [65], while Gibson et al. use Jaccard Similarity to group hosts on the World Wide Web [56]. Partitioning based on modularity is another type of clustering algorithm. This algorithm optimizes modularity, which measures the number of intra-cluster edges compared to that of a random graph [86, 88].
- Hierarchy-centric algorithms create hierarchical communities. For example, Newman and Girvan progressively remove edges that serve as connectors [89] (this means that certain nodes have to go through this particular path in order to reach another sets of nodes). The result of each iteration forms a layer in the hierarchy. Modularity can also be used to create hierarchical communities [29]. The algorithm continuously merges small communities until the modularity is maximized.

In this thesis, we use a group-centric and a network-centric clustering algorithm to identify a group of closely connected programs that have interviewed for the same jobs. The group-centric clustering algorithm we use considers any groups of three or more nodes, and creates a cluster if the density of the group satisfies a minimum value. For the network-centric clustering algorithm, we use the Louvain Modularity algorithm [14] to partition the academic programs into mutually exclusive clusters. This algorithm does not require a pre-specified number of clusters. Additionally, the resolution parameter of this algorithm enables us to compute a hierarchical structure of clusters. Furthermore, this algorithm is the only community detection algorithm that is implemented in the graph analytics software (Gephi [11]) used in this study.

Finally, we mention LinkedIn, a social networking service aimed at professional workers. A majority of analytics on LinkedIn data leverage connections among users [72, 81, 120] such as analyzing degrees of separation between users in the LinkedIn graph to understand the growth of professional network. Other data-driven studies analyzed users' skills [74] to recommend other similar skills that the users can add to their profiles, and examined the distribution of job titles [104] for different locations. To the best of our knowledge, no work has analyzed academic programs or educational background with respect to employment using LinkedIn data.

To the best of our knowledge, this study is the first to use graph analytics in the context of co-op education. We show that various graph properties are naturally related to competition and relationships among academic programs in the co-op job market.

Chapter 3

Data and Methodology

This chapter first provides background information regarding the co-op process in the university we are studying (Section 3.1) and subsequently an overview of our data set (Section 3.2). We then present our problem statement in Section 3.3 and our methodology in Section 3.4. We end with a discussion of the limitations of the proposed methodology in Section 3.5.

3.1 Background

The first part of this section describes the co-op employment process of the Canadian educational institution we are studying. In the second part of this section, we explain the layout of its academic programs.

3.1.1 Co-op Employment Process

An academic year runs from September to the following August, and includes three terms, each lasting four months. Students in the co-op system need to complete a minimum of five work terms out of six possible work terms over fourteen total terms during their undergraduate studies. In any given term, some students are taking courses on campus and others are on co-op work terms.

Figure 3.1 describes the co-op education employment process. At the beginning of each term, employers advertise job postings in a centralized on-line recruiting system. Each posting includes job information, targeted seniority (junior vs. senior students) and targeted academic programs.

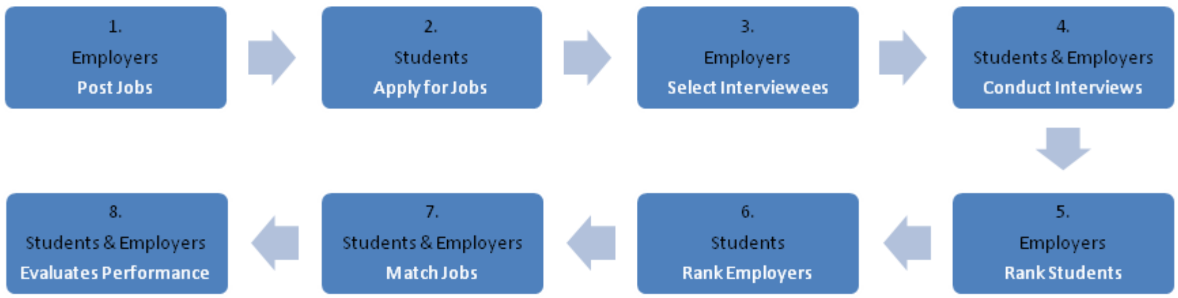


Figure 3.1: Co-op employment process

Students can view all this job information included in the postings. Although students are advised to focus their efforts on applying to jobs that target their program, students can apply to any jobs that are of potential interest to them. Employers then review the submitted applications and select students to interview. Once students are chosen, they must attend the scheduled interviews. Once interviews have been completed, employers and students rank each other from 1 (most interested) to 9 (least interested). Employers have the option to not rank any students, but students must rank the employers with whom they interviewed. The recruiting system uses these rankings to match students to jobs through minimization of the sum of the rankings from both the students and employers. Finally, at the end of every work term, employers and students evaluate each other.

3.1.2 Layout of Academic Programs

This educational institution has six faculties: Applied Health Science (AHS), Arts, Engineering, Environment, Mathematics, and Science. Each faculty is comprised of a number of programs, which varies across faculties. Here are some sample programs of each faculty: AHS contains the Recreation and Leisure Studies program, the Sociology and Economics programs are in the Arts faculty, Engineering has Computer Engineering and Nanotechnology Engineering (and other programs), Environment includes Planning and Geomatics, Computer Science and Actuarial Science are in the Mathematics faculty, and Faculty of Science offers Physics and Earth Science programs. There are 122 academic programs in total offered by this educational institution. Students from any one of these programs may enroll in co-op education. All Engineering programs and several programs from other faculties have mandatory co-op education.

We clarify some naming conventions that will be used in this thesis. A faculty or a program is capitalized when it is mentioned. For example, Computer Science refers to the Computer Science program in the Mathematics faculty, while computer science represents the study of the principles and use of computers. Furthermore, in the Mathematics faculty, there is the Mathematics program. This program is both its own area of study as well as a gateway to other specialized programs such as Actuarial Science. There are three cross-faculty programs: Computing and Financial Management (Arts and Mathematics), Software Engineering (Mathematics and Engineering), and Mathematical Physics (Science and Mathematics). For simplicity, we have chosen the latter faculty to be their "base faculty". We will discuss whether our choice correctly reflected the co-op employment situation.

3.2 Data Overview

We use co-op interview data from the Spring 2014 term, spanning from May to August. The data set contains information about students (program and academic year) and job postings (company name, job title, targeted programs and seniority). We also know which student applied for which job and which interviews he or she obtained.

During Spring 2014, there were 110 programs that had at least one student who interviewed for at least one job. Of the 110 programs, 17 programs only had one such student; we omitted these students and programs from our analysis. In the remaining 93 programs, there were 4,194 co-op students who obtained at least one interview, and 2,890 jobs that interviewed at least one student. In total there were 16,855 student-job interview pairs. On average, each job interviewed 5.83 students and each student had 4.02 interviews.

Figure 3.2 shows the number of students, applications, jobs (distinct jobs that interviewed at least one student), and interviews (sum of student-job interview pairs) by faculty. Figure 3.3 shows the number of programs, applications per student, interviews per student, and interviews per application by faculty. Arts and Mathematics have significantly more programs, 24 and 27 respectively. Faculties with more students had more applications and interviews. We normalize the application count and interview count by the number of students in each faculty. Even though the Engineering and Mathematics faculties had the highest application count per student, their success rate of obtaining an interview was the lowest. Their interview count per student was similar in magnitude to other faculties.

In terms of academic seniority, as shown in Figure 3.4 and 3.5, there were fewer fourth year students participating in the co-op process in Spring 2014 due to the scheduling of work terms. First year students had the most normalized and total number of applications, while fourth year students had the fewest. However, the success rate of obtaining an interview was higher as the academic year increased, shown in magenta bar in Figure 3.3. Even though third year students had the highest number of distinct jobs for which they interviewed and total interviews, the number of interviews per student across different academic years was similar, which is 4.02.

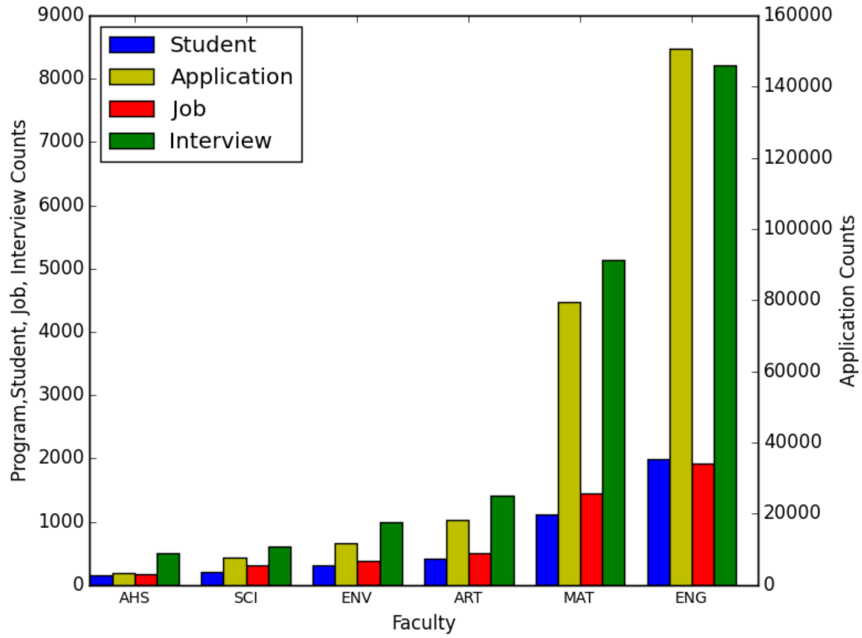


Figure 3.2: By faculty, student, application, job, and interview counts

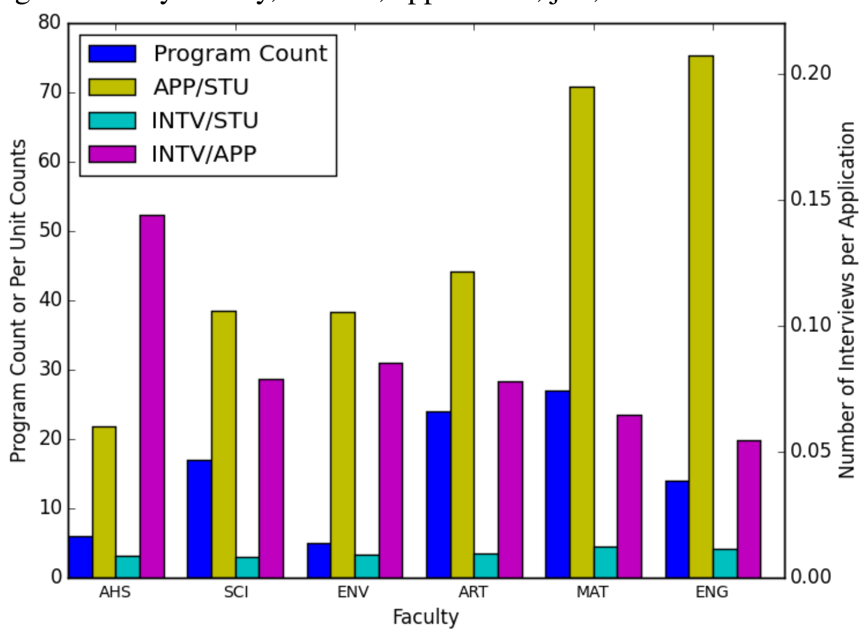


Figure 3.3: By faculty, the program count, applications per student (APP/STU), interviews per student (INTV/STU), and applications per interview (APP/INTV)

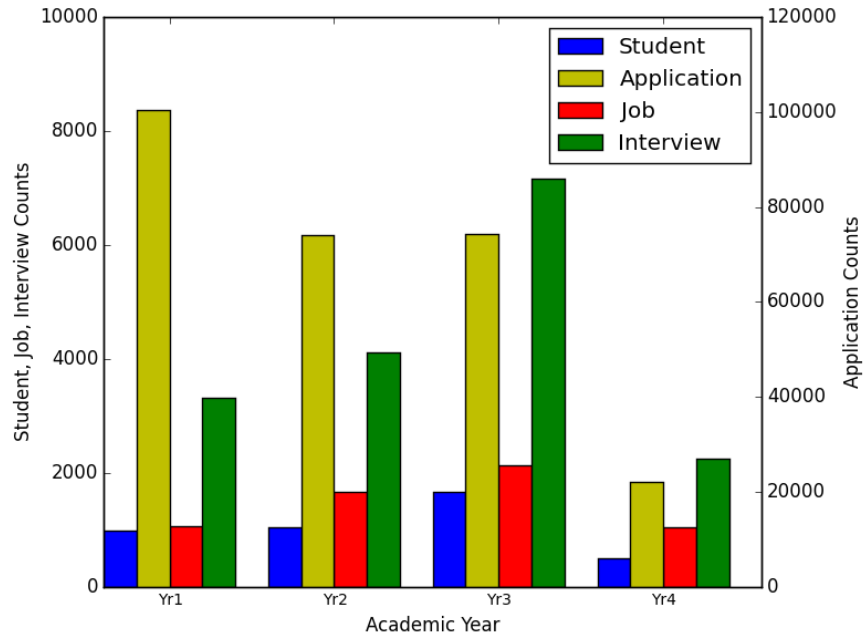


Figure 3.4: By academic year, student, application, job, and interview counts

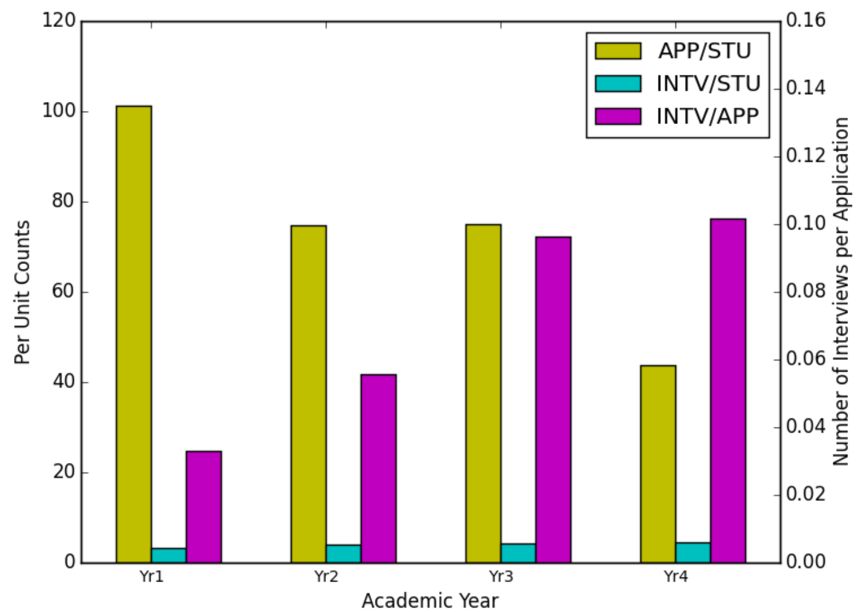


Figure 3.5: By academic year, the number of applications per student (APP/STU), interviews per student (INTV/STU), and applications per interview (APP/INTV)

3.3 Problem Statement

The goal of a co-op system is provide an adequate and suitable job pool and to match students with suitable jobs. In the institution we are studying, employers explicitly list the targeted academic programs in their job postings, and students search for jobs by the targeted academic program. We hypothesize that the co-op system can be improved with a better understanding of the relationships and competition among academic programs. In this thesis, we test this hypothesis by developing a methodology to facilitate the understanding of these relationships and the competition.

3.4 Methodology

We use data corresponding to student-job interview pairs, with each student labeled with his or her academic program and year, and each interview associated with a job ID, job title and job description. We propose a methodology that relies on transforming the student-job interview pairs to a graph $G = (V, E)$, more specifically a weighted directed graph with a set of vertices V and a set of edges E. Let e_{ij} be the weight of the edge (E_{ij}) from vertex v_i to v_j. The vertices correspond to academic programs. The directed edges represent relationships among programs, defined as the percentage of jobs that interviewed at least one student from both programs. Let J_i be a list of distinct jobs with whom students from program (vertex) v_i have interviewed. We define e_{ij} as the fraction of jobs in J_i that also appear in J_j:

$$e_{ij} = \frac{|J_i \cap J_j|}{|J_i|} \tag{3.1}$$

It can also be interpreted as a conditional probability, which is the probability that a job interviewed at least a student from program j given that this job interviewed at least a student from program i.

The direction of edges is important. For a given program node v_i, an incoming edge weight from another connected node v_j measures the fraction of J_j that also interviewed at least one student from v_i. Thus, a large incoming edge weight of v_i from v_j means that most job interviewing at least one student from v_j also interview at least one student from v_i. An outgoing edge weight (leaving v_i and entering v_j) represents the fraction of jobs that interviewed at least one student from v_i and at least one student from v_j. Thus, a large outgoing edge weight means that most jobs interviewing at least one student from v_i also interview a student from the other program.

We illustrate the edge weights with an example. Table 3.1 shows the interviews that nine students from three different programs (A, B and C) have obtained. There are four different jobs.

Table 3.1: Sample interview data

Student ID	Program Name	Job ID
1	A	1
2	C	2
3	B	1
3	B	2
4	B	1
5	A	2
6	A	3
7	C	2
8	C	4

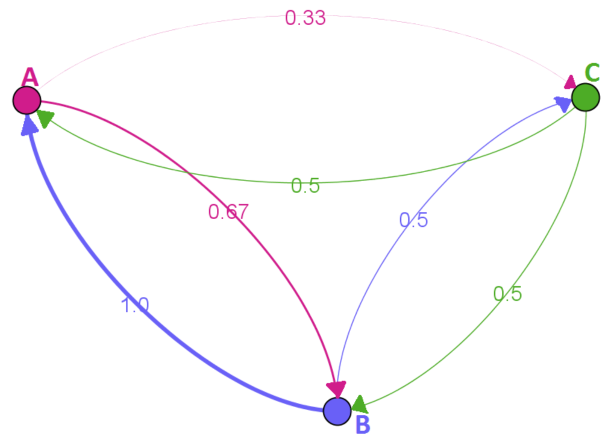


Figure 3.6: An example of a program graph

In this example, two students from program B (Student ID 3 and 4) interviewed for the same job (Job ID 1). Students in Program A interviewed for three distinct jobs, while students in programs B and C each interviewed for two distinct jobs. The lists of distinct jobs for each of the three programs are: $J_A = \{1, 2, 3\}$, $J_B = \{1, 2\}$, and $J_C = \{2, 4\}$.

The program graph based on this example is shown in Figure 3.6, and the edges are colour-coded by the source node. The edge weight from Program A to Program B is $|\{1, 2\}|/|\{1, 2, 3\}| = 2/3 = 0.67$, representing 67 percent of jobs that interviewed at least one student from Program A also interviewed at least one student from Program B. The edge weight from Program B to Program A is $|\{1, 2\}|/|\{1, 2\}| = 2/2 = 1$, meaning 100 percent of jobs that a student from Program B interviewed with also interviewed some students in Program A. Thus, the larger the edge weight, the stronger the relationship and competition between two programs.

Note that our definition of edge weights assumes that a relationship between two programs exists if at least one student from each program *interviewed* for the same job; furthermore, if there are many such jobs, then the edge weight will be larger. Thus, we are assuming that if a student obtains an interview for some job, he or she is qualified for it. Our data set does include information about which job a student ultimately obtained, but building a program graph based on job placements does not make sense: since most jobs only hire one student, there would be very few or no edges among programs.

Our data set could be used to create other graphs: rather than programs, the nodes could be jobs or students. Since our goal is to analyze relationships across programs, we use programs as nodes. Analyzing the other graphs is an interesting direction for future work.

Having explained how our program graph is constructed, we now clarify how its properties are related to the types and extent of relationships among academic programs in the context of co-op jobs:

- **Clusters:** Clusters in a graph represents a set of closely connected nodes. In the context of the co-op job market, clusters represent a group of related programs whose students interview for similar jobs.
- **Outliers:** Given a particular graph clustering, we define outliers as nodes that have strong connections to other nodes from multiple clusters (as opposed to "normal" nodes whose connections are mostly to other nodes within the same cluster). In our analysis, outliers correspond to multi-disciplinary programs: students from those programs have interviews in common with students from several different program clusters.
- **Fan-out:** (Weighted) fan-out measures the (weighted) number of outgoing edges of a node in a graph. In our context, weighted fan-out corresponds to the competition that a program faces from other programs. High weighted fan-out means that most jobs interviewing at least one student from the given program also interviewed students from other programs. As we will explain shortly, we use a slightly modified version of standard weighted fan-out that takes into account the fact that our edge weights are defined in terms of set intersections (of the job sets of different programs).

In the remainder of this section, we describe the graph algorithms that we use in this thesis to identify program clusters, multi-disciplinary programs and programs facing strong competition. To carry out the analysis, we use Gephi [11] and Python with the Networkx package [60], which implement the algorithms we need. Table 3.2 summarizes the graph properties that are relevant to the algorithms we are using, where N_v is the neighbourhood of a vertex v (i.e., the set of nodes directly connected to v) and $d(u, v)$ is the length of the shortest path between vertices u and v. Note that our program graph cannot have an incoming edge without a corresponding outgoing edge or vice-versa. There is no edge if two programs have no jobs in common, but there is always an incoming and outgoing edge if they have at least one job in common.

3.4.1 Finding Similar Programs

We search for groups of similar programs using two algorithms: finding near-cliques and community detection.

Near-Cliques

A clique is a group of nodes that are fully connected, i.e., it has a density of one. A near-clique is a group of nodes whose subgraph consisting of them and their edges has a density of nearly one,

Table 3.2: Definitions of Relevant Graph Metrics

Terminology	**Definition**	**Formula**
Graph-based Metric		
Connected Component	Maximal connected subgraph	
Density	Number of edges as a fraction of the maximal possible number of edges	$\frac{\lvert E\rvert}{\lvert V\rvert * (\lvert V\rvert - 1)}$
Diameter	Maximum length of all shortest paths between a node and any other node	$max_{u,v}\{d(u,v)\}$
Average path length	Average length of all shortest paths	$avg_{u,v}\{d(u,v)\}$
Global Clustering Coefficient	Percentage of closed triplets out of the total number of triplets, where a triplet is any three nodes in a graph	$\frac{\sum_{\text{Closed triplets}}(\sum_{u \neq v} e_{uv})}{\sum_{triplets}(\sum_{u \neq v} e_{uv})}$
Node-based Metric		
Degree Centrality	Number of ties that a node has (in-degree+out-degree). Since the number of in-coming edges equals the number of out-going edges in our program graph, in-degree equals out-degree.	$2 * \lvert N_v\rvert$
Closeness Centrality	Reciprocal of the sum of the shortest paths from a given node u to all other nodes.	$\frac{1}{\sum_{u \neq v} d(u,v)}$
Betweenness Centrality	Number of times a node acts as a bridge along the shortest path between two other nodes	$\sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(v)$ is the number of those paths that go through v.

i.e., a group of nodes that is nearly fully connected. We define a near-clique as a group of nodes having a density of at least 0.8. However, since our program graph is weighted, we want to find near-cliques with large edge weights. To do this, we first remove all edges from our program graph except the five percent with the largest edge weights. The resulting graph may leave some nodes disconnected, while other pairs of nodes may only have an incoming or an outgoing edge.

We then make one more change to the resulting graph: we remove edge directions and simply retain an edge between two programs if there is either an incoming or an outgoing edge. Finally, we apply a near-clique finding algorithm (see Figure A.2 of Appendix A for the Python source code) to the remaining graph. We only output maximal near-cliques (by removing those that are subsets of larger ones), and only those of size at least 3.

Community Detection

A community/cluster is a group of vertices that are densely connected with one another, but sparsely connected with other communities. We use an unsupervised algorithm, Louvain Modularity [14], which is implemented in Gephi.

The goal of this algorithm is to maximize a modularity metric, Q, which compares the discovered communities with random connectedness [86]. Newman [87] introduced modularity for weighted undirected graphs. We translate this metric to fit our weighted directed graphs as follows. Let c_i be the community that a node i belongs to, and e_{ij} be the edge weight between node i and j. The fraction of total edges that is within clusters is shown in Equation 3.2.

$$\frac{\sum_{ij} e_{ij}\delta(c_i, c_j)}{\sum_{ij} e_{ij}} = \frac{1}{m}\sum_{j} e_{ij}\delta(c_i, c_j) \tag{3.2}$$

where the function $\delta(c_i, c_j)$ is equal to 1 if $c_i = c_j$ (i.e. node i and j belong to the same cluster) and 0 otherwise. Let $m = \sum_{ij} e_{ij}$ and $k_i = \sum_j e_{ij}$ (i.e., the sum of the weights of the edges that connect to node i). For a graph where the degrees of vertices are the same and edges are randomly connected, the probability of an edge existing between node i and j is $\frac{k_i k_j}{2m}$. As a result, modularity is calculated as

$$Q = \frac{1}{m}\sum_{i,j}(e_{ij} - \frac{k_i k_j}{m})\delta(c_i, c_j) \tag{3.3}$$

$Q = 0$ means that the community detection result is no better than random. The maximum value for Q is 1. Higher modularity indicates a better partition result.

The Louvain Modularity method [14] is iterative and includes two phases. In the first phase, each node starts in a different community. Then, for each node i, we compute the gain in modularity if i is moved to the community that its neighbour (j) belongs to. If the gain is positive, the change happens; otherwise i remains in its original community. This process is repeated iteratively and sequentially until no further improvements can be made. The outcome of the first phase is only a local optimum of modularity since the order of processing of the nodes will affect the result. In the second phase, a new graph is created such that nodes are communities in the results from the first phase, and edge weights are the sum of edge weights between nodes in the two communities. We reapply the process in the first phase on this new graph. The algorithm stops when the maximum modularity is reached. To account for the effect of order, we run this algorithm multiple times and keep the result with the highest modularity.

One drawback of [14] is that it avoids creating small clusters. Lambiotte et al. [77] adds a resolution parameter t to account for this disadvantage. The new modularity definition is shown in Equation 3.4. The default t value is 1; smaller values of t lead to more and smaller communities. We will try different values of t in steps of 0.1.

$$Q_{new}(t) = (1-t) + \frac{1}{m}\sum_{i,j}(e_{ij}t - \frac{k_i k_j}{m})\delta(c_i, c_j) \tag{3.4}$$

The clusters we obtain correspond to programs whose students interview for similar jobs. To describe a cluster in terms of the types of jobs it represents, we will select keywords that frequently appear in the corresponding job titles.

3.4.2 Finding Multi-Disciplinary Programs

To find multi-disciplinary programs, we leverage the clusters obtained via the community detection algorithm described in the previous section. Intuitively, if an academic program has strong connections to other programs from multiple clusters (each of which corresponds to different types of jobs), it may be multi-disciplinary.

For each program, we calculate a multi-disciplinarity measure as follows. Let i be the number of communities/clusters we have discovered. For each community i, let p_i be the fraction of the total weight of the outgoing edges from the given program to the programs only in i. Then, the we compute the entropy of the distribution of edge weights among different communities simply as $\sum_i -p_i \log_2 p_i$. High entropy means that the given program has strong links to programs in multiple clusters and therefore may be multi-disciplinary. The Python code for this process is provided in Figure A.3 of Appendix A.

We illustrate our notion of entropy with an example using the Medicinal Chemistry program. Students in Medicinal Chemistry had interviews in common with eight other programs from four clusters, call them red, blue, purple, and green, as shown in Figure 3.7 where nodes are colour-coded by their clusters. Only out-going edges of Medicinal Chemistry are relevant, since they represent the percentage of jobs from $J_{MedicinalChemistry}$ that also interviews students from

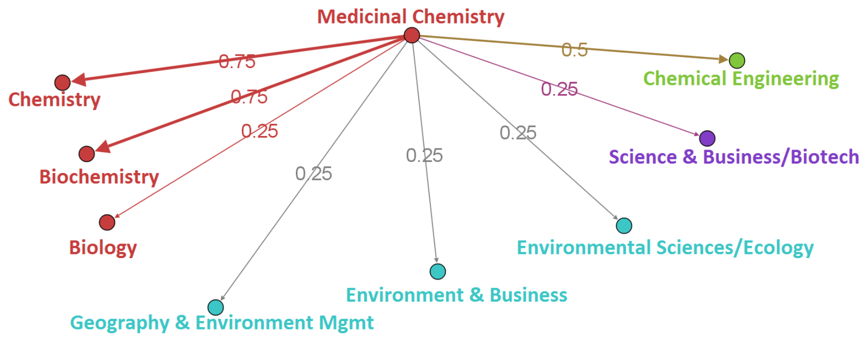


Figure 3.7: Direct competitors of Medicinal Chemistry, colour-coded by clusters

its direct competitors. The sum of all out-going edge weights of Medicinal Chemistry is 3.25. $p_{red} = (\sum_{\text{i=competitors in red}} e_{MedicinalChemistry,i})/3.25 = (0.75 + 0.75 + 0.25)/3.25 = 0.54$, which is the sum of weights of edges from Medicinal Chemistry to the programs in the red cluster. Similarly, $p_{blue} = 0.23$, $p_{green} = 0.15$, and $p_{purple} = 0.08$. Thus, $entropy_{MedicinalChemistry} = -p_{red}log_2p_{red} - p_{blue}log_2p_{blue} - p_{purple}log_2p_{purple} - p_{green}log_2p_{green} = 1.67$

3.4.3 Finding Competing Programs

We define the extent of competition that a program faces using a "set fan-out" metric. We want to measure the percentage of jobs that interviewed students from the given program which also interviewed at least one student from another program. For a given node (program) i, we define:

$$\text{Set Fan Out}_i = \frac{|\cup_{j \neq i} (J_i \cap J_j)|}{|J_i|} \tag{3.5}$$

A set fan-out of zero means that all the jobs that interviewed at least one student from program i only interviewed students from i and no other program. Students from such a program may have specialized skills that students from other programs do not have. On the other hand, a set fan-out of one means that every job that interviewed at least one student from program i also interviewed at least one student from another program. In other words, there were no jobs that exclusively interviewed students from i and therefore students from i may be facing strong competition for jobs.

Returning to the example from Table 3.1, $J_A = \{1, 2, 3\}$, $J_B = \{1, 2\}$, and $J_C = \{2, 4\}$. For Program A, its set fan-out is $\frac{|(J_A \cap J_B) \cup (J_A \cap J_C)|}{|J_A|} = \frac{|\{1, 2\}|}{|\{1, 2, 3\}|} = \frac{2}{3} = 0.67$. It means that students from Program A competed with students from other programs in 67 percent of their jobs. 33 percent of jobs that interviewed students from Program A did not interview students from other programs. The set fan-out for Program B is 1 and for Program C is 0.5.

3.5 Limitations

Our results have three key limitations. First, our findings are based on interview results from Spring 2014 term only. Second, we do not know the rationale behind a student's decision to apply for certain jobs and the employers' criterion for student selection for interviews. We assume that obtaining an interview means that the student is potentially qualified for the position. Third, we are aware that there are various confounding factors to students' and employers' behaviour in the co-op process. For example, employers who are alumnus of the academic institution might behave differently than others who are not. We cannot verify all factors, but we will describe the findings concluded from our dataset.

Chapter 4

Results and Discussion

This chapter begins with an analysis of the full program graph consisting of all academic programs that had at least two students in the co-op system in Spring 2014 (Section 4.1). In Section 4.2, we study the effect of academic seniority on competition and relationships among academic programs by examining a separate program graph using only the interviews obtained from senior students, namely the senior program graph. We end this chapter with a further investigation of some of the interesting and unexpected findings from our program graph analysis (Section 4.3).

We would like to answer following hypotheses and questions in this chapter:

- The program graph has similar characteristics as social graphs, such as the Facebook user graph, studied in the literature.
- Senior students are more specialized and complete for jobs with students from fewer programs.
- The layout of academic programs should well reflect the relationships of academic programs in the co-op system.
- Cross-faculty programs or those with two areas in their names are more multi-disciplinary. For multi-disciplinary programs, do they consist of well-rounded individuals or sets of students with specialized skills?
- A majority of programs should have many jobs that only interview their students since a particular academic program should arm its students with specialized skills. For more specialized programs, whether their jobs that did not interview students from other programs are truly tailored to them? For less specialized programs, do the same direct competitors appear in every single job interview or do certain competitors cover a specific type of jobs?

4.1 Full Program Graph Analysis

We first analyze the graph properties to provide statistics of the full program graph in Section 4.1.1. We then study competition and relationships among programs from three perspectives: similarity of programs in Section 4.1.2; multi-disciplinary programs in Section 4.1.3; and programs facing strong competition in Section 4.1.4.

4.1.1 Full Program Graph Statistics

We illustrate the full program graph in Figure 4.1. It consists of 93 programs and 1,866 pairs of edges. The graph is organized using the Fruchterman Reingold layout in Gephi. Programs located at the edge do not have any particular meaning. The graph is not particularly readable due to the large number of nodes and edges. In the rest of the thesis, we may only show subgraphs of the program graph to enhance visibility. For example, a graph with 5% thickest edges refers

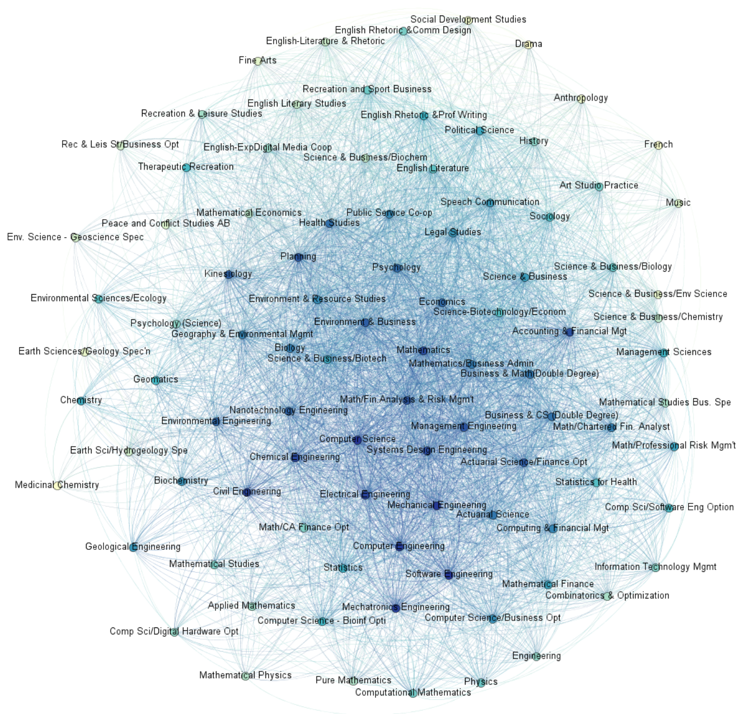


Figure 4.1: Full program graph

to a subgraph that includes edges in the top 5th percent of edge weights; a graph composed of cliques refers to a subgraph that includes nodes and edges participating in the cliques.

We hypothesized that the program graph has similar traits as social graphs analyzed frequently in the graph literature. For instance, 99.91 percent of Facebook users are inter-connected [113] and the Facebook graph has only 3.74 degrees of separation [6]. We found that our program graph is significantly more connected than the social graphs—with only 0.57 degrees of separation—as all academic programs are inter-connected. We also found that "broad" program such as Psychology and cross-faculty programs had links with the most other programs and served as bridge nodes in the program graph. Furthermore, we found that the Computer Science program in the Mathematics faculty was strongly connected with many programs in the Engineering faculty. Some programs in Science, Applied Health Sciences, and Arts faculties had strong interview overlap with each other as well.

Overall, the program graph is connected, meaning that any given program has paths which reach every other program. The diameter of the graph is 3, the average path length is 1.57 with a variance of 0.04, and the degree of separation among academic programs is 0.57. The density of the entire graph is 0.44, indicating that 44 percent of all possible pairs of programs had at least one interview in common. The global clustering coefficient is 0.64, indicating that 64 percent of triplets formed closed triplets. As a result, the program graph is much more connected compared to, say, the Facebook social graphs, whose degree of separation is 3.74 and 99.91 percent of its nodes are connected.

For individual nodes, we evaluate their importance by degree centrality, closeness centrality, and betweenness centrality.

Recall that degree centrality measures the number of ties that a node has. Since edges come in pairs in the program graph, in-degree is the same as out-degree. For simplicity, we only discuss in-degree in the rest of this chapter. The in-degree of the program graph ranges from 8 to 84, with an average of 40.13. This indicates that a program competes with an average of 40.13 other programs among its interviews. The programs with a small in-degree usually had a small number of jobs that interviewed students in these programs. For example, students in the Medical Chemistry program only interviewed for four distinct jobs. On the other hand, programs with the highest in-degree are cross-faculty programs such as Environment & Business and "broad" or "fundamental" programs such as Mathematics. It makes sense that cross-faculty programs lead to more diverse competition for jobs due to being multi-disciplinary. It is also logical that fundamental programs had interviews in common with many other programs.

Taking weights into account, Figure 4.2 shows that the *weighted in-degree* and *weighted out-degree* are not correlated. Weighted in-degree spans from 0.26 to 17.73 with an average value of 3.93, whereas weighted out-degree steadily decreases from 6.00 to 1.78 with an average value

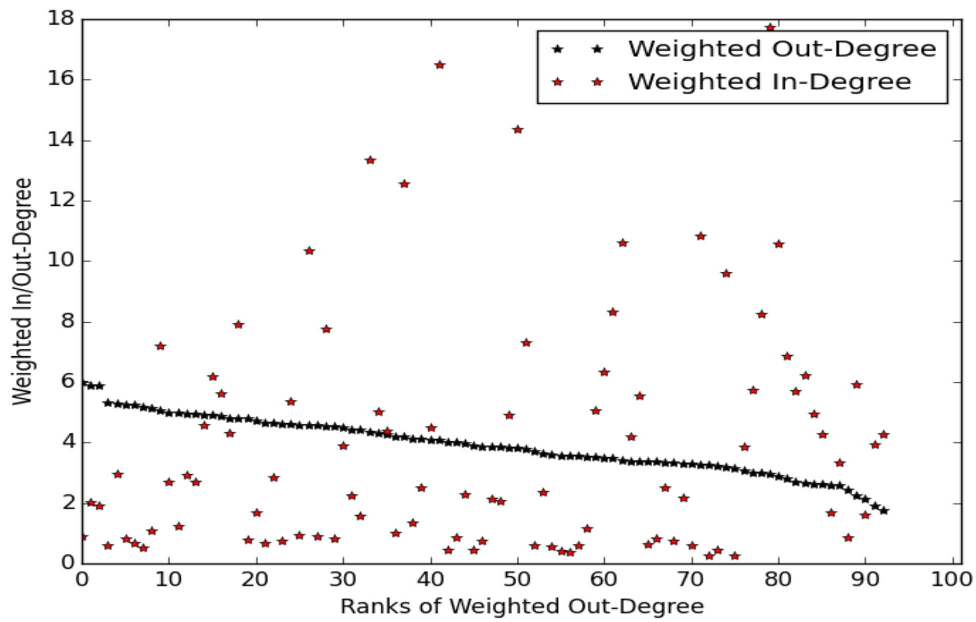


Figure 4.2: Weighted in-degree and out-degree (descending order) in the program graph

of 3.93. Recall that the in-coming edge weight of a program i from j is the fraction of J_j that were in common with J_i ($e_{ji} = \dfrac{|J_i \cap J_j|}{|J_j|}$), and the out-going edge weight of this program i to j is the fraction of J_i that were in common with J_j ($e_{ji} = \dfrac{|J_i \cap J_j|}{|J_i|}$). Figure 4.2 also shows that weighted in-degree has greater variability than weighted out-degree. The denominators of out-going edges in the weighted out-degree calculation are the same (J_i), while the denominators of in-coming edges in the weighted in-degree calculation are different (J_j where j corresponds to program i's neighbour). Since the number of jobs that interviewed students of other programs is more diverse, there was more variability observed in the weighted in-degree.

Closeness centrality measures the reachability of a node in the graph. Nodes with a higher closeness centrality value are closer to other nodes. In the context of co-op, it can indicate the potential of a program to compete with other programs. The closeness centrality of this graph ranges from 0.51 to 0.92, with an average of 0.65. Mathematics and Environment & Business again had the highest closeness centrality. Conversely, Medical Chemistry had the lowest closeness centrality. The majority of programs had the same ranking in closeness centrality and degree centrality.

Betweenness centrality indicates whether a program serves as a bridge that lies on a shortest path between many paris of nodes. It ranges from 0.11 to 347.11, with an average value of 52.26. The Mathematics and Environment & Business programs have significantly higher betweenness centrality (above 300) than others (under 200). The Mathematics faculty has 13 programs that are exclusively for students in second year or above. Prior to switching into these programs, students must stay in the general Mathematics program, implying that the Mathematics program might be an intermedium that connects with these 13 programs. The Environment & Business program could potentially connect programs with environmental, technical and business focuses.

The *edge weight* itself is an important measurement of direct competition. The 3,732 total edges in the program graph have weights ranging from 0.001 to 0.895, as shown in Figure 4.3. The edge weights first decrease sharply and then slowly smooth out. There are many programs with relative low edge weights, indicating low competition for jobs which interviewed students from those program.

Figure 4.4 illustrates the full program graph with 5% thickest edges. Programs are colour-coded by their academic faculty. The Computer Science program from the Mathematics faculty stands out in this graph. It has the highest degree centrality by a wide margin. Notably, it has many thick in-coming edges, which indicates that the Computer Science program is a strong competitor to its neighbours.

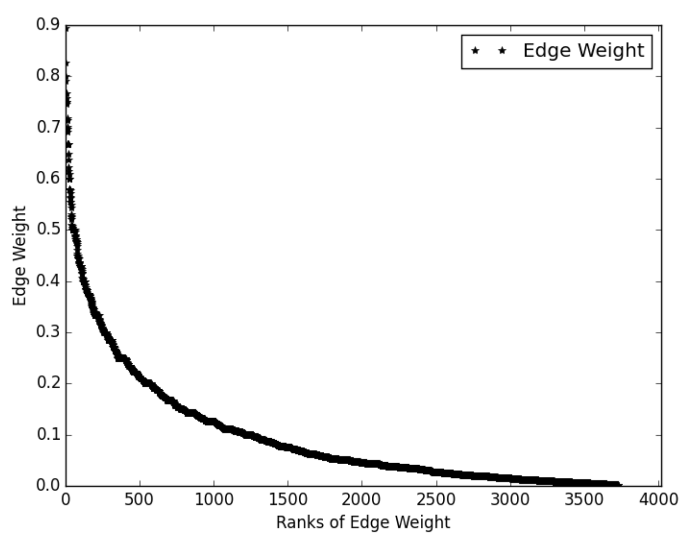


Figure 4.3: Edge weight in descending order

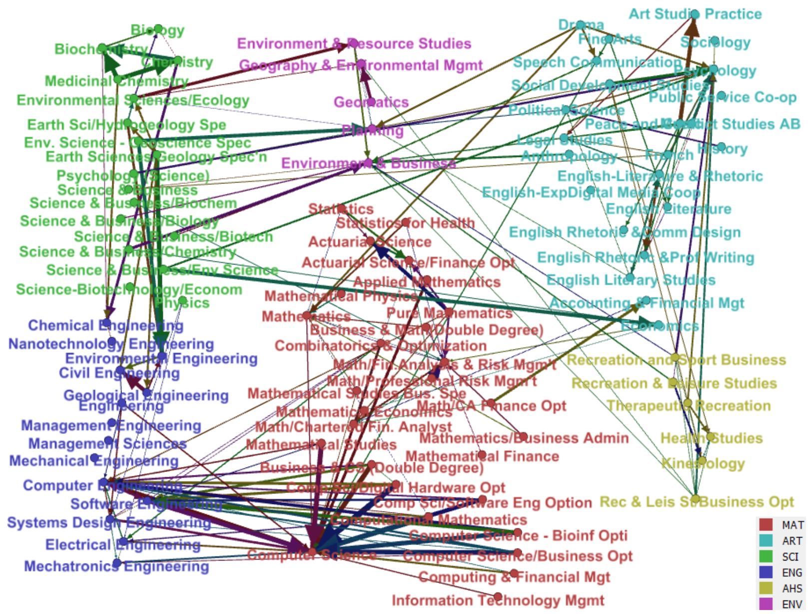


Figure 4.4: Program graph with edges in the top 5th percentile of edge weights, colour-coded by faculty

For inter-faculty relationships, many programs in the Mathematics faculty are related to the Computer Science program. They also had a close relationship with Software Engineering, Computer Engineering and other programs. Some programs in the Science faculty were connected with Civil, Environmental, Geological, and Chemical Engineering. Additionally, they were connected with Economics (in the Arts faculty), Psychology (in the Arts faculty), and Planning (in the Environment faculty). Surprisingly, the Applied Health Sciences (AHS) faculty had the strongest connections with programs in Environment and Arts.

For intra-faculty competitions, all programs in the Environment and AHS faculties were closely connected with other programs from their own faculty. On the other hand, the other faculties contained multiple connected components, and also some programs in these faculties, such as Mechanical Engineering in the Engineering faculty, did not have any edges in the top 5th percentile of edge weights.

4.1.2 Finding Similar Programs

The goal of this section is to produce clusters of programs that were strongly connected. The programs in the same clusters are similar in the context of co-op employment since their students interviewed for the same jobs. As explained earlier, we cluster programs using two methods: a group-centric algorithm and a community detection algorithm. The first part of this section discusses the near-cliques generated by the first algorithm. The second part analyzes the clusters created by the community detection algorithm.

We hypothesized that programs from different departments or different faculties should be in separate clusters, but we found some exceptions. For example, students in the Psychology program of the Arts faculty had many interviews in common with students from AHS programs. Furthermore, the Science & Business program from the Science faculty was closely related to the programs in the Arts faculty. We also hypothesized that Computer Science and Software Engineering are very similar, which turned out to be true since these two programs were always in the same cluster. However, we also found strong connections to Computer Science, Software Engineering and Computer Engineering from other clusters, indicating that many students from other programs also interviewed for software-related jobs. Notably, the majority of these students were junior: second year or below.

Near-Cliques

Recall that we define near-cliques to be the subgraphs with density no less than 0.8. In order to focus on significant connections, we analyze a subgraph of the program graph consisting only of the top 5 percent edges with the largest edge weights. There are 135 near-cliques in this filtered graph, listed in Appendix C.1. Specifically, there are 10 near-cliques of 7 programs, 82 near-cliques of 6 programs, 21 near-cliques of 5 programs, 18 near-cliques of 4 programs, and 4 near-cliques of 3 programs. Figure 4.5 plots these 135 near-cliques, which consist of 57 distinct programs and 202 distinct edges.

This graph is split into three connected components. The blue connected component is a clique involving chemical and biological programs. The red component is centred around Computer Science, Computer Engineering, and Software Engineering. However, the left part of this connected component is not connected to any of the computing programs; rather, the Financial Analysis and Risk Management (FARM) program is a bridge that connects the financial and computing parts of this component. In the green connected component, there are more such bridge programs. Psychology connects health related programs with Arts related programs. Planning and Environmental Science/Ecology connect Arts related programs with Environment related programs. Near-cliques in the green connected component are less intertwined compared to those in the red connected component.

While we expected Computer Science and Software Engineering to be part of a clique, we thought that Computer Engineering would be more strongly connected to Electrical Engineering

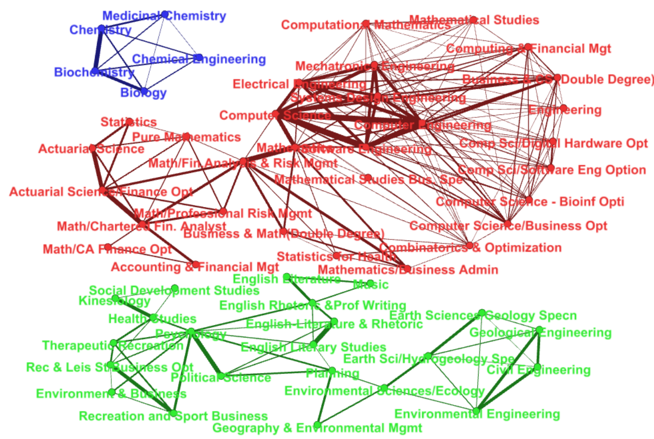


Figure 4.5: Graph composed of 135 near-cliques found in the full program graph, colour coded by connected components

(both of which are in the same department) than Computer Science. That was not the case: 92 out of 135 near-cliques contained Computer Science, Software Engineering, and Computer Engineering. 953 jobs interviewed students from at least two programs in these 92 cliques. Figure 4.6 shows a word cloud [37] corresponding to frequent keywords in the job titles from these cliques; we removed insignificant words such as "student" and "intern". Most of these jobs are in software development. Furthermore, 910 out of 953 of these jobs were advertised to

Figure 4.6: Word cloud of job titles of jobs that interviewed students from any two programs in the 92 cliques that containing Computer Science, Software Engineering, and Computer Engineering.

junior or intermediate students. 690 students who were not from Computer Science, Software Engineering, or Computer Engineering had also interviewed for these 953 jobs. 467 of them were students from second year or below, and only 14 students were in their fourth (final) year, revealing that junior students from non-computing departments also interview for software jobs.

Community Detection

Recall from Section 3.4.1 that we use the Louvain Modularity method to partition the graph into non-overlapping clusters. This algorithm maximizes the modularity for a given user-supplied resolution parameter. The number of clusters increases when the resolution parameter decreases.

Figure 4.7 summarizes the community detection results with the resolution parameter ranging from 1.5 to 0.1. As the resolution parameter decreases, the number of clusters created increases from 2 to 37 (# of Communities). As the average density of clusters increases (Avg Community Density), the percentage of intra-cluster edges decreases (Intra-Cluster Edge%) and the percentage of inter-cluster edges increases (Inter-Cluster Edge%). Even though the general trend of modularity is declining (Modularity), the resolution parameter greater than 1.2 has lower modularity than the subsequent results.

Ideally, the goal is to maximize modularity, percentage of intra-cluster edges, and average

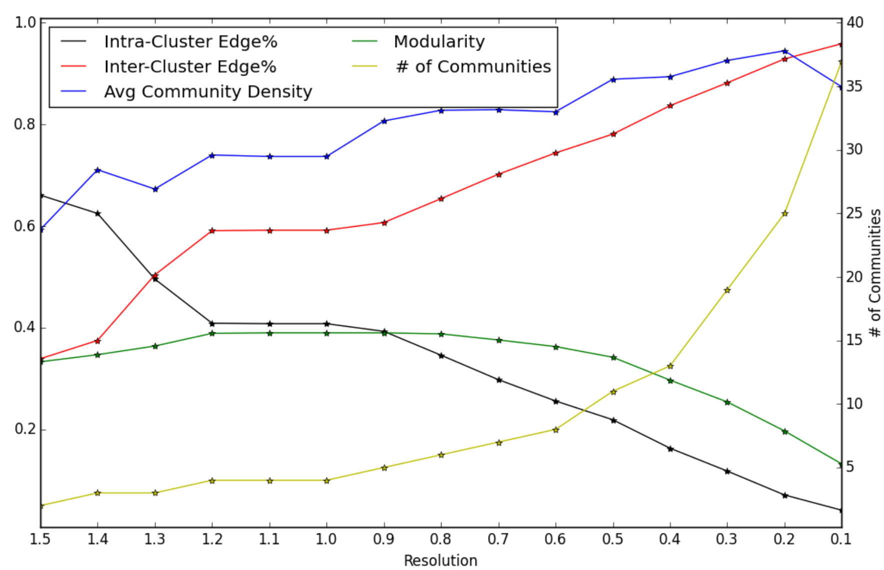


Figure 4.7: Community detection result

density of clusters, and minimize the percentage of inter-cluster edges. We study four community detection results in detail. The selected resolution parameters are 1.5 (2 clusters), 1.1 (4 clusters), 0.6 (8 clusters), and 0.1 (37 clusters). For visibility, the figures illustrating the communities (Figure 4.8, 4.9, 4.10, and 4.11) only contain the top 5 percent edges with the largest weights.

As the resolution parameter declines, large clusters are divided into smaller clusters, forming a hierarchy. Near the top of the hierarchy, the four clusters roughly (but not exactly) correspond to academic faculties in our institution. On the other hand, the 23 clusters at the bottom naturally correspond to different academic specializations and job categories, which can help employers target their jobs and can help students plan their academic careers.

We summarize the hierarchy corresponding to our community detection results in Figure 4.12.

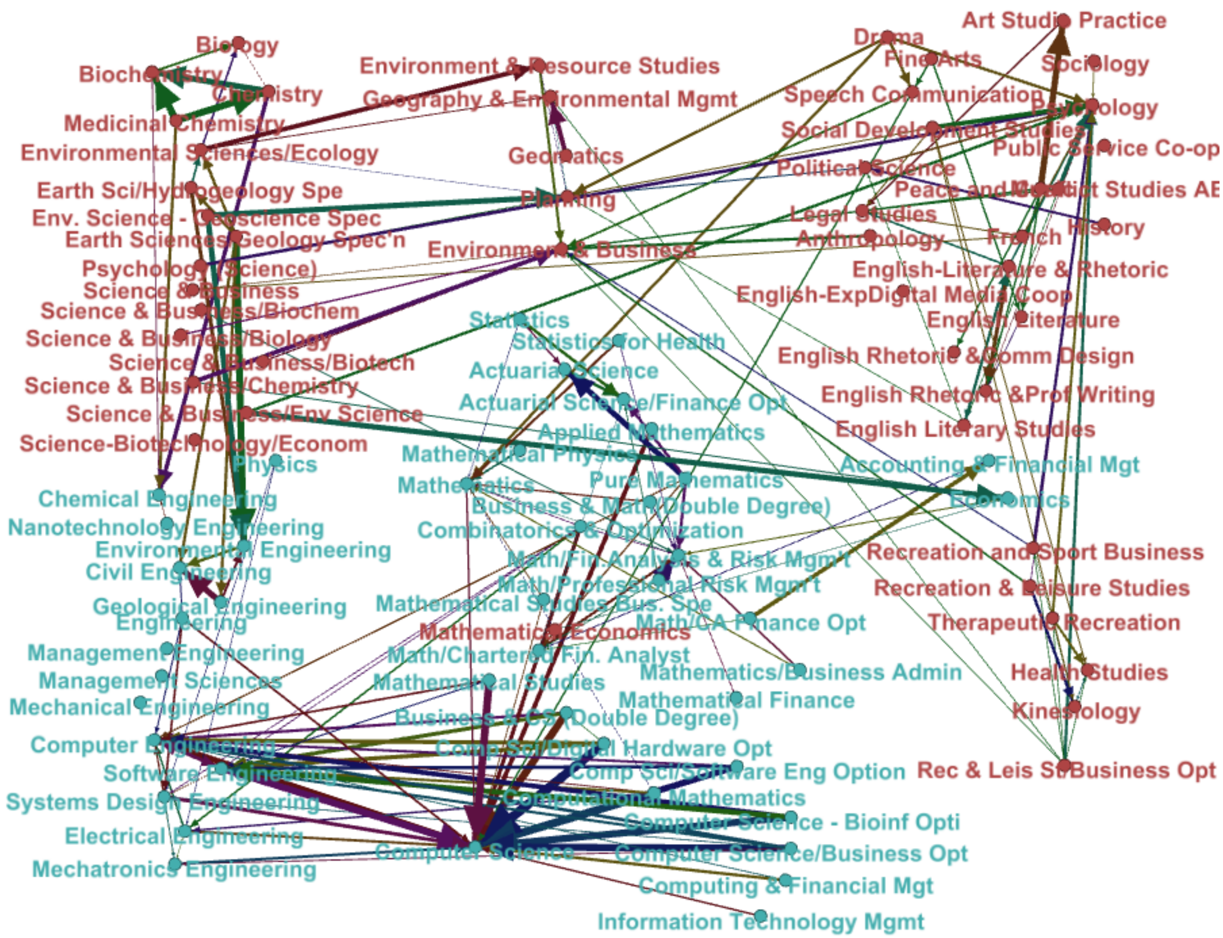


Figure 4.8: Full program graph with 5% thickest edges, colour-coded by clusters (2 clusters)

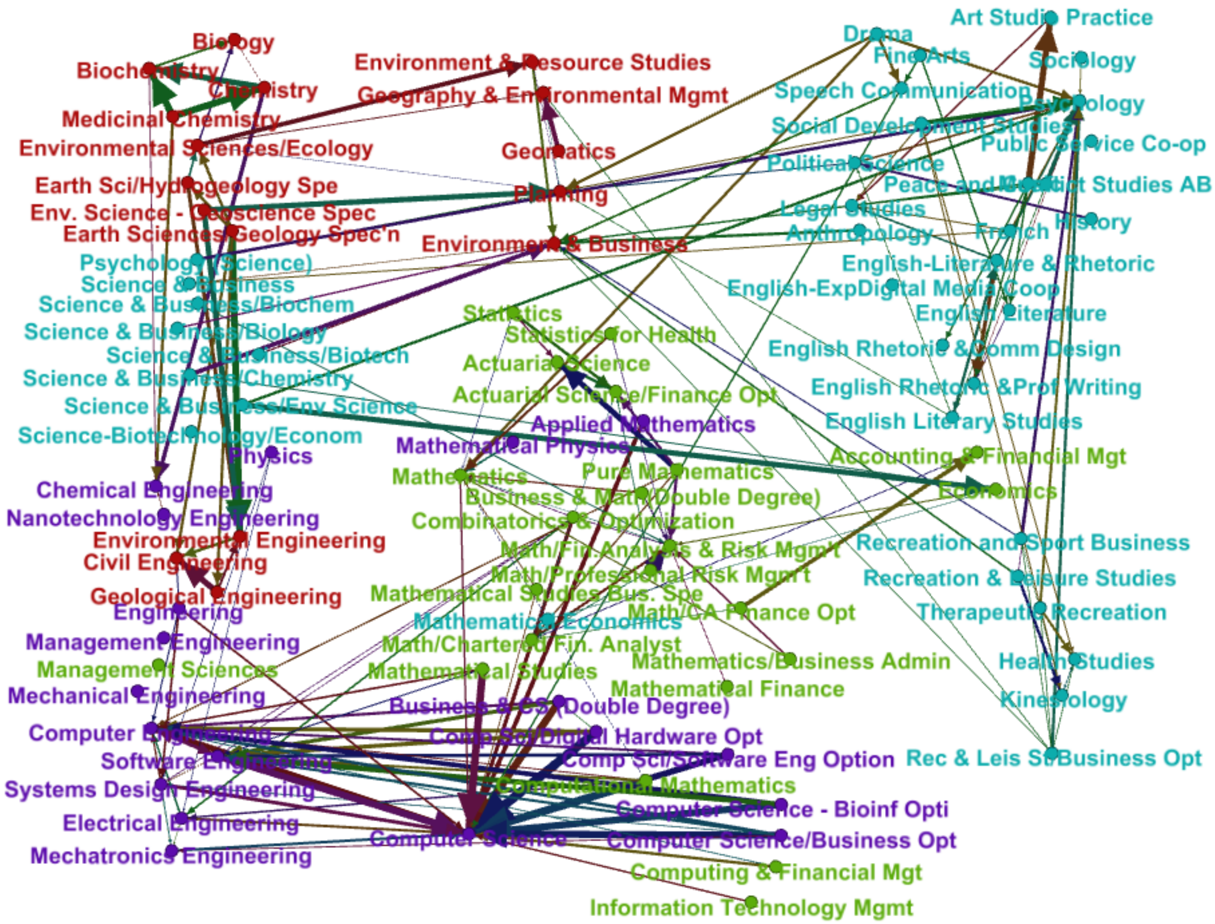


Figure 4.9: Full program graph with 5% thickest edges, colour-coded by clusters (4 clusters)

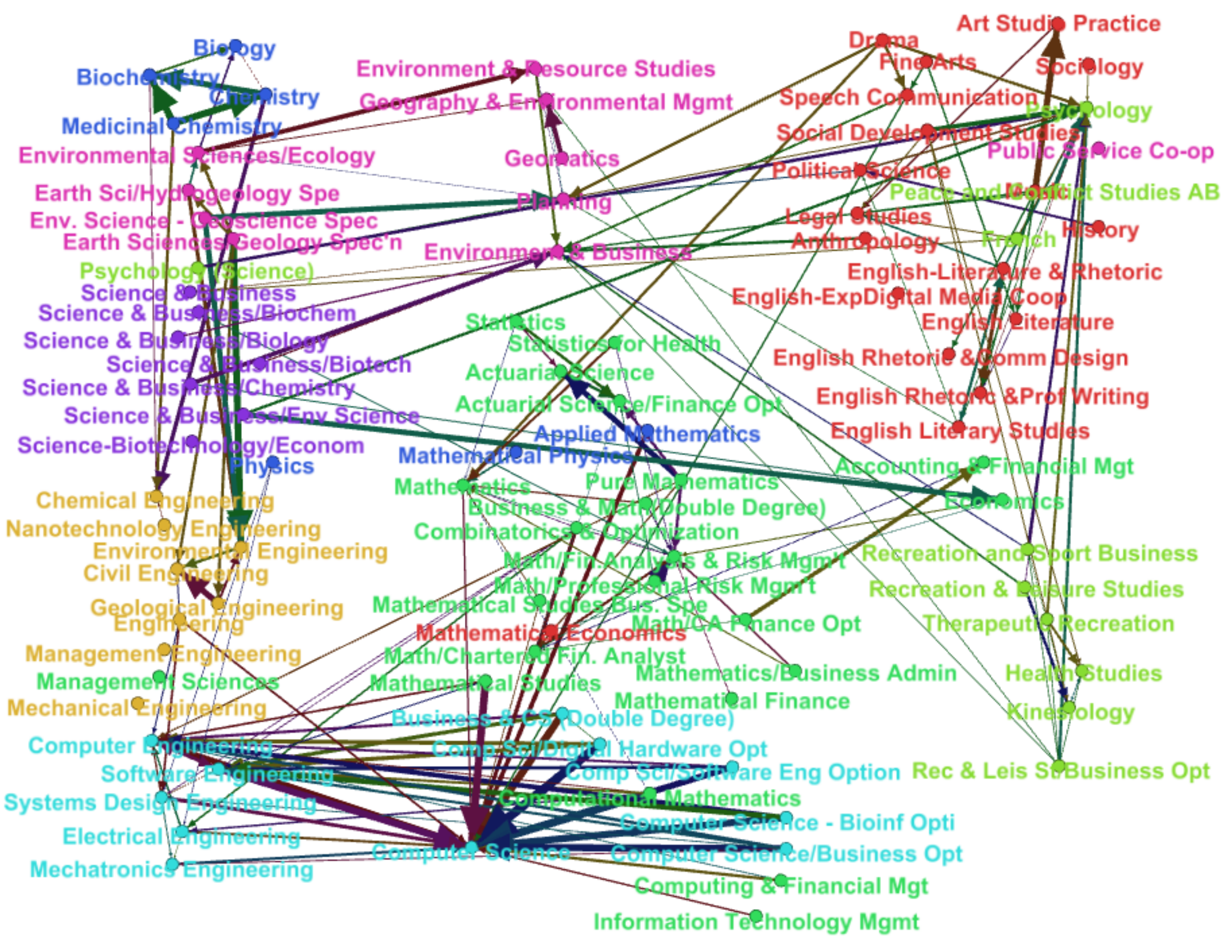


Figure 4.10: Full program graph with 5% thickest edges, colour-coded by clusters (8 clusters)

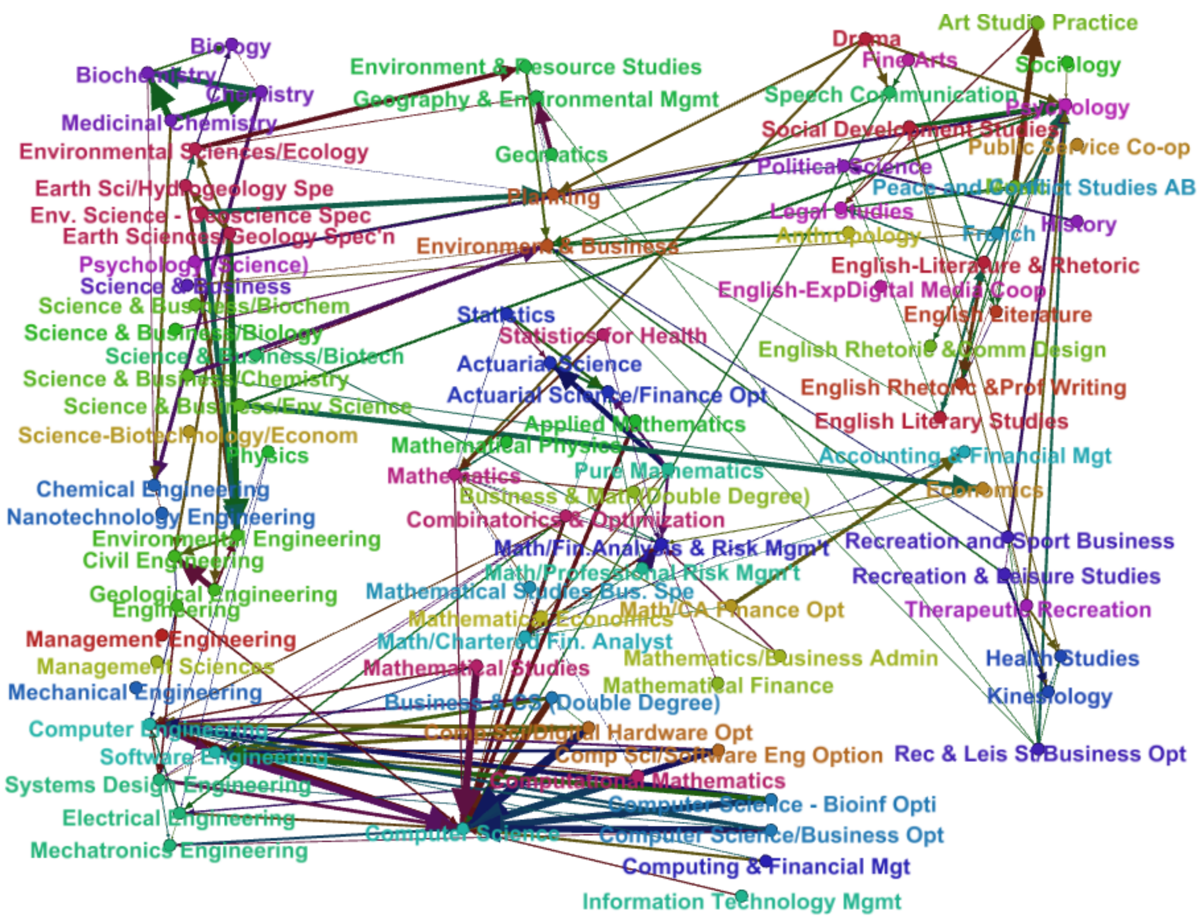


Figure 4.11: Full program graph with 5% thickest edges, colour-coded by clusters (35 clusters)

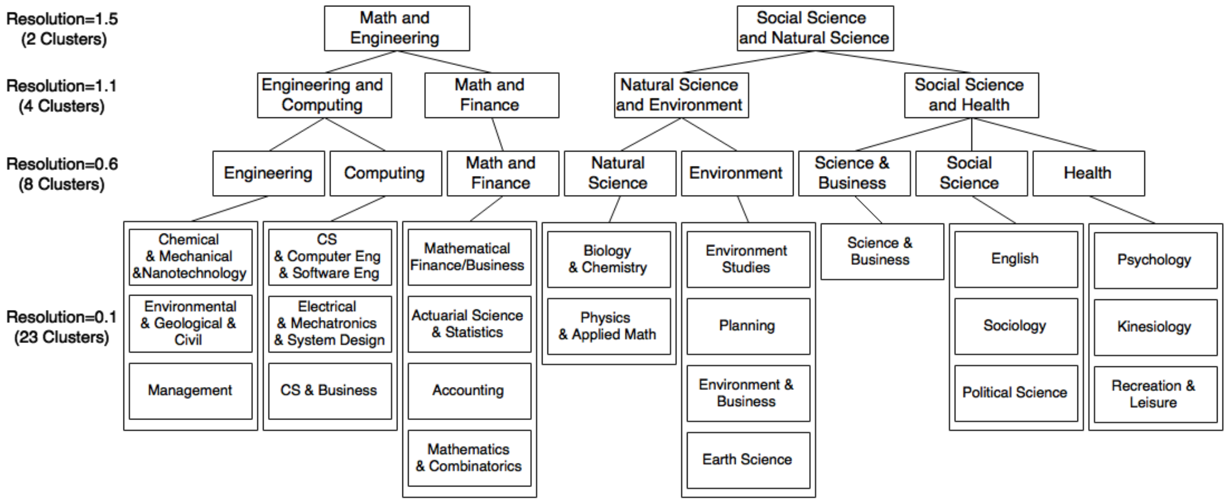


Figure 4.12: Hierarchy of partition results for various resolution parameters

When the resolution parameter is 1.5, the Math and Engineering cluster splits into two clusters: Computing & Engineering and Math & Finance. The former cluster is then divided into Engineering and Computing, which is then broken into three smaller clusters.

Throughout the first three layers, the AHS and Environment faculties are not fragmented. Most of the arts-related programs stay in one cluster, except Psychology and Economics. Psychology (from the Arts faculty) is in the same cluster as programs from AHS, while Economics (also from the Arts faculty) is with financial programs from the Mathematics faculty. The Engineering and Mathematics faculties are divided into three clusters. Similarly, programs in the Science faculty are allocated into three different clusters. These findings show that in the co-op system, AHS and Environment programs are more related.

Note that when the resolution parameter is 0.1 (Figure 4.11), there are 37 clusters. After manual inspection, we removed some clusters that appeared to represent noise rather than important relationships, and were left with 23 clusters.

Representative keywords from the job titles corresponding to the different clusters can be extracted with the help of word clouds. For example, Figure 4.13 shows the word cloud of job titles of common jobs in the Computing cluster (in the third layer, where the total number of clusters is 8). Employers who advertise to the Computing cluster can consider include these keywords in their job titles. Students from the programs of this cluster can search jobs with the frequent keywords. The word clouds for the other clusters are in Appendix C.2.

Figure 4.13: Word cloud of job titles of common jobs in the Computing cluster given 8 clusters

4.1.3 Finding Multi-Disciplinary Programs

Recall that our methodology for identifying multi-disciplinary programs requires a clustering result; once we have a clustering, for each program, we compute the entropy of the edge weight distribution across different clusters. In this section, we explore multi-disciplinary programs using two clustering results from the previous section: with 8 clusters and 37 clusters.

We hypothesized that cross-faculty programs or those with two areas in their name are more multi-disciplinary. We found that two such programs were indeed multi-disciplinary: Science & Business and English Rhetoric & Professional Writing. However, there were many other "regular" programs that appeared to be multi-disciplinary as well.

Table 4.1 lists the five programs with the highest and lowest entropy given 8 and 37 clusters. Of the five most multi-disciplinary programs found in either case, there were some "regular" programs such as Biology and Psychology. This was unexpected since they are not obviously cross-disciplinary programs, and will be investigated further later in this thesis. There were also some programs that we expected to be multi-disciplinary such as Science & Business and Environment & Business. On the other hand, Software Engineering and several other specialized programs were on both non multi-disciplinary lists.

Table 4.1: Most and least multi-disciplinary programs given 8 and 37 clusters

With 8 Clusters		With 37 Clusters	
Programs with Highest Entropy	**Programs with Lowest Entropy**	**Programs with Highest Entropy**	**Programs with Lowest Entropy**
Biology	Math/Professional Risk Management	Psychology	Chemistry
Sociology	Actuarial Science / Finance Opt	English Rhetoric & Professional Writing	Geological Engineering
English Rhetoric & Professional Writing	Software Engineering	Science & Business / Biotech	Software Engineering
Science & Business	Mathematical Finance	Speech Communication	Mechatronics Engineering
Speech Communication	Mechatronics Engineering	Environment & Business	Computer Engineering

4.1.4 Finding Competing Programs

We now search for programs with high set fan-out. Recall that these are the programs whose students tend to compete with students from other programs. High set fan-out indicates few jobs that only interviewed students from that particular program and no other programs.

We hypothesized that a majority of programs should have many jobs that only interview their students because students from other programs may not have the necessary skills or background knowledge. However, this is incorrect. Only 6 out of 93 programs had more than 20 percent of jobs that exclusively interviewed their students.

Figure 4.14 plots the set fan-out of all programs in descending order. Approximately 85 percent of the programs have the property that over 90 percent of the jobs that interview their students also interview students from at least one other program. Further, 29 programs have zero jobs that only interview their students; these programs are mostly from the Arts, Science, and Mathematics faculties. These programs were connected to an average of 24.31 other programs (in-degree), and had an average of only 4.9 students who interviewed for 15 jobs on average. It is likely that few jobs target these small programs, and even when jobs targeted these programs, they were likely not the sole targets. We will verify the potential rationale shortly.

At the other end of the curve, only 6 programs shared less than 80 percent of their jobs with other programs (or, in other words, there were only 6 programs where more than 20 percent of their jobs only interviewed their students). They are Civil Engineering, Account & Financial Management (AFM), Kinesiology, Mechanical Engineering, French, and Planning. Again, we

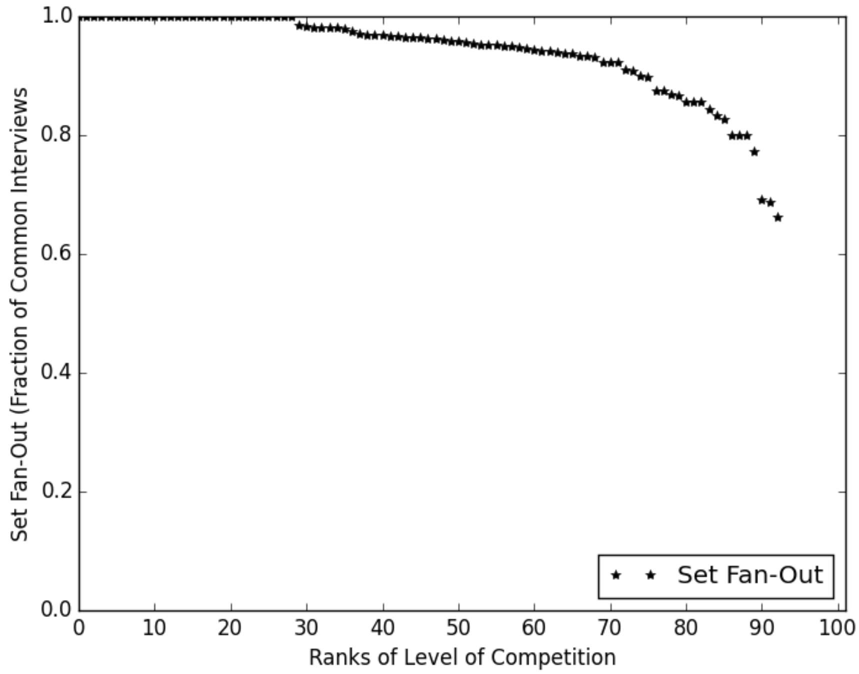


Figure 4.14: Set Fan-Out of all programs in descending order

will investigate these programs further in Section 4.3.

4.2 Senior Program Graph Analysis

Competition among programs may vary depending on the academic years. To investigate this, we analyze a program graph corresponding the the interview information of senior students (third year and up). The senior program graph contains 88 programs that have at least two senior students involved in the co-op process and 1,315 pairs of directed edges.

We hypothesized that senior students are more specialized and compete for jobs with students from fewer distinct programs. We found that the clusters obtained from the senior program graph were similar, and the importance of nodes, measured by centrality, did not vary significantly. However, the nodes in this graph generally had lower in-degree than those in the complete program graph and the density of the senior program graph was lower. This means that an employer offering a senior-level job tends to interview (senior) students from fewer distinct programs than a junior-level job, which confirms our hypothesis. On the other hand, the edge weights of the senior program graph is higher, meaning that the relationships that do exist are stronger. That *is, senior students from a particular pair of programs either did not compete at all (no edge) or interviewed for many jobs in common (thicker edges). If programs remained closely connected in the senior program graph, we considered this to be strong evidence of their similarity.*

4.2.1 Senior Program Graph Statistics

Overall, the senior program graph was still one connected component and its diameter remained at three. However, the average path length increased to 1.68 from 1.57. The density decreased from 0.44 to 0.34, i.e. 10 percent fewer program pairs had any interviews in common. Its global clustering coefficient also decreased from 0.64 to 0.54. The lower density and global clustering coefficient indicated that the programs of senior students were less inter-connected. Therefore senior students were more specialized in the context of co-op employment.

For individual nodes, as before, we assess their importance by degree centrality, closeness centrality and betweenness centrality. These three centrality measurements correspond to number of programs whose students interview for the same jobs, the reachability of a node to other nodes in the graph, and the bridge function of a program, respectively.

The degree centrality of nodes in the senior program graph ranged from 4 to 66. Compared to the full program graph, the average in-degree decreased from 40 to 30, meaning that, on average, senior students from a particular program had interviews in common with students from 10 fewer programs. However, the programs with the highest and lowest degree centrality were similar to those in the full program graph. The only difference is that Computer Science had the eighth highest number of direct competitors, but now its rank drops to 19. It means that fewer senior students from other programs had the same interviews with senior students from the Computer Science program, but junior students from several many other programs routinely interviewed for junior-level software and developer positions (as we noticed earlier). Adding edge weights, both the averages and the ranges of weighted in-degree (0.09-12.52 with an average of 2.89) and weighted out-degree (0.95-5.33 with an average of 2.89) were lower in the senior student graph. The programs with the highest and lowest weighted in-degree were the same as in the full program graph.

The closeness centrality of nodes in the senior program graph varied from 0.41 to 0.81, with an average of 0.6. Since the range for the full program graph was 0.51 to 0.92 and the average was 0.65, it indicate that senior students had fewer interviews in common with senior students from other programs. Medical Chemistry still had the lowest closeness centrality, and Mathematics and Environment & Business programs again had the highest closeness centrality.

On the other hand, the average betweenness centrality increased to 59.29 from 52.26 compared to the full program graph. Considering the extreme cases of the betweenness centrality, Environment & Business and Mathematics were still the highest. Their betweenness centrality measures increased from 311.09 and 347.11 to 371.17 and 365.95 respectively compared to the full program graph. The rise in this metric makes sense since fewer programs competed directly with each other in the senior program graph.

Considering edge weights alone, the range is from 0.002 to 1, with an average of 0.1. The range is higher than the one in the full program graph. Figure 4.15 illustrates a subgraph of the senior program graph, which is colour-coded by faculty, with the 5 percent heaviest edges. Compared to the full program graph (Figure 4.4), programs in Mathematics and Science faculties have higher intensity of competition while programs in Engineering have less intra-faculty competition. Computer Science is still central. However, its Engineering neighbours had less overlap with each other as indicated by thinner and fewer edges.

4.2.2 Finding Similar Programs

We follow the same approach as in the full graph analysis to find clusters of similar programs. We first search for near-cliques in the senior program graph where the clusters of three or more programs with senior students need to have a minimum density value of 0.8. Then, we apply the community detection algorithm to partition the graph into clusters.

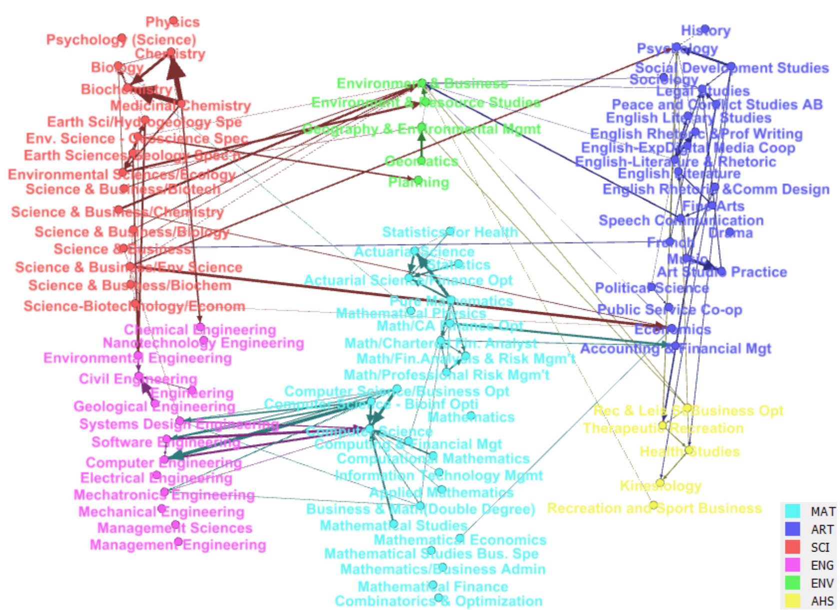


Figure 4.15: Senior student graph with 5% thickest edges, colour-coded by faculty

Near-Cliques

We found only 25 near-cliques, listed in Appendix C.3, in the senior program graph versus 135 in the full program graph. These near-cliques involve 46 programs and 104 edges. Even though there are still three connected components (Figure 4.16), the structure of them is significantly different than Figure 4.5.

In the blue connected component, some environmental and earth science related programs are included through the connection with Biology. Moreover, the five programs on the left form a clique in the full program graph, but senior Biology students only compete with senior Chemistry and Biochemistry students. It shows that senior Biology students may shift from a natural science focus to an environmental focus.

In the red connected component, the computing-related clique is smaller compared to that in the full program graph. Also, Finance-related programs are split into Statistics/Actuarial Science and Finance. Pure Mathematics and Business & Mathematics serve as bridges that glue the various parts into one connected component. Recall from the full program graph (Section 4.1.2) that Computer Science, Computer Engineering, and Software Engineering appeared together in 92 out of 135 near-cliques. In the senior program graph, only two out of 25 near-cliques included these three programs. Thus, the computing-related clique now includes fewer programs than

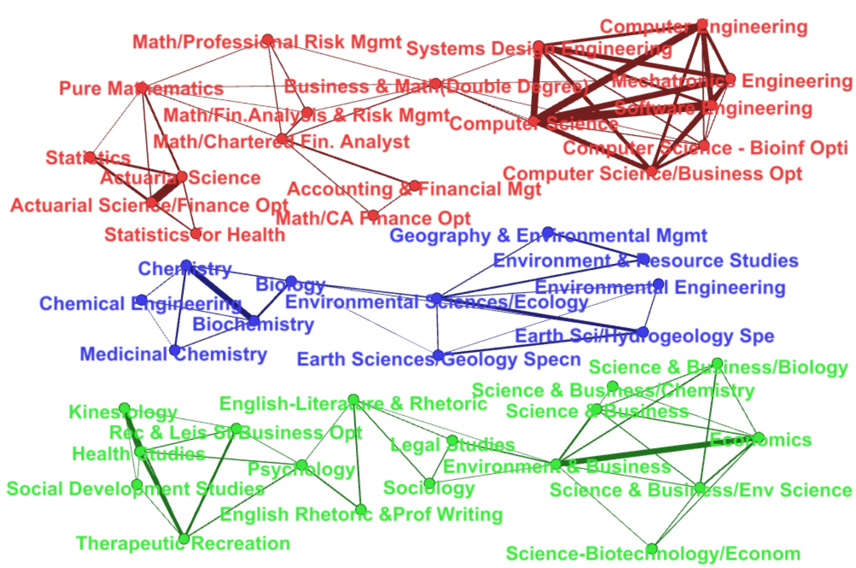


Figure 4.16: Graph composed of 25 near-cliques in the senior program graph, colour-coded by connected components

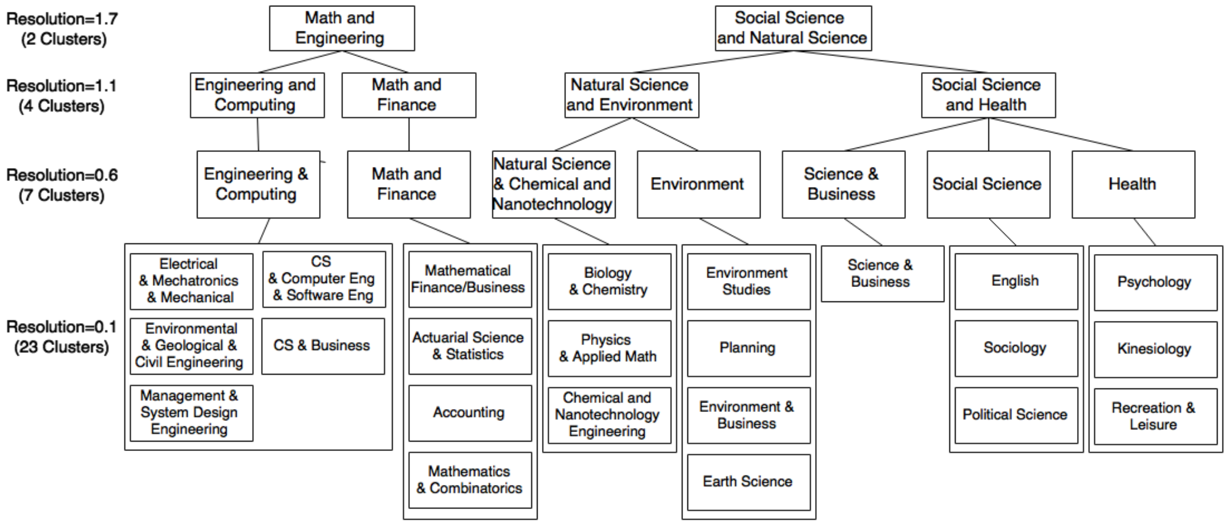


Figure 4.17: Hierarchy of partition results of senior program graph

in the full program graph. Again, this implies that junior students from many other programs interviewed for entry level software-related jobs, but not senior students.

In the green connected component, there are still three parts. However, the Arts part shrinks compared to the full program graph. That means fewer programs in the Arts faculty were strongly inter-connected. This finding supported our hypothesis that senior students were more specialized. In terms of the bridge programs, Psychology still serves as a connector. However, instead of Planning and Environmental Sciences/Ecology programs, Environment & Business connects the Science & Business part with others. It is unexpected that Environment & Business, as a program in the Environment faculty, forms a near-clique with Sociology, Legal Studies, and English-Literature & Rhetoric, which are all from the Arts faculty. This will be investigated in Section 4.3.1.

Community Detection

We used the same four resolution parameters as in the full program graph: 1.7 (2 clusters in Figure 4.18), 1.1 (4 clusters in Figure 4.19), 0.6 (7 clusters in Figure 4.20), and 0.1 (35 clusters in Figure 4.21). Figure 4.17 summarizes the cluster hierarchy.

Comparing to partition results in the full program graph, most of the clusters remain the same. The main differences are in the Engineering faculty, as we explain below.

Civil, Geological, and Environmental Engineering are within the Environment cluster from the beginning instead of being grouped with other Engineering programs. When the resolution parameter equals 0.6, Engineering and Computing remains a single cluster; however, Chemical and Nanotechnology Engineering are no longer a part of this cluster and instead they are part of the natural science cluster. Therefore, there are only seven clusters in the senior program graph. At the last level, the Engineering & Computing cluster was broken down differently. Electrical, Mechatronic and System Design Engineering are no longer in the same cluster. Instead, senior students in System Design Engineering have tighter competition with Management Engineering, and Electrical, Mechatronics and Mechanical Engineering are in the same cluster.

Combining the community detection results on both the full program graph and the senior program graph, we confirm our choice of "home" faculty for two out of three cross-faculty programs in Section 3.1.2. Specifically, we correctly chose Faculty of Mathematics for Computing and Financial Management and the Faculty of Engineering for Software Engineering. This is because Computing and Financial Management and other programs in Faculty of Mathematics formed a Mathematical Finance and Business cluster, and Software Engineering, Computer Engineering, and Computer Science formed a cluster under the Engineering and Computing cluster. However, we incorrectly assigned the Mathematical Physics program to Faculty of Mathematics; this program and Physics were assigned to the Natural Sciences cluster.

4.2.3 Finding Multi-Disciplinary Programs

Table 4.2 lists the most and least multi-disciplinary programs given 7 and 33 clusters. We use the same methodology as before (entropy of the distribution of edge weights among the clusters).

Compared to the results on the full program graph (Table 4.1), even though the order of these programs has changed slightly, Biology, Psychology and Environment & Business remained multi-disciplinary. Recall that these programs connect multiple near-cliques to form a connected component in the previous section.

Notably, in the context of senior students, English Literary Studies, Mathematics, and Management Engineering were multi-disciplinary, while French and Geomatics were not. Digging deeper, Figure 4.22 and 4.23 show the word clouds for the job titles for which junior and senior Management Engineering students interviewed. Junior students interviewed for engineering, coordinator, analyst and/or project related positions. However, it is rather difficult to identify a dominant keyword for senior jobs. As a result, it is not surprising that the senior graph revealed Management Engineering to be multi-disciplinary but the complete graph did not.

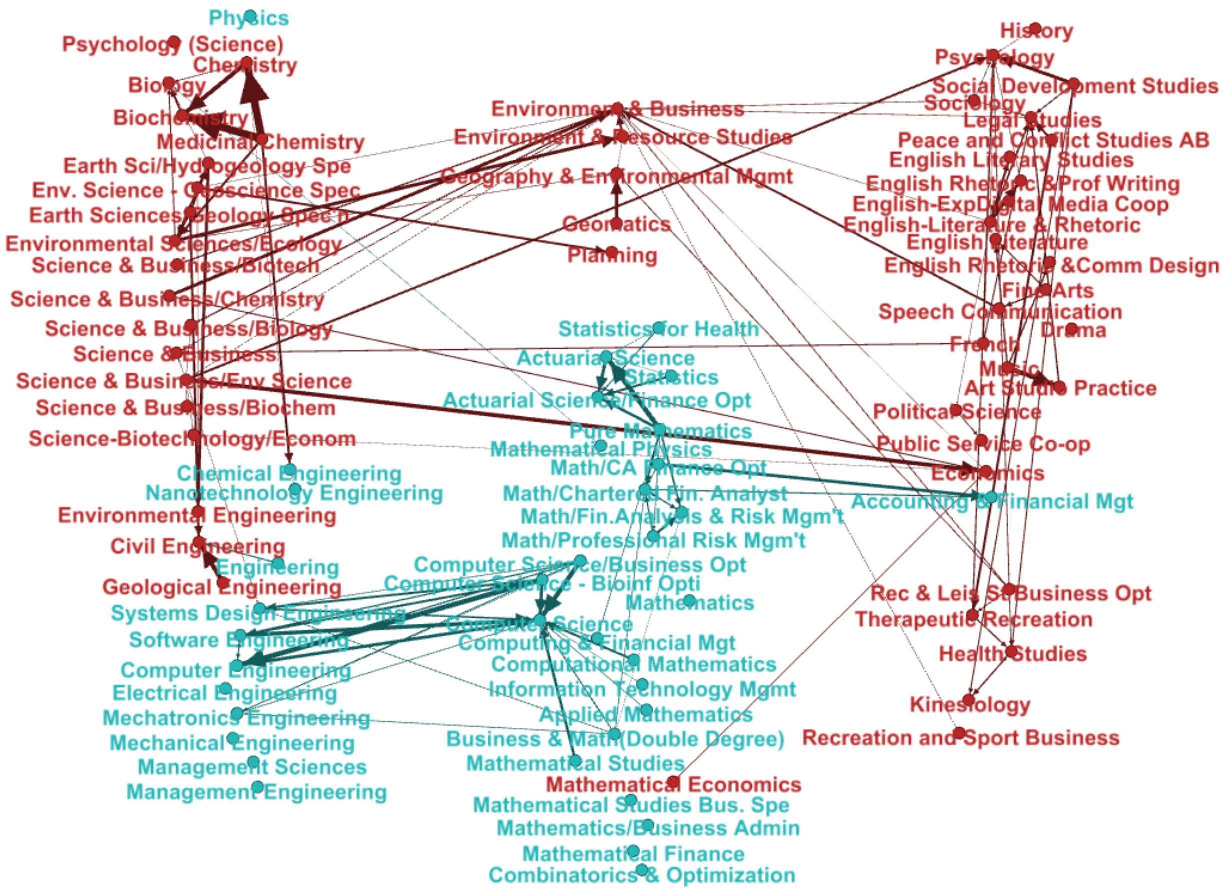


Figure 4.18: Senior program graph with 5% thickest edges, colour-coded by clusters (2 clusters)

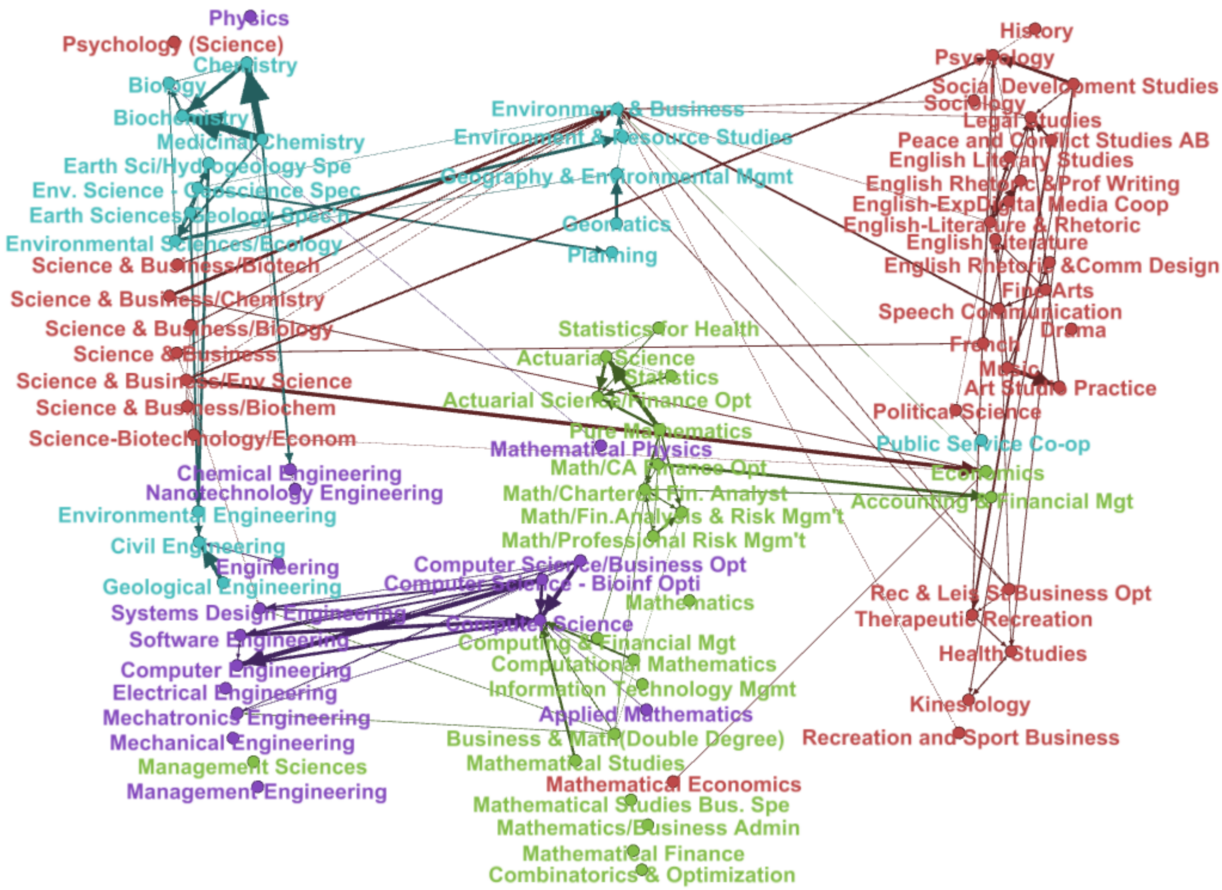


Figure 4.19: Senior program graph with 5% thickest edges, colour-coded by clusters (4 clusters)

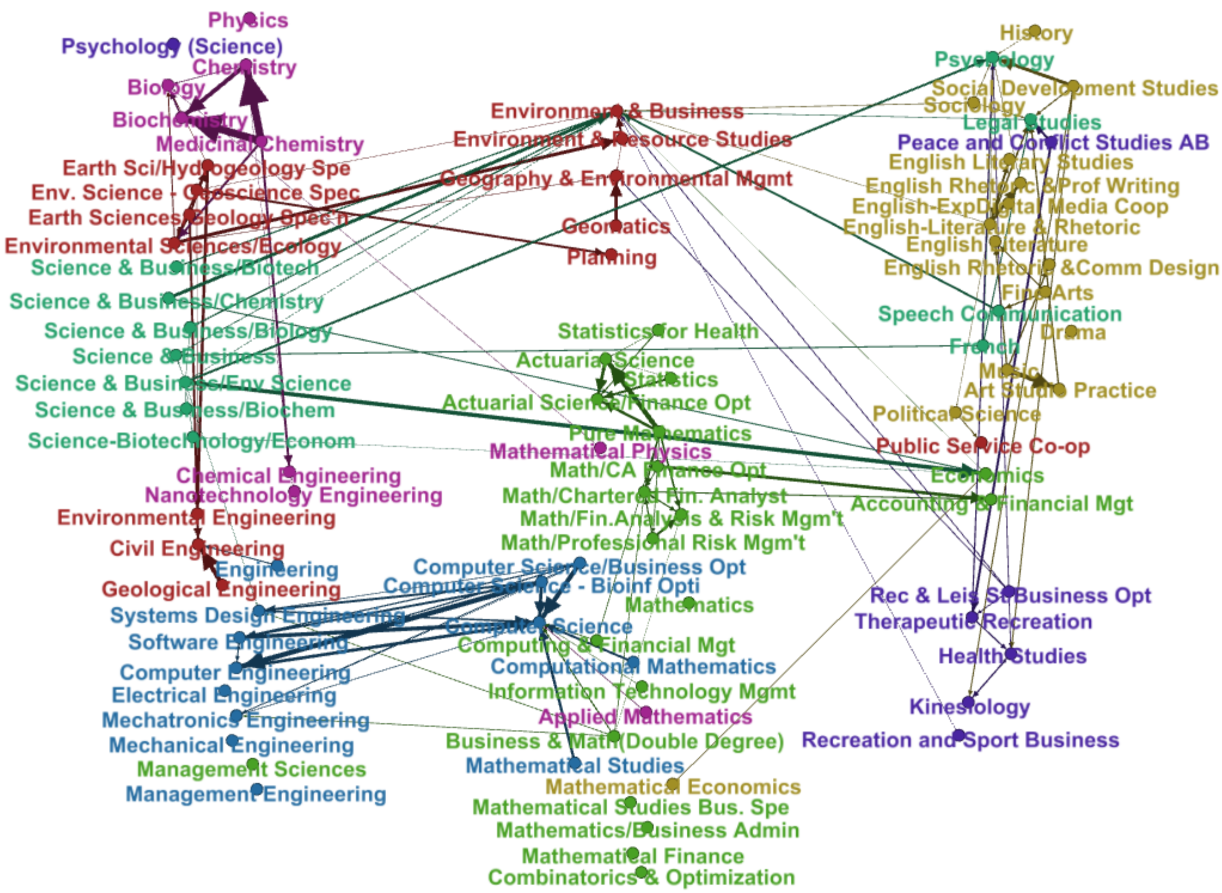


Figure 4.20: Senior program graph with 5% thickest edges, colour-coded by clusters (7 clusters)

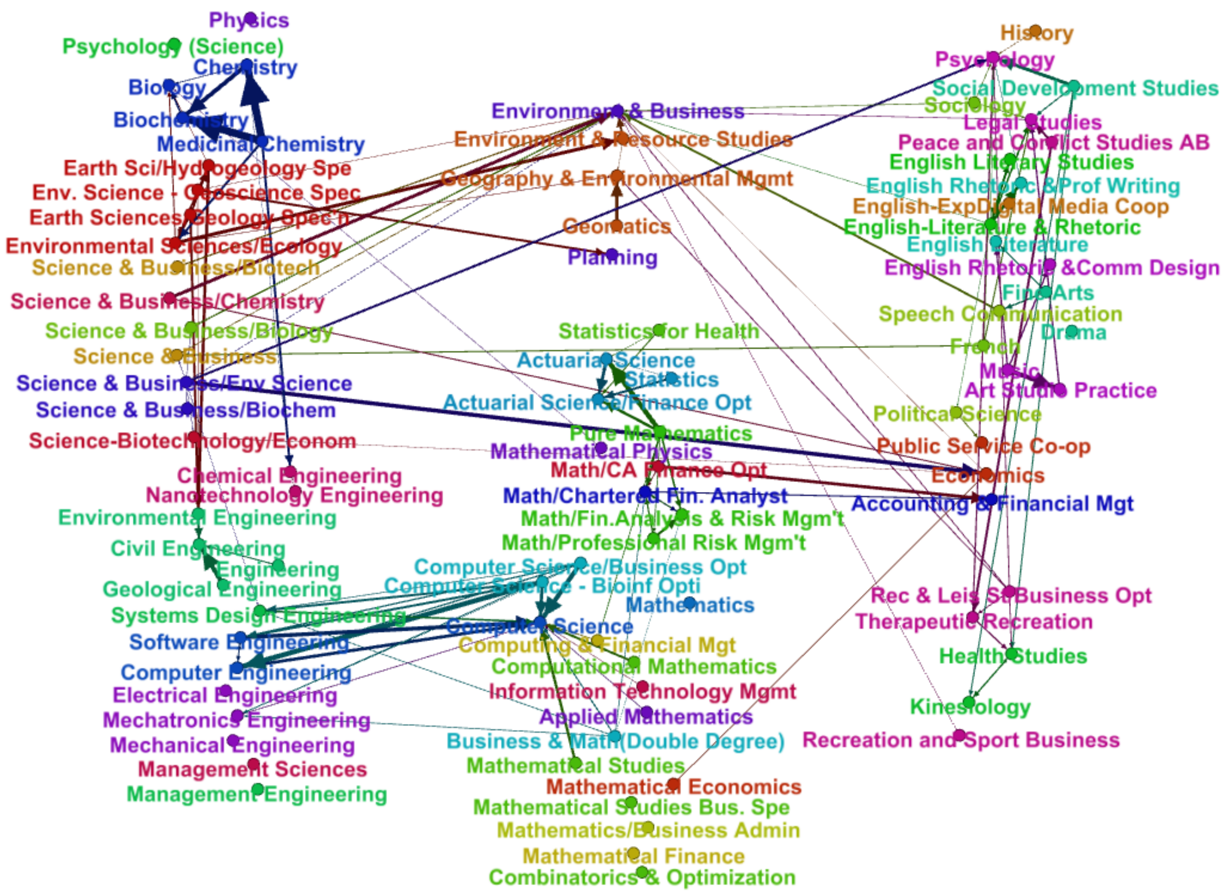


Figure 4.21: Senior program graph with 5% thickest edges, colour-coded by clusters (33 clusters)

4.2.4 Finding competing among Programs

Recall that extent of competition a program faces is measured by set fan-out, which measures the percentage of jobs that also interviewed students from other programs. Programs with the smallest set fan-out are more specialized in the context of co-op employment, and programs with the largest set fan-out may face the highest competition. Figure 4.24 plots the fan out within the senior program graph in descending order. Compared to the full program graph, set fan out appears generally lower: only half the programs have over 90 percent of their jobs interviewing students from other programs. This supports our hypothesis of senior students being more specialized.

However, there are still 16 programs, such as Business & Mathematics, which do not have any jobs that interviewed only their students. The average number of senior co-op students in these programs is only 3.4, which is very small. There might be few jobs that specifically target these programs, so even senior students for these programs had to interview for jobs advertised to other programs. We will return to this issue in Section 4.3.

On the other hand, there are 8 programs where more than 30 percent of the jobs only interviewed their own students. They are Mathematical Studies Business, Environmental Science - Geoscience, Information Technology Management, AFM, Kinesiology, Chemical, Mechanical, and Civil Engineering. French and Planning were one of the top six specialized programs in the full program graph, but not in the senior program graph. Civil Engineering is the only program

Table 4.2: Most and least multi-disciplinary programs given 7 and 33 clusters in the senior program graph

With 7 Communities		**With 33 Communities**	
Programs with Highest Entropy	**Programs with Lowest Entropy**	**Programs with Highest Entropy**	**Programs with Lowest Entropy**
Science & Business/Biochem	Geological Engineering	Mathematics	Geological Engineering
English Literary Studies	Software Engineering	Psychology	Mathematical Studies
Science & Business/Env Science	French	English Rhetoric & Professional Writing	Chemistry
Biology	Mechatronics Engineering	Environment & Business	Geomatics
Science & Business	Civil Engineering	Management Engineering	Mathematical Physics

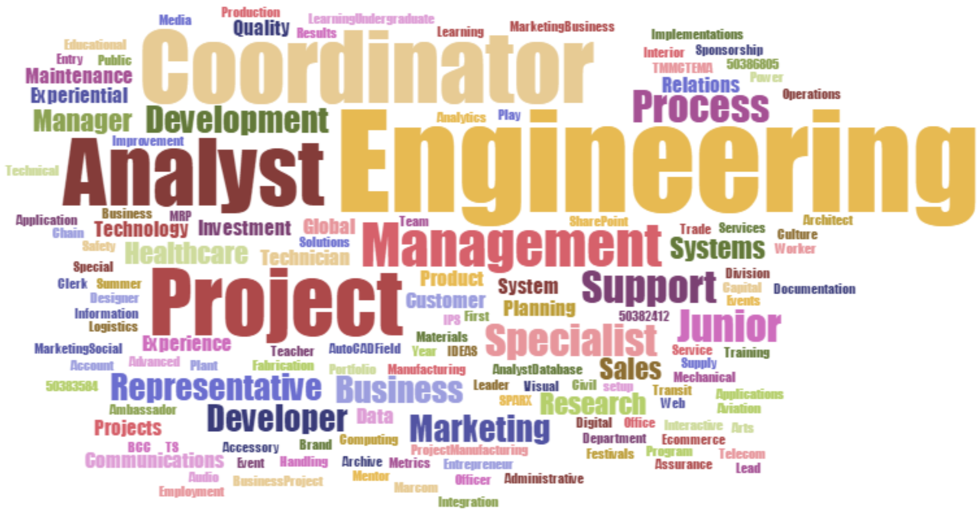


Figure 4.22: Word cloud for job titles of jobs that interviewed junior Management Engineering students

Figure 4.23: Word cloud for job titles of jobs that interviewed senior Management Engineering students

that was the least multi-disciplinary and specialized based on our definition.

4.3 Further Investigation

This section further investigates some of the unexpected results we have reported so far. Unless specified otherwise, we use the senior program graph: since we saw that senior students appear to be more specialized, any evidence of strong competition in the senior graph warrants further study.

4.3.1 Finding Similar Programs

We found that the layout of academic programs did not reflect the relationships of academic programs in the co-op system. For instance, a significant fraction of jobs that interviewed students from the Faculty of Science were business and marketing related.

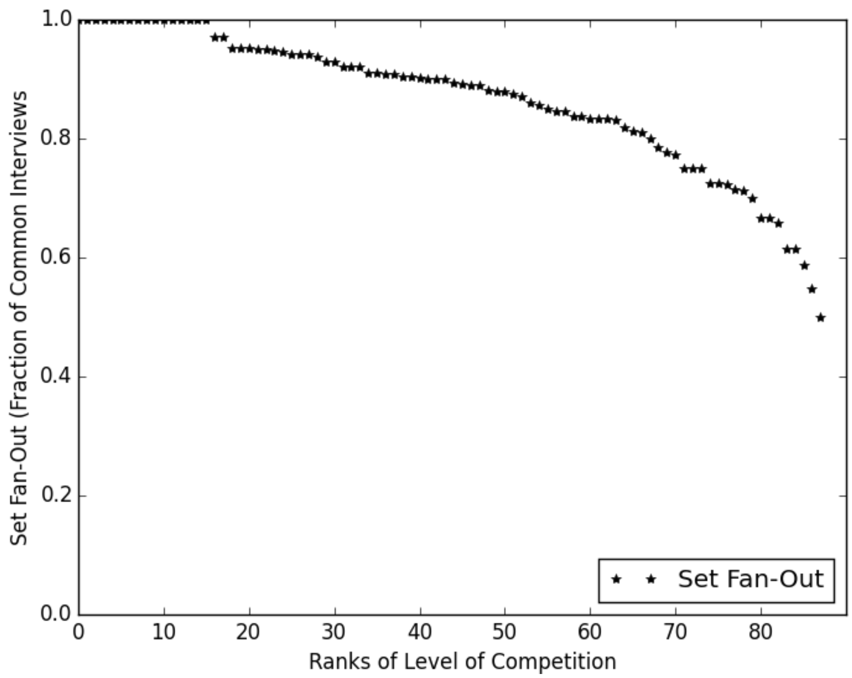


Figure 4.24: Level of competition of programs in the senior program graph in descending order

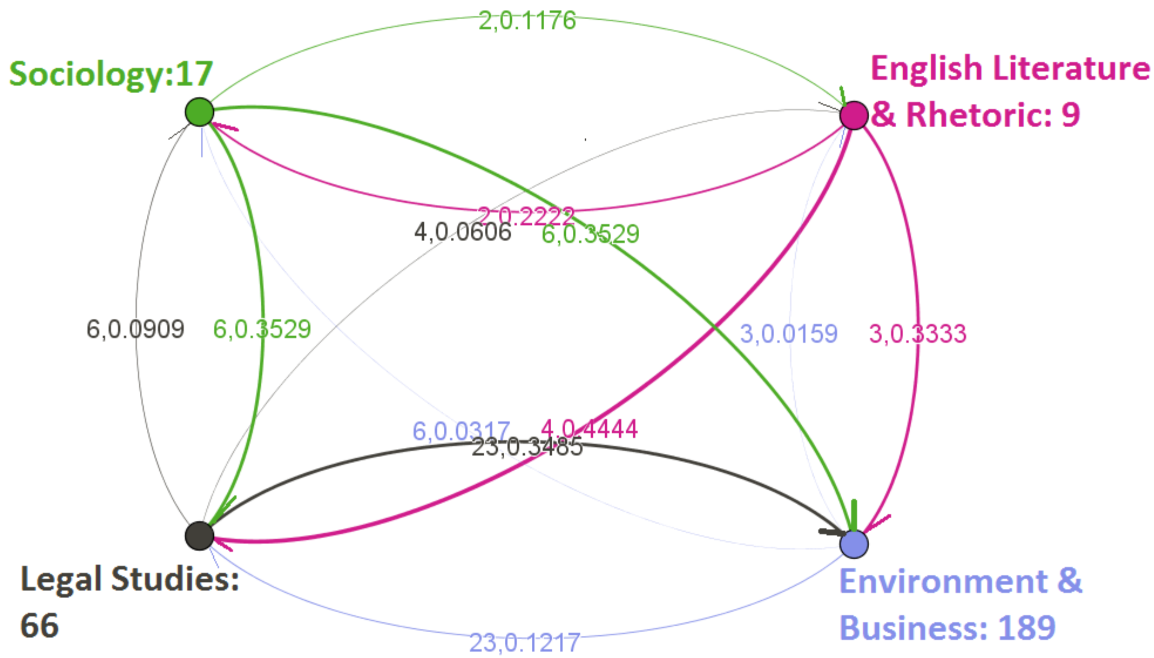


Figure 4.25: Subgraph of unexpected near-clique consisting of Sociology, Legal Studies, English-Literature & Rhetoric and Environment & Business

Near-Cliques

Recall that most of near-cliques found in the senior program graph were expected, but the near-clique of Sociology, Legal Studies, English-Literature & Rhetoric and Environment & Business was not expected since Environment & Business is from a different faculty. Furthermore, we expected Pure Mathematics to be specialized but it was in several near-cliques. We investigated the reasons that might lead to both surprising findings. We found that the common jobs in the first case were marketing and communication related, but only a few students from Environment & Business and Legal Studies participated in three or more such interviews. For the second case, we discovered that students from Pure Mathematics were interviewed for trading jobs, which also interviewed students from several near-cliques.

Figure 4.25 shows the clique of the senior program graph consisting of Sociology, Legal Studies, English-Literature & Rhetoric and Environment & Business. The nodes are labelled with the program name and the number of jobs that interviewed their students. The edges are labelled with the number of common jobs and edge weights and colour-coded by the source node.

For each pair of programs in this near-clique, we studies the types of jobs that are in common by analyzing the frequent keywords in their job titles. Pairs that contain English Literature & Rhetoric is not studied since the low number of jobs in common (less than five) and the findings may be insignificant. For the 23 jobs that interviewed students from both Environment & Business and Legal Studies, only 6 of these jobs advertised to Environment & Business, and none specified Legal Studies. In terms of the job titles (Figure 4.26), they were mostly business, marking or coordinator jobs, which were not legal or environmental. Sociology had six job interviews in common with Legal Studies and a different six job interviews with Environment & Business. Despite that these 12 jobs were different and they were also not in common with the 23 aforementioned jobs, job titles of these 12 jobs (Figure 4.27 and Figure 4.28) shared some common keywords, such as marketing.

In terms of the number of unique students participated in these common interviews, English-Literature & Rhetoric, Sociology, Environment & Business, and Legal Studies had 1, 4, 27, and 14 students, respectively, participated in the common jobs in this near-clique. Only one Environment & Business student and four Legal Studies students had three of more of such interviews. These five students may truly interested in marketing and communication jobs, and they probably should also be considered as suitable candidates for other similar positions. Due to our limited knowledge of students' and employers' behaviour, the reasons behind this situation is not obvious.

Students from Pure Mathematics only interviewed for 11 jobs, but they had the same interviews with students from 18 different programs and appeared in three near-cliques (Figure 4.29), two of which are cliques. Pure Mathematics was a bridge program that connected the two cliques through trading jobs. As we show in Figure 4.30, these jobs did not interview exclusively from programs belonging to one of the two cliques, and were mostly trading related.

Next, we investigate whether or not the other jobs common to the two cliques involving Pure Mathematics were also trading related. The answer is no. The jobs that interviewed students in the actuarial clique (left) were mainly actuarial positions (Figure 4.31), and the jobs that interviewed students in the FARM clique (right) were analyst positions relating to business and risk management (Figure 4.32). As a result, the connections between Pure Mathematics and several students from the two cliques lead to the high betweenness centrality of Pure Mathematics because of trading positions.

Figure 4.26: Word cloud of job titles of 23 jobs that interviewed both Environment & Business and Legal Studies

Figure 4.27: Word cloud of job titles of 6 jobs that interviewed both Environment & Business and Sociology

Figure 4.28: Word cloud of job titles of 6 jobs that interviewed both Legal Studies and Sociology

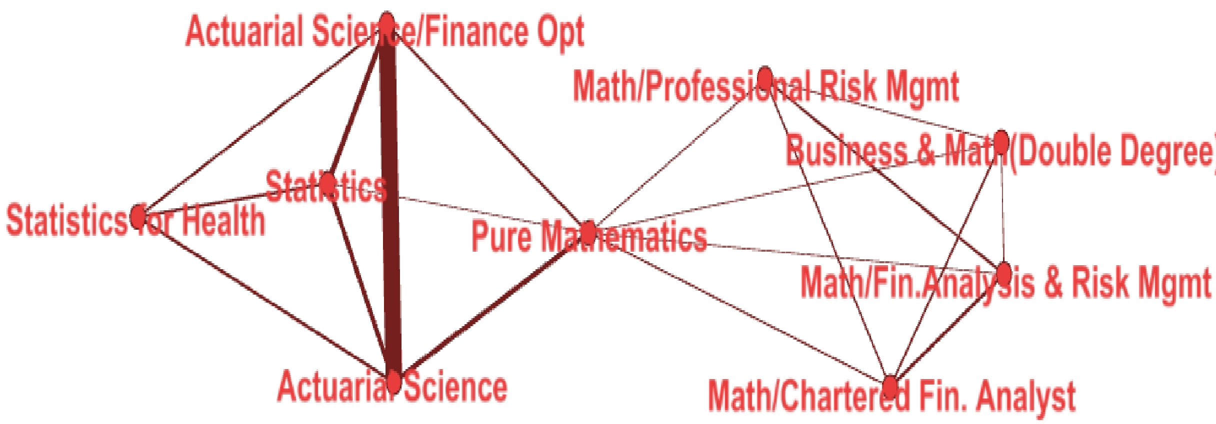


Figure 4.29: Near-cliques that contain Pure Mathematics

Figure 4.30: Word cloud of job titles of 11 jobs that interviewed from Actuarial Clique

Figure 4.31: Word cloud of job titles of 75 jobs that interviewed from Actuarial Clique

Figure 4.32: Word cloud of job titles of 54 jobs that interviewed from FARM Clique

Community Detection

Most of our community detection results were expected. For instance, it is intuitive that the Computer Science program had a strong relationship with Computer Engineering and Software Engineering. However, there are several interesting clusters that are not aligned with our hypothesis which require further analysis:

1. **Hypothesis:** The Science faculty should be in the same group with Engineering and Mathematics faculties. They will form a Science, Technology, Engineering and Mathematics (STEM) cluster. **Finding:** the Science faculty was in the same cluster with the Environment, Arts and AHS faculties.
2. **Hypothesis:** Programs in the same department should belong to the same cluster. **Finding:** Electrical Engineering and Computer Engineering are offered by the same department but were in different clusters when the resolution parameter is 0.1.

For the first unexpected result, we found that not only did the Science faculty have a small fraction of jobs in common with Engineering and Mathematics faculties, but also the majority of jobs which interviewed Science students were business-related. There were 293 jobs that interviewed at least one senior Science student. 85 of them also interviewed at least one senior Engineering student. 41 out of these jobs did not initially target any Science programs. 25 of them also advertised to Arts, AHS, or Environment programs. In addition to those 85 jobs, 46 jobs interviewed at least one senior Mathematics student. However, these jobs were not exclusive to Mathematics or Science students. 34 of these jobs choose non-STEM programs as their targets and also interviewed students from non-STEM programs. Jobs that ended up interviewing Science students were heavily business, marketing or research-oriented (Figure 4.33), which were not popular jobs in the Engineering and Mathematics faculties. As a result, the Engineering and Mathematics faculties did not have strong links with Science programs and therefore were not in the same cluster. One interesting issue that we cannot vaerify due to lack of data is whether Science students were always interested in business jobs and therefore various Science & Business programs were created, or whether the creation of Science & Business programs has caused Science students to seek out business related jobs.

For the second unexpected result, Electrical and Computer Engineering were only related through a small number of software jobs. Computer Engineering jobs were much more software oriented while Electrical Engineering jobs were more hardware oriented. Senior students in Electrical Engineering interviewed for 198 jobs, and senior students in Computer Engineering interviewed for 237 jobs. Surprisingly, there were only 42 common jobs, and they were software or developer positions. For the 195 jobs in $J_{ComputerEng} - (J_{ElectricalEng} \cap J_{ComputerEng})$, only 49

Figure 4.33: Word cloud of job titles of jobs that interviewed at least a Science Student and students from other faculties

jobs (25 percent) considered Electrical Engineering as their targets. These 49 jobs were primarily developer and software positions, which were more related to Computer Science and Software Engineering (Figure 4.34). For the 156 jobs in $J_{ElectricalEng} - (J_{ElectricalEng} \cap J_{ComputerEng})$, 96 jobs (62 percent) still advertised to Computer Engineering, but ended up not interviewing any Computer Engineering senior students. Even though these 96 jobs were somewhat related to software (Figure 4.35), there were still many of them that were hardware oriented. As a result, we could potentially advise employers to target Mechatronics and Mechanical Engineering, in addition to Electrical Engineering, if the positions are not software related.

Figure 4.34: Word cloud of job titles of 49 jobs in $J_{ComputerEng} - (J_{ElectricalEng} \cap J_{ComputerEng})$ and advertised to Electrical Engineering

Figure 4.35: Word cloud of job titles of 96 jobs in $J_{ElectricalEng} - (J_{ElectricalEng} \cap J_{ComputerEng})$ and advertised to Computer Engineering

4.3.2 Finding Multi-disciplinary Programs

There are two ways in which a program may appear to be multi-disciplinary. First, it may have multiple types of students, some predominantly competing for jobs with one cluster of programs and other competing for jobs with a different cluster. For instance, perhaps a Science & Business program consists of half science-oriented students and half business-oriented students. The other possibility is that almost every student in a multi-disciplinary program is well-rounded and obtains interviews for diverse jobs. *In the first case, the multi-disciplinary program consists of various groups of specialized students, while in the second case, the multi-disciplinary program consists of well-rounded individuals. We found that multi-disciplinary programs in fact consist of well-rounded students.*

We back up this finding by focusing on Psychology, Biology, Management Engineering, and Environment & Business. The first two programs consistently had the highest entropy and seem to be the "bridge" programs shown in Figure 4.5. Management Engineering is one of the programs that is especially multi-disciplinary in the context of senior students. Environment & Business represents programs that are intuitively multi-disciplinary due to having two disciplines in their names. For each student, we computed the number of clusters that its competitors are from. Figures 4.36 and 4.37 plot the cumulative percentage of students over the number of clusters for 7 and 33 clusters, respectively. All four of these programs programs had 20 percent or fewer students that competed with other students from only cluster,

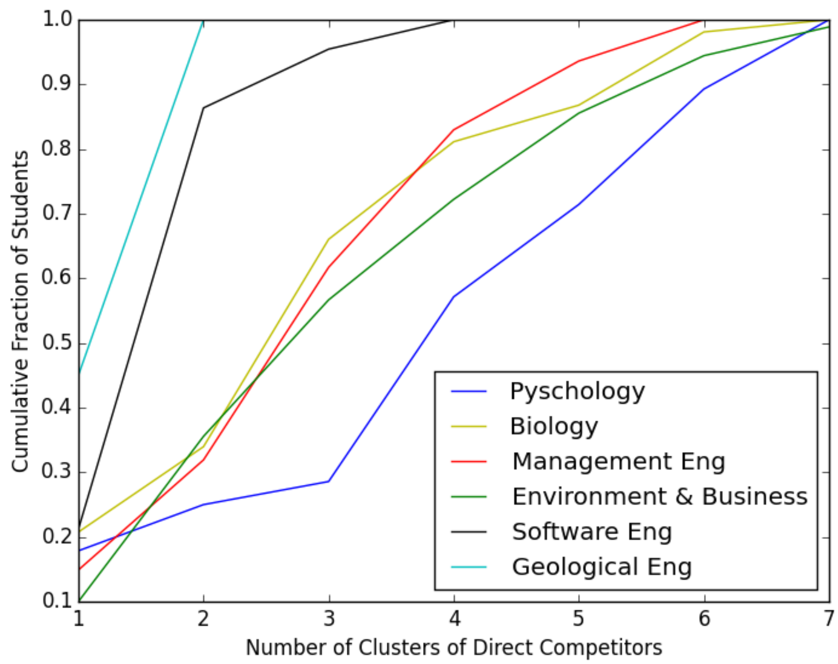


Figure 4.36: Cumulative percentage of students over number of clusters of direct competitors (7 clusters)

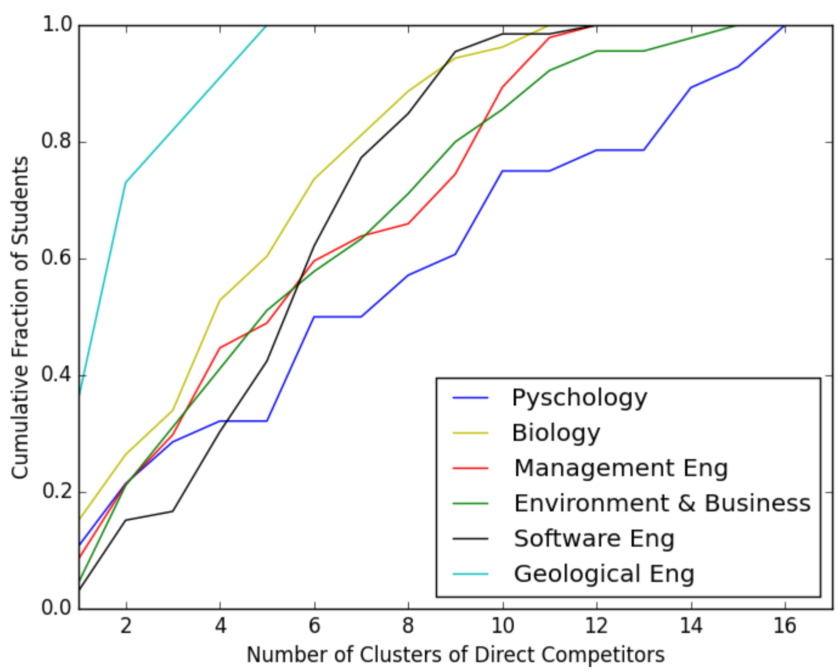


Figure 4.37: Cumulative percentage of students over number of clusters of direct competitors (33 clusters)

For the least multi-disciplinary programs, such as Software Engineering and Geological Engineering, we found that the students are specialized. In particular, for Geological Engineering, most of the senior students interview for jobs that only interview other students from one cluster, even when all the programs were broken down into 33 clusters.

Other potentially multi-disciplinary programs such as Science & Business and Business & Mathematics did not have the highest entropy of distribution of their edge weights and were not found to be particularly multi-disciplinary. In fact, 40 percent of the connections to Business & Math came from its own cluster.

4.3.3 Finding Competing Programs

More specialized programs (i.e. those having low set fan-out) beg the question as to whether their jobs that did not interview students from other programs are truly tailored to the specialized program. We found that most of the job titles of the jobs that exclusively interviewed students from the specialized programs were very different than the job titles of the jobs that interviewed students from the specialized programs and their direct competitors.

For programs with the highest set fan-out, do the same direct competitors appear in every single job interview or do certain competitors cover a specific type of jobs? We found that these programs only had one or two types of competitors.

For the programs with low set fan-out, we select Civil Engineering as an example. We found that students in Civil Engineering had interviewed for many civil or structural jobs that were exclusive to Civil Engineering students. Civil Engineering had 100 senior students and 155 distinct jobs. Though the percentage of jobs that only interviewed students from Civil Engineering is close to 45 percent, 85 jobs still interviewed senior students from other programs. Out of these 85 jobs, Environmental, Mechanical, and Geological Engineering appeared in 22 percent, 14 percent and 14 percent of the jobs respectively. For the 70 jobs that did not interview from other programs, 69 of them advertised to Civil Engineering, and 22 of them only targeted Civil Engineering. Moreover, Figure 4.38 shows the word cloud of job titles of 85 Civil Engineering jobs that also interviewed students from other programs, while Figure 4.39 shows the word cloud of job titles of 70 jobs that only interviewed students from Civil Engineering. Despite "Engineering" being the most frequent word, these 70 jobs had keywords that were more tailored to Civil Engineering, and not applicable to Environmental Engineering and other programs. For instance, such keywords include civil, bridge, structural, traffic and transportation.

Out of 17 programs with 100 percent set fan-out (i.e., no jobs that only interviewed their own students), Business & Mathematics students had the highest number of distinct jobs that interviewed them (25 jobs) and Environmental Sciences/Ecology had the highest number of students

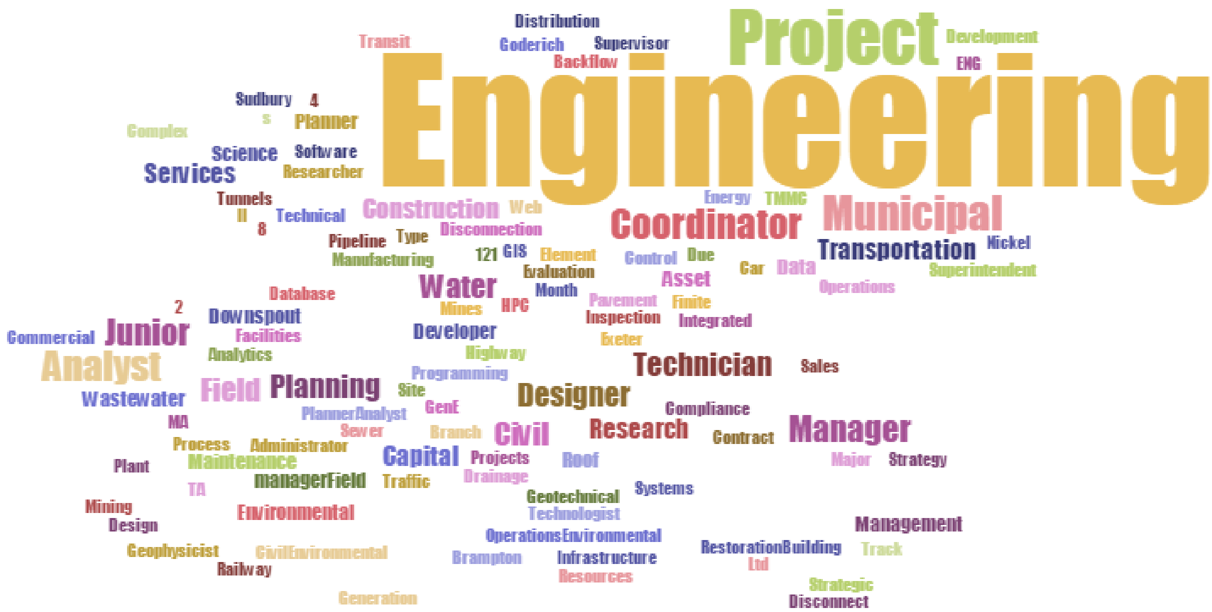


Figure 4.38: Word cloud of job titles of 85 Civil Engineering jobs that also interviewed students from other programs

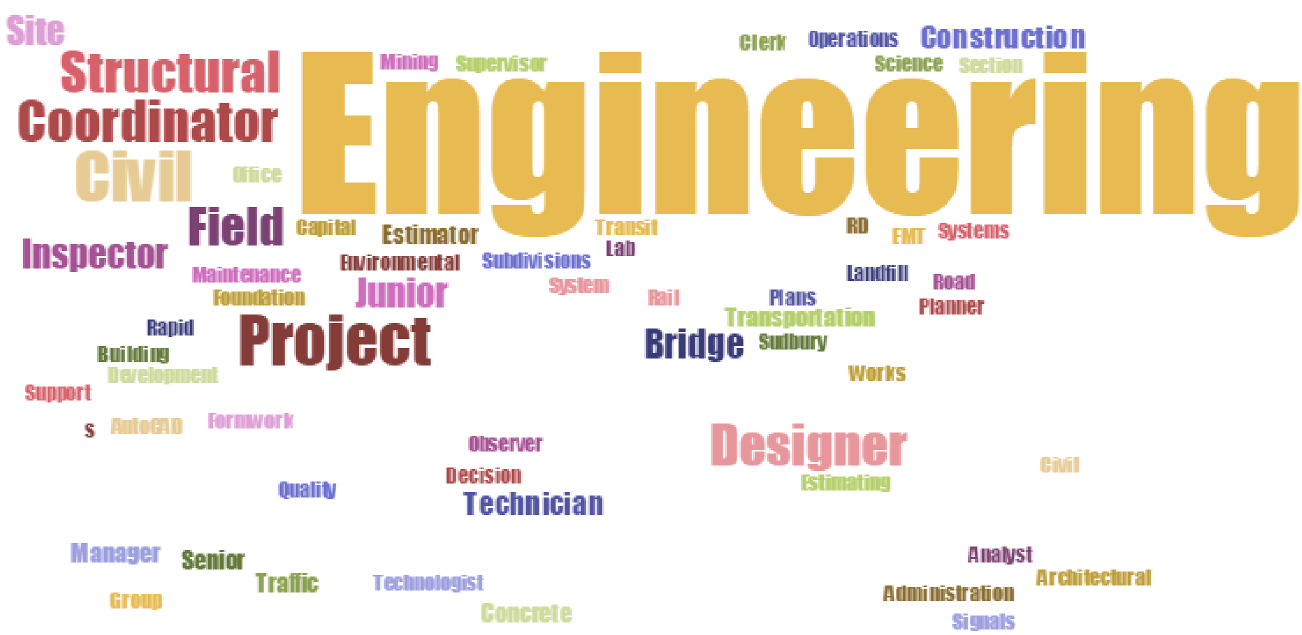


Figure 4.39: Word cloud of job titles of 70 Civil Engineering jobs that only interviewed students from Civil Engineering

that participated in any interviews (8 students). For Business & Mathematics, while these 25 jobs interviewed an average of 17 students, they only interviewed one Business & Mathematics student. Furthermore, no jobs target to the Business & Mathematics program specifically. 10 of these 25 jobs were computer science related, while the rest were financial positions.

For the Environmental Sciences/Ecology program, its senior students interviewed for 23 jobs. For 5 out of 23 jobs, more than 50 percent of interviewees were from Environmental Science/Ecology. 14 of 23 jobs also interviewed senior students from Environment & Resource Studies. In the remaining of jobs, students in the Environmental Sciences/Ecology program had direct competitors from 19 other programs, which belonged to 5 different clusters from a total of 7 clusters. As a result, for Environmental Sciences/Ecology, it had one dominant competitor, and many small competitors.

In the co-op context, we may recommend the university attract more job opportunities for these programs. Prior to the creation of such opportunities, students from these programs may consider applying for jobs that target their dominant competitor.

Chapter 5

Conclusion and Future Work

In this thesis, we presented a solution towards improving the co-operative education process. We observed that academic programs are typically used by students and employers to advertise and search for jobs, but it is not always clear how one program differs from another, especially given that universities have recently been creating new programs. In response to this problem, we developed a novel methodology to characterize the relationships among academic programs with respect to the job interviews obtained by students from these programs. The insight behind the methodology was to transform co-op interview data into a *program graph* which revealed that students from certain programs interview for the same jobs as those from other programs. We performed graph analyses such as finding communities, nodes connected to many communities and nodes strongly connected to their neighbours to describe the program relationships.

We applied the proposed methodology on a lage co-op data set from a major Canadian university. Our main findings along with their significance can be summarized as follows.

- The layout of academic programs did not always aligned well with the groups of closely-connected programs in the program graph. For example, Electrical Engineering and Computer Engineering are in the same Electrical and Computer Engineering department, but they were placed in different clusters in the community detection results. Programs from different programs or even separate faculties such as Economics from Faculty of Arts and Professional Risk Management from Faculty of Mathematics were more strongly connected. On the other hand, our clustering and community detection results naturally correspond to job categories and academic specializations. We developed a new job classification hierarchy that can be used in the co-op system to advertise jobs to groups of related programs. For example, one of the clusters we identified consists of Software Engineering, Computer Engineering and Computer Science, which aligns with software jobs. Our hierarchy is also useful for students as it shows the types of jobs that students from different programs qualify for, and therefore can help them choose programs of study that correspond to their desired career.

- We identified multi-disciplinary programs as those which have strong connections to multiple clusters. Interestingly, some programs that were not expected to be multi-disciplinary, such as Psychology, were identified as such. These programs include students who obtained interviews for various types of jobs. On the other hand, some programs that we expected to be multi-disciplinary, such as Business & Math, were not (in our context). These results can help students select academic programs that will give them broad skills and job qualifications, and can help institutions confirm that programs designed to be multi-disciplinary are in fact producing students who qualify (i.e., are able to obtain interviews) for various types of jobs.
- By examining connections to immediate neighbours in the program graph, we identified a number of programs where there were no jobs that only interviewed students from that particular program. That is, students from that program always "competed" with students from other programs. The university may use these results to attract more employers that offer jobs to these programs.
- We found that the program graph computed only from senior students' interviews was slightly different: the clusters obtained from this graph were similar to those obtained by using all the data, but the nodes generally had lower fan-out. This suggests that employers offering senior-level jobs tend to interview senior students from fewer distinct departments. In particular, mostly Computer Science students interview for senior sofware jobs, but junior students from several other programs routinely interview for junior-level software jobs. Our findings can help the university decide whether to seek out employers offering junior or senior level jobs in various categories.

To summarize, characterizing competition and relationships among academic programs can help employers broaden their search for qualified students, can help students plan their academic and employment careers, can help students and institutions identify multi-disciplinary programs, and can help institutions decide which types of employers to recruit to the co-op system.

In this thesis, we took a step towards improving co-operative education. There is much more data-driven work that can be done in this space. In particular, the graph may be segmented differently in order to analyze other effects, such as gender. We are also interested in analyzing other co-op related graphs, such as those linking individual students or job postings. We also plan to develop a job recommender system for students and a student recommender system for employers based on the proposed methodology.

APPENDICES

Appendix A

Python Code for Generating Near-Cliques and Multi-Disciplinary Programs

```python
import networkx as nx
import networkx.algorithms as algo
import pandas as pd
from itertools import *
from operator import itemgetter
import csv
import statistics
import scipy.stats

#Create graph
DG=nx.DiGraph()

#Add nodes
NodeData=pd.read_csv('.\Full (Nodes).csv')
Nodedata = NodeData['ID']
DG.add_nodes_from(Nodedata)

#Add edges
EdgeData=pd.read_csv('.\Graph_EdgeTable_Directed.csv')
Edgedata = [(EdgeData['SID'][i],EdgeData['TID'][i],EdgeData['WEIGHT'][i]) for i in range(0,len(EdgeData))]
DG.add_weighted_edges_from(Edgedata)

#Conver to Undirected Graph
UG = nx.Graph(DG)

#Subgraph filter by edge weight
def createsubgraph(graph,w):
    SG=nx.DiGraph([(u,v,d) for (u,v,d) in graph.edges(data=True) if d['weight'] >=w])
    return SG

#Print programs for given IDs.
def getprogramnames(prgmid,data):
    prgmname=[data['PRGM'][i] for i in prgmid]
    return prgmname

#Create subgraphs
SG=createsubgraph(DG,0.318)
USG = nx.Graph(SG)
```

Figure A.1: Python code for graph set up

```
######################Similar Programs#######################
#find nearcliques which density is greater than 0.80
def findnearcliques(graph):
    nearcliques=[]
    #sort all the nodes by degree in descending order
    NodebyDegree=sorted(graph.degree_iter(),key=itemgetter(1),reverse=True)
    #all nodes with degree more than 3
    NodebyDegree3=[n for n in NodebyDegree if n[1]>=2]

    #iterate all subsets from size= Max degree to size=3
    for j in range(NodebyDegree[0][1]+1,3-1,-1):
        #iterate all nodes with no less than 3 neighbors
        for i in range(0, len(NodebyDegree3)):
            if NodebyDegree3[i][1]<j-1:
                break
            #iterate all neighbors
            neighbor=graph.neighbors(NodebyDegree3[i][0])
            #add base node
            neighbor.append(NodebyDegree3[i][0])
            for k in list(combinations(neighbor,j)):
                done=True
                #check if k belongs to any subsets of near cliques,
                #then go to next k
                if len(nearcliques)!=0:
                    for l in nearcliques:
                        if set(k).issubset(l):
                            done=False
                            break
                    if done==False:
                        continue
                subgraph=graph.subgraph(k)
                if nx.density(subgraph)>=0.8:
                    nearcliques.append(k)
    return nearcliques

def main_similarprogram():
#Similar Cliques: Write to textfile
with open('similarprograms_0.8.csv', 'wb') as myfile:
    wr = csv.writer(myfile, quoting=csv.QUOTE_ALL)
    wr.writerow([getprogramnames(i,NodeData) for i in findnearcliques(USG)])
```

Figure A.2: Python code for finding near-cliques

```
#####################Hetereogenious Program Function######################
def findheterprogram(graph,PartitionResults,res):
    hetercount={}
    for i in graph.nodes():
        #create array for summing out-going edge weights for each cluster
        clustercount=[0]*len(list(set(PartitionResults[res])))
        neighbor=graph.neighbors(i)
        for j in neighbor:
            clustercount[PartitionResults[res][j]]+=graph[i][j]['weight']
        hetercount[i]=clustercount
    return hetercount

def main_heterprogram():
    PartitionFile='.\Modularity\Partition_Results_SR_byNode.csv'
    PartitionResults=pd.read_csv(PartitionFile)

    resolution=[0.6,0.1]

    #test different resolution limits
    for res in resolution:
        heterprogram=findheterprogram(DG,PartitionResults,str(res))
        #Heter Programs: Write to textfile
        with open('Heterprograms_'+str(res)+'.csv', 'wb') as myfile:
            wr = csv.writer(myfile, quoting=csv.QUOTE_ALL)
            wr.writerow(['Prgm','Neighbor','Variance','Entropy']+ \
            list(set(PartitionResults[str(res)])))
            for i in heterprogram:
                prgm=getprogramnames([i],NodeData)
                wr.writerow(prgm+[len(DG.neighbors(i)),statistics.variance(heterprogram[i]),
                scipy.stats.entropy(heterprogram[i])]+heterprogram[i])
```

Figure A.3: Python code for measuring multi-disciplinary programs

Appendix B

Reduced Program Graph

Table B.1: Top 5 programs with the largest out-going edge weights for all programs (Part 1)

Base Program	Competing Program	Weight	Interview in Common	Rank
Accounting & Financial Mgt	Math/Fin.Analysis & Risk Mgm't	0.3253	54	1
	Math/Chartered Fin. Analyst	0.2349	39	2
	Economics	0.1988	33	3
	Mathematics/Business Admin	0.1566	26	4
	Actuarial Science/Finance Opt	0.1506	25	5
Actuarial Science	Actuarial Science/Finance Opt	0.5259	61	1
	Mathematics	0.3448	40	2
	Math/Fin.Analysis & Risk Mgm't	0.25	29	3
	Statistics	0.1724	20	4
	Math/Chartered Fin. Analyst	0.1724	20	5
Actuarial Science/Finance Opt	Actuarial Science	0.372	61	1
	Math/Fin.Analysis & Risk Mgm't	0.3171	52	2
	Mathematics	0.2927	48	3
	Math/Chartered Fin. Analyst	0.2195	36	4
	Mathematics/Business Admin	0.1768	29	5
Anthropology	Environment & Business	0.5	3	1
	Sociology	0.3333	2	2
	Speech Communication	0.3333	2	3
	Psychology	0.3333	2	4
	Mathematical Economics	0.1667	1	5
Applied Mathematics	Computer Science	0.6	9	1
	Computer Engineering	0.2667	4	2
	Mathematics	0.2667	4	3
	Mathematical Physics	0.2	3	4
	Mechatronics Engineering	0.2	3	5
Art Studio Practice	Legal Studies	0.4286	9	1
	Planning	0.3333	7	2
	English Rhetoric &Prof Writing	0.2857	6	3
	Psychology	0.2857	6	4
	Music	0.2381	5	5
Biochemistry	Biology	0.4615	24	1
	Chemical Engineering	0.3846	20	2
	Chemistry	0.3462	18	3
	Health Studies	0.1731	9	4
	Nanotechnology Engineering	0.1346	7	5
Biology	Biochemistry	0.2637	24	1
	Environment & Business	0.2308	21	2
	Chemical Engineering	0.1978	18	3
	Psychology	0.1758	16	4
	Mathematics	0.1758	16	5
Business & CS (Double Degree)	Computer Science	0.7914	129	1
	Software Engineering	0.5644	92	2
	Computer Engineering	0.4969	81	3
	Systems Design Engineering	0.3129	51	4
	Electrical Engineering	0.2699	44	5
Business & Math(Double Degree)	Mathematics	0.4167	50	1
	Math/Fin.Analysis & Risk Mgm't	0.4	48	2
	Computer Science	0.2417	29	3
	Mathematics/Business Admin	0.1917	23	4
	Accounting & Financial Mgt	0.175	21	5
Chemical Engineering	Mechanical Engineering	0.2881	68	1
	Nanotechnology Engineering	0.2119	50	2
	Management Engineering	0.1525	36	3
	Mathematics	0.1271	30	4
	Civil Engineering	0.1144	27	5
Chemistry	Biochemistry	0.6923	18	1
	Chemical Engineering	0.5769	15	2
	Biology	0.3462	9	3
	Nanotechnology Engineering	0.2308	6	4
	Medicinal Chemistry	0.1154	3	5
Civil Engineering	Environmental Engineering	0.25	51	1
	Mechanical Engineering	0.2304	47	2
	Geological Engineering	0.1814	37	3
	Chemical Engineering	0.1324	27	4
	Management Engineering	0.1324	27	5
Combinatorics & Optimization	Computer Science	0.5625	9	1
	Electrical Engineering	0.4375	7	2
	Software Engineering	0.375	6	3
	Systems Design Engineering	0.375	6	4
	Math/Fin.Analysis & Risk Mgm't	0.375	6	5
Comp Sci/Digital Hardware Opt	Computer Science	0.8947	17	1
	Software Engineering	0.5789	11	2
	Computer Engineering	0.5789	11	3
	Business & CS (Double Degree)	0.3684	7	4
	Mechatronics Engineering	0.3158	6	5
Comp Sci/Software Eng Option	Computer Science	0.7188	23	1
	Software Engineering	0.5313	17	2
	Computer Engineering	0.4688	15	3
	Electrical Engineering	0.3125	10	4
	Mechatronics Engineering	0.2188	7	5
Computational Mathematics	Computer Science	0.8261	19	1
	Software Engineering	0.4783	11	2
	Electrical Engineering	0.4348	10	3
	Computer Engineering	0.3913	9	4
	Mathematics	0.3478	8	5
Computer Engineering	Computer Science	0.7015	416	1
	Software Engineering	0.5076	301	2
	Electrical Engineering	0.3592	213	3
	Mechatronics Engineering	0.3255	193	4
	Systems Design Engineering	0.317	188	5
Computer Science	Computer Engineering	0.4848	416	1
	Software Engineering	0.4814	413	2
	Systems Design Engineering	0.2867	246	3
	Mechatronics Engineering	0.2855	245	4
	Electrical Engineering	0.2727	234	5
Computer Science - Bioinf Opti	Computer Science	0.7568	28	1
	Software Engineering	0.6486	24	2
	Computer Engineering	0.6216	23	3
	Business & CS (Double Degree)	0.3243	12	4
	Electrical Engineering	0.2973	11	5

Table B.2: Top 5 programs with the largest out-going edge weights for all programs (Part 2)

Base Program	Competing Program	Weight	Interview in Common	Rank
	Computer Science	0.766	72	1
	Computer Engineering	0.5213	49	2
Computer Science/Business Opt	Software Engineering	0.4894	46	3
	Mechatronics Engineering	0.3936	37	4
	Systems Design Engineering	0.383	36	5
	Computer Science	0.5045	56	1
	Computer Engineering	0.3874	43	2
Computing & Financial Mgt	Software Engineering	0.3063	34	3
	Math/Fin.Analysis & Risk Mgm't	0.2793	31	4
	Systems Design Engineering	0.2613	29	5
	Speech Communication	0.5	2	1
	Planning	0.5	2	2
Drama	Psychology	0.5	2	3
	Mathematics	0.5	2	4
	Social Development Studies	0.25	1	5
	Environmental Engineering	0.5556	5	1
	Environmental Sciences/Ecology	0.4444	4	2
Earth Sci/Hydrogeology Spe	Geological Engineering	0.3333	3	3
	Civil Engineering	0.3333	3	4
	Biology	0.3333	3	5
	Environmental Engineering	0.75	3	1
	Earth Sci/Hydrogeology Spe	0.5	2	2
Earth Sciences/Geology Spec'n	Geological Engineering	0.5	2	3
	Environmental Sciences/Ecology	0.5	2	4
	Civil Engineering	0.5	2	5
	Math/Fin.Analysis & Risk Mgm't	0.403	54	1
	Environment & Business	0.3209	43	2
Economics	Mathematics	0.291	39	3
	Accounting & Financial Mgt	0.2463	33	4
	Mathematics/Business Admin	0.2239	30	5
	Computer Science	0.4766	234	1
	Computer Engineering	0.4338	213	2
Electrical Engineering	Mechatronics Engineering	0.3544	174	3
	Software Engineering	0.3381	166	4
	Systems Design Engineering	0.2933	144	5
	Systems Design Engineering	0.45	9	1
	Computer Science	0.45	9	2
Engineering	Civil Engineering	0.4	8	3
	Computer Engineering	0.4	8	4
	Mechatronics Engineering	0.25	5	5
	English-Literature & Rhetoric	0.5	4	1
	Political Science	0.375	3	2
English Literary Studies	Planning	0.375	3	3
	Psychology	0.375	3	4
	Psychology (Science)	0.25	2	5
	English Rhetoric &Prof Writing	0.3704	10	1
	Speech Communication	0.3333	9	2
English Literature	Mathematics	0.3333	9	3
	Psychology	0.2963	8	4
	Environment & Business	0.2222	6	5
	English Rhetoric &Prof Writing	0.3125	5	1
	Legal Studies	0.3125	5	2
English Rhetoric &Comm Design	English Literature	0.25	4	3
	Music	0.1875	3	4
	History	0.1875	3	5
	Psychology	0.359	14	1
	Legal Studies	0.3077	12	2
English Rhetoric &Prof Writing	Environment & Business	0.2821	11	3
	English Literature	0.2564	10	4
	Speech Communication	0.2564	10	5
	Planning	0.2941	5	1
	Psychology	0.2941	5	2
English-ExpDigital Media Coop	English Literature	0.2353	4	3
	Political Science	0.2353	4	4
	Speech Communication	0.2353	4	5
	English Rhetoric &Prof Writing	0.5556	5	1
	English Literary Studies	0.4444	4	2
English-Literature & Rhetoric	Legal Studies	0.4444	4	3
	Psychology	0.4444	4	4
	Political Science	0.3333	3	5
	Environmental Engineering	0.6667	4	1
	Planning	0.6667	4	2
Env. Science - Geoscience Spec	Chemical Engineering	0.3333	2	3
	Environment & Business	0.3333	2	4
	Earth Sci/Hydrogeology Spe	0.1667	1	5
	Economics	0.2205	43	1
	Environment & Resource Studies	0.2154	42	2
Environment & Business	Planning	0.2051	40	3
	Mathematics	0.1744	34	4
	Math/Fin.Analysis & Risk Mgm't	0.1641	32	5
	Environment & Business	0.4828	42	1
	Geography & Environmental Mgmt	0.3448	30	2
Environment & Resource Studies	Planning	0.3448	30	3
	Environmental Engineering	0.1954	17	4
	Environmental Sciences/Ecology	0.1609	14	5
	Civil Engineering	0.505	51	1
	Geological Engineering	0.2475	25	2
Environmental Engineering	Chemical Engineering	0.2376	24	3
	Environment & Business	0.198	20	4
	Mechanical Engineering	0.1881	19	5
	Environment & Resource Studies	0.6087	14	1
	Biology	0.4348	10	2
Environmental Sciences/Ecology	Geography & Environmental Mgmt	0.3913	9	3
	Planning	0.3478	8	4
	Environment & Business	0.2609	6	5
	English Literature	0.4286	3	1
	Speech Communication	0.4286	3	2
Fine Arts	Computer Science	0.4286	3	3
	English-ExpDigital Media Coop	0.2857	2	4
	Art Studio Practice	0.2857	2	5

Table B.3: Top 5 programs with the largest out-going edge weights for all programs (Part 3)

Base Program	Competing Program	Weight	Interview in Common	Rank
French	Science & Business	0.4	2	1
	Legal Studies	0.4	2	2
	Psychology	0.4	2	3
	Peace and Conflict Studies AB	0.2	1	4
	Mathematical Studies Bus. Spe	0.2	1	5
Geography & Environmental Mgmt	Planning	0.3483	31	1
	Environment & Resource Studies	0.3371	30	2
	Environment & Business	0.3034	27	3
	Geomatics	0.1798	16	4
	Environmental Engineering	0.1236	11	5
Geological Engineering	Civil Engineering	0.6981	37	1
	Environmental Engineering	0.4717	25	2
	Mechanical Engineering	0.1887	10	3
	Chemical Engineering	0.1509	8	4
	Management Engineering	0.0755	4	5
Geomatics	Geography & Environmental Mgmt	0.6154	16	1
	Planning	0.2692	7	2
	Environment & Resource Studies	0.1538	4	3
	Electrical Engineering	0.1154	3	4
	Software Engineering	0.1154	3	5
Health Studies	Kinesiology	0.3735	31	1
	Psychology	0.2048	17	2
	Therapeutic Recreation	0.1325	11	3
	Mathematics	0.1205	10	4
	Biochemistry	0.1084	9	5
History	Political Science	0.4762	10	1
	Psychology	0.3333	7	2
	English Rhetoric &Prof Writing	0.2857	6	3
	Legal Studies	0.2857	6	4
	Economics	0.2857	6	5
Information Technology Mgmt	Computer Science	0.4286	9	1
	Computer Engineering	0.2857	6	2
	Systems Design Engineering	0.2857	6	3
	Management Engineering	0.2381	5	4
	Economics	0.2381	5	5
Kinesiology	Health Studies	0.3827	31	1
	Psychology	0.2469	20	2
	Biology	0.1481	12	3
	Environment & Business	0.1358	11	4
	Mathematics	0.1358	11	5
Legal Studies	Environment & Business	0.3485	23	1
	Psychology	0.3333	22	2
	Speech Communication	0.2273	15	3
	Mathematics	0.2273	15	4
	Economics	0.2121	14	5
Management Engineering	Computer Science	0.2519	67	1
	Mechanical Engineering	0.2406	64	2
	Systems Design Engineering	0.2331	62	3
	Electrical Engineering	0.218	58	4
	Computer Engineering	0.1955	52	5
Management Sciences	Math/Fin.Analysis & Risk Mgm't	0.3	12	1
	Mathematics/Business Admin	0.275	11	2
	Systems Design Engineering	0.275	11	3
	Management Engineering	0.25	10	4
	Computer Science	0.25	10	5
Math/CA Finance Opt	Accounting & Financial Mgt	0.5789	11	1
	Math/Chartered Fin. Analyst	0.3684	7	2
	Computing & Financial Mgt	0.2632	5	3
	Mathematical Finance	0.2105	4	4
	Actuarial Science	0.2105	4	5
Math/Chartered Fin. Analyst	Math/Fin.Analysis & Risk Mgm't	0.4902	50	1
	Accounting & Financial Mgt	0.3824	39	2
	Actuarial Science/Finance Opt	0.3529	36	3
	Mathematics/Business Admin	0.2255	23	4
	Actuarial Science	0.1961	20	5
Math/Fin.Analysis & Risk Mgm't	Mathematics	0.3566	87	1
	Mathematics/Business Admin	0.2377	58	2
	Accounting & Financial Mgt	0.2213	54	3
	Economics	0.2213	54	4
	Actuarial Science/Finance Opt	0.2131	52	5
Math/Professional Risk Mgm't	Math/Fin.Analysis & Risk Mgm't	0.6383	30	1
	Math/Chartered Fin. Analyst	0.383	18	2
	Actuarial Science/Finance Opt	0.383	18	3
	Actuarial Science	0.2766	13	4
	Accounting & Financial Mgt	0.234	11	5
Mathematical Economics	Economics	0.3636	4	1
	Mathematics/Business Admin	0.2727	3	2
	Environment & Business	0.2727	3	3
	Computer Science	0.2727	3	4
	Mathematics	0.2727	3	5
Mathematical Finance	Math/Fin.Analysis & Risk Mgm't	0.3774	20	1
	Actuarial Science/Finance Opt	0.3396	18	2
	Actuarial Science	0.3019	16	3
	Mathematics/Business Admin	0.2642	14	4
	Math/Chartered Fin. Analyst	0.2264	12	5
Mathematical Physics	Mathematics	0.4444	4	1
	Applied Mathematics	0.3333	3	2
	Nanotechnology Engineering	0.3333	3	3
	Biology	0.3333	3	4
	Computer Science	0.3333	3	5
Mathematical Studies	Computer Science	0.8	16	1
	Computer Engineering	0.45	9	2
	Software Engineering	0.4	8	3
	Mechatronics Engineering	0.3	6	4
	Mathematics	0.3	6	5
Mathematical Studies Bus. Spe	Computer Science	0.3846	5	1
	Mathematics	0.3846	5	2
	Management Engineering	0.3077	4	3
	Mathematics/Business Admin	0.3077	4	4
	Electrical Engineering	0.3077	4	5

Table B.4: Top 5 programs with the largest out-going edge weights for all programs (Part 4)

Base Program	Competing Program	Weight	Interview in Common	Rank
Mathematics	Computer Science	0.4229	148	1
	Computer Engineering	0.2486	87	2
	Math/Fin.Analysis & Risk Mgm't	0.2486	87	3
	Software Engineering	0.2257	79	4
	Systems Design Engineering	0.1571	55	5
Mathematics/Business Admin	Math/Fin.Analysis & Risk Mgm't	0.4361	58	1
	Mathematics	0.406	54	2
	Environment & Business	0.2331	31	3
	Economics	0.2256	30	4
	Actuarial Science/Finance Opt	0.218	29	5
Mechanical Engineering	Mechatronics Engineering	0.3046	106	1
	Chemical Engineering	0.1954	68	2
	Electrical Engineering	0.1954	68	3
	Management Engineering	0.1839	64	4
	Civil Engineering	0.1351	47	5
Mechatronics Engineering	Computer Science	0.4871	245	1
	Computer Engineering	0.3837	193	2
	Electrical Engineering	0.3459	174	3
	Systems Design Engineering	0.3419	172	4
	Software Engineering	0.336	169	5
Medicinal Chemistry	Chemistry	0.75	3	1
	Biochemistry	0.75	3	2
	Chemical Engineering	0.5	2	3
	Environmental Sciences/Ecology	0.25	1	4
	Science & Business/Biotech	0.25	1	5
Music	Art Studio Practice	0.7143	5	1
	English Rhetoric &Comm Design	0.4286	3	2
	English Literature	0.4286	3	3
	English Rhetoric &Prof Writing	0.4286	3	4
	Environment & Business	0.4286	3	5
Nanotechnology Engineering	Chemical Engineering	0.365	50	1
	Electrical Engineering	0.292	40	2
	Mechanical Engineering	0.2701	37	3
	Computer Science	0.2555	35	4
	Computer Engineering	0.2336	32	5
Peace and Conflict Studies AB	Psychology	0.5714	4	1
	Legal Studies	0.4286	3	2
	Therapeutic Recreation	0.2857	2	3
	Biology	0.2857	2	4
	French	0.1429	1	5
Physics	Electrical Engineering	0.3636	8	1
	Systems Design Engineering	0.3636	8	2
	Computer Science	0.3182	7	3
	Mathematics	0.3182	7	4
	Mechatronics Engineering	0.2273	5	5
Planning	Environment & Business	0.25	40	1
	Geography & Environmental Mgmt	0.1938	31	2
	Environment & Resource Studies	0.1875	30	3
	Psychology	0.1563	25	4
	Economics	0.125	20	5
Political Science	Psychology	0.4468	21	1
	Planning	0.4043	19	2
	Economics	0.2553	12	3
	Mathematics	0.2553	12	4
	Legal Studies	0.234	11	5
Psychology	Mathematics	0.2393	28	1
	Environment & Business	0.2222	26	2
	Planning	0.2137	25	3
	Legal Studies	0.188	22	4
	Political Science	0.1795	21	5
Psychology (Science)	Psychology	0.5263	10	1
	Planning	0.3684	7	2
	Kinesiology	0.3158	6	3
	Therapeutic Recreation	0.2105	4	4
	Political Science	0.2105	4	5
Public Service Co-op	Economics	0.3	21	1
	Planning	0.2714	19	2
	Environment & Business	0.1857	13	3
	Environment & Resource Studies	0.1429	10	4
	Math/Fin.Analysis & Risk Mgm't	0.1429	10	5
Pure Mathematics	Actuarial Science	0.6667	8	1
	Actuarial Science/Finance Opt	0.5	6	2
	Math/Fin.Analysis & Risk Mgm't	0.5	6	3
	Math/Chartered Fin. Analyst	0.4167	5	4
	Computer Engineering	0.4167	5	5
Rec & Leis St/Business Opt	Psychology	0.5	4	1
	Therapeutic Recreation	0.375	3	2
	Recreation and Sport Business	0.375	3	3
	Geography & Environmental Mgmt	0.375	3	4
	Health Studies	0.375	3	5
Recreation & Leisure Studies	Kinesiology	0.5	8	1
	Environment & Business	0.4375	7	2
	Psychology	0.375	6	3
	Health Studies	0.3125	5	4
	Recreation and Sport Business	0.25	4	5
Recreation and Sport Business	Psychology	0.5	10	1
	Environment & Business	0.4	8	2
	Kinesiology	0.35	7	3
	Geography & Environmental Mgmt	0.3	6	4
	Speech Communication	0.3	6	5
Science & Business	Environment & Business	0.3684	21	1
	Math/Fin.Analysis & Risk Mgm't	0.2632	15	2
	Economics	0.2456	14	3
	Mathematics	0.2456	14	4
	Psychology	0.2281	13	5
Science & Business/Biochem	Math/Fin.Analysis & Risk Mgm't	0.4	4	1
	Science & Business/Biotech	0.3	3	2
	Math/Chartered Fin. Analyst	0.3	3	3
	Science & Business	0.3	3	4
	Accounting & Financial Mgt	0.3	3	5

Table B.5: Top 5 programs with the largest out-going edge weights for all programs (Part 5)

Base Program	Competing Program	Weight	Interview in Common	Rank
Science & Business/Biology	Environment & Business	0.4286	12	1
	Economics	0.2857	8	2
	Math/Fin.Analysis & Risk Mgm't	0.2857	8	3
	Science & Business/Biotech	0.25	7	4
	Legal Studies	0.25	7	5
Science & Business/Biotech	Environment & Business	0.4151	22	1
	Mathematics/Business Admin	0.2642	14	2
	Economics	0.2642	14	3
	Mathematics	0.2642	14	4
	Math/Fin.Analysis & Risk Mgm't	0.2453	13	5
Science & Business/Chemistry	Environment & Business	0.6	6	1
	Economics	0.4	4	2
	Science & Business/Biotech	0.3	3	3
	Science & Business	0.3	3	4
	Accounting & Financial Mgt	0.3	3	5
Science & Business/Env Science	Economics	0.6667	4	1
	Psychology	0.5	3	2
	Science & Business/Biochem	0.3333	2	3
	Science & Business/Biology	0.3333	2	4
	Science-Biotechnology/Econom	0.3333	2	5
Science-Biotechnology/Econom	Economics	0.3214	9	1
	Environment & Business	0.2857	8	2
	Accounting & Financial Mgt	0.25	7	3
	Management Engineering	0.2143	6	4
	Systems Design Engineering	0.2143	6	5
Social Development Studies	Psychology	0.6	3	1
	Kinesiology	0.4	2	2
	Health Studies	0.4	2	3
	Legal Studies	0.4	2	4
	Planning	0.4	2	5
Sociology	Psychology	0.3793	11	1
	Environment & Business	0.3103	9	2
	Legal Studies	0.2759	8	3
	Planning	0.2414	7	4
	Science & Business/Biology	0.2069	6	5
Software Engineering	Computer Science	0.7468	413	1
	Computer Engineering	0.5443	301	2
	Mechatronics Engineering	0.3056	169	3
	Electrical Engineering	0.3002	166	4
	Systems Design Engineering	0.2911	161	5
Speech Communication	Environment & Business	0.4531	29	1
	Psychology	0.2969	19	2
	Mathematics	0.2813	18	3
	Economics	0.2656	17	4
	Legal Studies	0.2344	15	5
Statistics	Actuarial Science/Finance Opt	0.5106	24	1
	Actuarial Science	0.4255	20	2
	Mathematics	0.3617	17	3
	Math/Fin.Analysis & Risk Mgm't	0.234	11	4
	Mathematics/Business Admin	0.2128	10	5
Statistics for Health	Mathematics	0.4815	13	1
	Math/Fin.Analysis & Risk Mgm't	0.3704	10	2
	Actuarial Science/Finance Opt	0.3333	9	3
	Actuarial Science	0.2963	8	4
	Statistics	0.2593	7	5
Systems Design Engineering	Computer Science	0.5503	246	1
	Computer Engineering	0.4206	188	2
	Mechatronics Engineering	0.3848	172	3
	Software Engineering	0.3602	161	4
	Electrical Engineering	0.3221	144	5
Therapeutic Recreation	Health Studies	0.4783	11	1
	Psychology	0.4783	11	2
	Kinesiology	0.3043	7	3
	Psychology (Science)	0.1739	4	4
	Recreation and Sport Business	0.1739	4	5

Appendix C

Detailed Results

C.1 Full Program Graph: Results of Near-Cliques

Table C.1: Results of near-cliques in the full program graph (Part 1)

1	Mechatronics Engineering	Mathematical Studies	Systems Design Engineering	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
2	Mechatronics Engineering	Systems Design Engineering	Business & CS (Double Degree)	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
3	Mechatronics Engineering	Systems Design Engineering	Combinatorics & Optimization	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
4	Mechatronics Engineering	Systems Design Engineering	Comp Sci/Digital Hardware Opt	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
5	Mechatronics Engineering	Systems Design Engineering	Comp Sci/Software Eng Option	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
6	Mechatronics Engineering	Systems Design Engineering	Computational Mathematics	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
7	Mechatronics Engineering	Systems Design Engineering	Computer Engineering	Computer Science - Bioinf Opti	Computer Science/Business Opt	Software Engineering	Computer Science
8	Mechatronics Engineering	Systems Design Engineering	Computer Engineering	Computer Science/Business Opt	Software Engineering	Electrical Engineering	Computer Science
9	Mechatronics Engineering	Systems Design Engineering	Computer Engineering	Computer Science/Business Opt	Software Engineering	Engineering	Computer Science
10	Systems Design Engineering	Business & CS (Double Degree)	Comp Sci/Digital Hardware Opt	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
11	Mechatronics Engineering	Systems Design Engineering	Combinatorics & Optimization	Computer Engineering	Electrical Engineering	Computer Science	
12	Mechatronics Engineering	Systems Design Engineering	Computational Mathematics	Computer Engineering	Electrical Engineering	Computer Science	
13	Mechatronics Engineering	Systems Design Engineering	Computer Engineering	Computer Science/Business Opt	Computing & Financial Mgt	Computer Science	
14	Mechatronics Engineering	Systems Design Engineering	Computer Engineering	Electrical Engineering	Engineering	Computer Science	
15	Mechatronics Engineering	Business & CS (Double Degree)	Comp Sci/Digital Hardware Opt	Computer Engineering	Software Engineering	Computer Science	
16	Mechatronics Engineering	Computational Mathematics	Computer Engineering	Computer Science/Business Opt	Electrical Engineering	Computer Science	
17	Mechatronics Engineering	Computational Mathematics	Computer Engineering	Software Engineering	Electrical Engineering	Computer Science	
18	Mathematical Studies	Systems Design Engineering	Business & CS (Double Degree)	Computer Engineering	Software Engineering	Computer Science	
19	Mathematical Studies	Systems Design Engineering	Combinatorics & Optimization	Computer Engineering	Software Engineering	Computer Science	
20	Mathematical Studies	Systems Design Engineering	Comp Sci/Digital Hardware Opt	Computer Engineering	Software Engineering	Computer Science	
21	Mathematical Studies	Systems Design Engineering	Comp Sci/Software Eng Option	Computer Engineering	Software Engineering	Computer Science	
22	Mathematical Studies	Systems Design Engineering	Computational Mathematics	Computer Engineering	Software Engineering	Computer Science	
23	Mathematical Studies	Systems Design Engineering	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science	
24	Mathematical Studies	Systems Design Engineering	Computer Engineering	Software Engineering	Engineering	Computer Science	
25	Mathematical Studies	Business & CS (Double Degree)	Comp Sci/Digital Hardware Opt	Computer Engineering	Software Engineering	Computer Science	
26	Mathematical Studies	Business & CS (Double Degree)	Comp Sci/Software Eng Option	Computer Engineering	Software Engineering	Computer Science	
27	Mathematical Studies	Business & CS (Double Degree)	Computational Mathematics	Computer Engineering	Software Engineering	Computer Science	
28	Mathematical Studies	Business & CS (Double Degree)	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science	
29	Mathematical Studies	Business & CS (Double Degree)	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science	
30	Mathematical Studies	Comp Sci/Digital Hardware Opt	Comp Sci/Software Eng Option	Computer Engineering	Software Engineering	Computer Science	
31	Mathematical Studies	Comp Sci/Digital Hardware Opt	Computational Mathematics	Computer Engineering	Software Engineering	Computer Science	
32	Mathematical Studies	Comp Sci/Digital Hardware Opt	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science	
33	Mathematical Studies	Comp Sci/Digital Hardware Opt	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science	
34	Mathematical Studies	Comp Sci/Software Eng Option	Computational Mathematics	Computer Engineering	Software Engineering	Computer Science	
35	Mathematical Studies	Comp Sci/Software Eng Option	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science	
36	Mathematical Studies	Comp Sci/Software Eng Option	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science	
37	Mathematical Studies	Computational Mathematics	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science	
38	Mathematical Studies	Computational Mathematics	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science	
39	Mathematical Studies	Computational Mathematics	Computer Engineering	Software Engineering	Electrical Engineering	Computer Science	
40	Mathematical Studies	Computer Engineering	Computer Science - Bioinf Opti	Computer Science/Business Opt	Software Engineering	Computer Science	

Table C.2: Results of near-cliques in the full program graph (Part 2)

41	Systems Design Engineering	Business & CS (Double Degree)	Combinatorics & Optimization	Computer Engineering	Software Engineering	Computer Science
42	Systems Design Engineering	Business & CS (Double Degree)	Comp Sci/Software Eng Option	Computer Engineering	Software Engineering	Computer Science
43	Systems Design Engineering	Business & CS (Double Degree)	Computational Mathematics	Computer Engineering	Software Engineering	Computer Science
44	Systems Design Engineering	Business & CS (Double Degree)	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science
45	Systems Design Engineering	Business & CS (Double Degree)	Computer Engineering	Software Engineering	Engineering	Computer Science
46	Systems Design Engineering	Combinatorics & Optimization	Comp Sci/Digital Hardware Opt	Computer Engineering	Software Engineering	Computer Science
47	Systems Design Engineering	Combinatorics & Optimization	Comp Sci/Software Eng Option	Computer Engineering	Software Engineering	Computer Science
48	Systems Design Engineering	Combinatorics & Optimization	Computational Mathematics	Computer Engineering	Software Engineering	Computer Science
49	Systems Design Engineering	Combinatorics & Optimization	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science
50	Systems Design Engineering	Combinatorics & Optimization	Computer Engineering	Software Engineering	Electrical Engineering	Computer Science
51	Systems Design Engineering	Combinatorics & Optimization	Computer Engineering	Software Engineering	Engineering	Computer Science
52	Systems Design Engineering	Comp Sci/Digital Hardware Opt	Comp Sci/Software Eng Option	Computer Engineering	Software Engineering	Computer Science
53	Systems Design Engineering	Comp Sci/Digital Hardware Opt	Computational Mathematics	Computer Engineering	Software Engineering	Computer Science
54	Systems Design Engineering	Comp Sci/Digital Hardware Opt	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science
55	Systems Design Engineering	Comp Sci/Digital Hardware Opt	Computer Engineering	Software Engineering	Engineering	Computer Science
56	Systems Design Engineering	Comp Sci/Software Eng Option	Computational Mathematics	Computer Engineering	Software Engineering	Computer Science
57	Systems Design Engineering	Comp Sci/Software Eng Option	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science
58	Systems Design Engineering	Comp Sci/Software Eng Option	Computer Engineering	Software Engineering	Engineering	Computer Science
59	Systems Design Engineering	Computational Mathematics	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science
60	Systems Design Engineering	Computational Mathematics	Computer Engineering	Software Engineering	Electrical Engineering	Computer Science
61	Systems Design Engineering	Computational Mathematics	Computer Engineering	Software Engineering	Engineering	Computer Science
62	Systems Design Engineering	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Engineering	Computer Science
63	Systems Design Engineering	Computer Engineering	Computer Science/Business Opt	Computing & Financial Mgt	Software Engineering	Computer Science
64	Business & CS (Double Degree)	Combinatorics & Optimization	Comp Sci/Digital Hardware Opt	Computer Engineering	Software Engineering	Computer Science
65	Business & CS (Double Degree)	Comp Sci/Digital Hardware Opt	Comp Sci/Software Eng Option	Computer Engineering	Software Engineering	Computer Science
66	Business & CS (Double Degree)	Comp Sci/Digital Hardware Opt	Computational Mathematics	Computer Engineering	Software Engineering	Computer Science
67	Business & CS (Double Degree)	Comp Sci/Digital Hardware Opt	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science
68	Business & CS (Double Degree)	Comp Sci/Digital Hardware Opt	Computer Engineering	Computing & Financial Mgt	Software Engineering	Computer Science
69	Business & CS (Double Degree)	Comp Sci/Digital Hardware Opt	Computer Engineering	Software Engineering	Electrical Engineering	Computer Science
70	Business & CS (Double Degree)	Comp Sci/Digital Hardware Opt	Computer Engineering	Software Engineering	Engineering	Computer Science
71	Business & CS (Double Degree)	Comp Sci/Software Eng Option	Computational Mathematics	Computer Engineering	Software Engineering	Computer Science
72	Business & CS (Double Degree)	Comp Sci/Software Eng Option	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science
73	Business & CS (Double Degree)	Comp Sci/Software Eng Option	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
74	Business & CS (Double Degree)	Computational Mathematics	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science
75	Business & CS (Double Degree)	Computational Mathematics	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
76	Business & CS (Double Degree)	Computational Mathematics	Computer Engineering	Software Engineering	Electrical Engineering	Computer Science
77	Business & CS (Double Degree)	Computer Engineering	Computer Science - Bioinf Opti	Computer Science/Business Opt	Software Engineering	Computer Science
78	Combinatorics & Optimization	Computational Mathematics	Computer Engineering	Software Engineering	Electrical Engineering	Computer Science
79	Comp Sci/Digital Hardware Opt	Comp Sci/Software Eng Option	Computational Mathematics	Computer Engineering	Software Engineering	Computer Science
80	Comp Sci/Digital Hardware Opt	Comp Sci/Software Eng Option	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science
81	Comp Sci/Digital Hardware Opt	Comp Sci/Software Eng Option	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
82	Comp Sci/Digital Hardware Opt	Computational Mathematics	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science
83	Comp Sci/Digital Hardware Opt	Computational Mathematics	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
84	Comp Sci/Digital Hardware Opt	Computational Mathematics	Computer Engineering	Software Engineering	Electrical Engineering	Computer Science
85	Comp Sci/Digital Hardware Opt	Computer Engineering	Computer Science - Bioinf Opti	Computer Science/Business Opt	Software Engineering	Computer Science
86	Comp Sci/Software Eng Option	Computational Mathematics	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Computer Science
87	Comp Sci/Software Eng Option	Computational Mathematics	Computer Engineering	Computer Science/Business Opt	Software Engineering	Computer Science
88	Comp Sci/Software Eng Option	Computational Mathematics	Computer Engineering	Software Engineering	Electrical Engineering	Computer Science
89	Comp Sci/Software Eng Option	Computer Engineering	Computer Science - Bioinf Opti	Computer Science/Business Opt	Software Engineering	Computer Science
90	Computational Mathematics	Computer Engineering	Computer Science - Bioinf Opti	Computer Science/Business Opt	Software Engineering	Computer Science
91	Computational Mathematics	Computer Engineering	Computer Science - Bioinf Opti	Software Engineering	Electrical Engineering	Computer Science
92	Computational Mathematics	Computer Engineering	Computer Science/Business Opt	Software Engineering	Electrical Engineering	Computer Science
93	Mechatronics Engineering	Mathematical Studies	Computer Engineering	Electrical Engineering	Computer Science	
94	Mechatronics Engineering	Business & CS (Double Degree)	Computer Engineering	Electrical Engineering	Computer Science	
95	Mechatronics Engineering	Comp Sci/Digital Hardware Opt	Computer Engineering	Electrical Engineering	Computer Science	
96	Mechatronics Engineering	Comp Sci/Software Eng Option	Computer Engineering	Electrical Engineering	Computer Science	
97	Mechatronics Engineering	Computer Engineering	Computer Science - Bioinf Opti	Electrical Engineering	Computer Science	
98	Mechatronics Engineering	Computer Engineering	Computing & Financial Mgt	Electrical Engineering	Computer Science	
99	Mathematical Studies	Computer Engineering	Computing & Financial Mgt	Software Engineering	Computer Science	
100	Systems Design Engineering	Combinatorics & Optimization	Software Engineering	Mathematics	Computer Science	
101	Systems Design Engineering	Computer Engineering	Computing & Financial Mgt	Engineering	Computer Science	
102	Combinatorics & Optimization	Computational Mathematics	Software Engineering	Mathematics	Computer Science	
103	Combinatorics & Optimization	Computational Mathematics	Electrical Engineering	Mathematics	Computer Science	
104	Comp Sci/Software Eng Option	Computer Engineering	Computing & Financial Mgt	Software Engineering	Computer Science	
105	Computational Mathematics	Computer Engineering	Computing & Financial Mgt	Software Engineering	Computer Science	
106	Computational Mathematics	Computer Engineering	Computing & Financial Mgt	Electrical Engineering	Computer Science	
107	Computational Mathematics	Computer Engineering	Software Engineering	Mathematics	Computer Science	
108	Computational Mathematics	Computer Engineering	Electrical Engineering	Engineering	Computer Science	
109	Computational Mathematics	Computer Engineering	Electrical Engineering	Mathematics	Computer Science	
110	Computer Engineering	Computer Science - Bioinf Opti	Computing & Financial Mgt	Software Engineering	Computer Science	
111	Actuarial Science/Finance Opt	Pure Mathematics	Math/Fin.Analysis & Risk Mgmt	Math/Professional Risk Mgmt	Math/Chartered Fin. Analyst	
112	Earth Sci/Hydrogeology Spe	Geological Engineering	Civil Engineering	Environmental Engineering	Earth Sciences/Geology Specn	
113	Biology	Medicinal Chemistry	Chemical Engineering	Chemistry	Biochemistry	
114	Mathematical Studies Bus. Spe	Combinatorics & Optimization	Mathematics	Computer Science		
115	Mathematical Studies Bus. Spe	Computational Mathematics	Mathematics	Computer Science		

Table C.3: Results of near-cliques in the full program graph (Part 3)

116	Therapeutic Recreation	Rec & Leis St/Business Opt	Recreation and Sport Business	Psychology
117	English Rhetoric &Prof Writing	English-Literature & Rhetoric	English Literary Studies	Psychology
118	English-Literature & Rhetoric	Political Science	English Literary Studies	Psychology
119	Business & Math(Double Degree)	Combinatorics & Optimization	Mathematics	Math/Fin.Analysis & Risk Mgmt
120	Business & Math(Double Degree)	Statistics for Health	Mathematics	Math/Fin.Analysis & Risk Mgmt
121	Business & Math(Double Degree)	Mathematics	Mathematics/Business Admin	Math/Fin.Analysis & Risk Mgmt
122	Combinatorics & Optimization	Statistics for Health	Mathematics	Math/Fin.Analysis & Risk Mgmt
123	Combinatorics & Optimization	Mathematics	Mathematics/Business Admin	Math/Fin.Analysis & Risk Mgmt
124	Statistics for Health	Mathematics	Mathematics/Business Admin	Math/Fin.Analysis & Risk Mgmt
125	Combinatorics & Optimization	Computer Science	Math/Fin.Analysis & Risk Mgmt	Mathematics
126	Therapeutic Recreation	Psychology	Health Studies	Rec & Leis St/Business Opt
127	Environment & Business	Psychology	Recreation and Sport Business	Rec & Leis St/Business Opt
128	Actuarial Science	Pure Mathematics	Statistics	Actuarial Science/Finance Opt
129	Actuarial Science	Pure Mathematics	Math/Chartered Fin. Analyst	Actuarial Science/Finance Opt
130	Earth Sci/Hydrogeology Spe	Environmental Engineering	Environmental Sciences/Ecology	Earth Sciences/Geology Specn
131	Psychology	Planning	Political Science	English Literary Studies
132	Geography & Environmental Mgmt	Environmental Sciences/Ecology	Planning	
133	Accounting & Financial Mgt	Math/CA Finance Opt	Math/Chartered Fin. Analyst	
134	English Rhetoric &Prof Writing	English Literature	Music	
135	Kinesiology	Health Studies	Social Development Studies	

C.2 Full Program Graph: Word Clouds of Job Titles in Community Detection Results

Figure C.1: Word cloud of job titles of common jobs in the Math and Engineering cluster (2 clusters)

Figure C.2: Word cloud of job titles of common jobs in the Social and Natural Science cluster (2 clusters)

Figure C.3: Word cloud of job titles of common jobs in the Engineering and Computing cluster (4 clusters)

Figure C.4: Word cloud of job titles of common jobs in the Math and Finance cluster (4 clusters)

Figure C.5: Word cloud of job titles of common jobs in the Natural Science and Environment cluster (4 clusters)

Figure C.6: Word cloud of job titles of common jobs in the Social Science and Health cluster (4 clusters)

Figure C.7: Word cloud of job titles of common jobs in the Engineering cluster (8 clusters)

Figure C.8: Word cloud of job titles of common jobs in the Computing cluster (8 clusters)

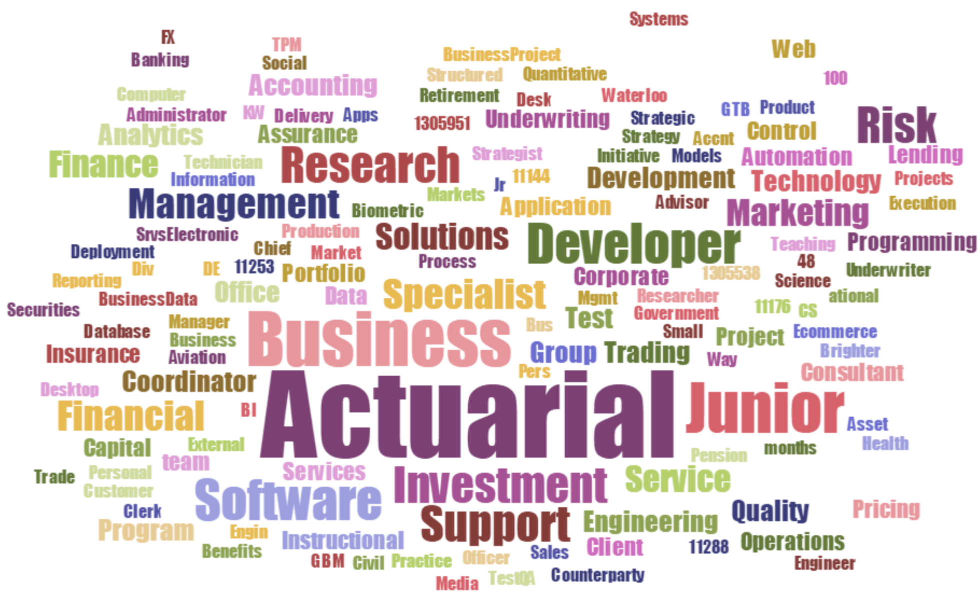


Figure C.9: Word cloud of job titles of common jobs in the Math and Finance cluster (8 clusters)

Figure C.10: Word cloud of job titles of common jobs in the Natural Science cluster (8 clusters)

Figure C.11: Word cloud of job titles of common jobs in the Environment cluster (8 clusters)

Figure C.12: Word cloud of job titles of common jobs in the Science & Business cluster (8 clusters)

Figure C.13: Word cloud of job titles of common jobs in the Social Science cluster (8 clusters)

Figure C.14: Word cloud of job titles of common jobs in the Health cluster (8 clusters)

C.3 Senior Program Graph: Results of Near-Cliques

Table C.4: Results of near-cliques in the senior program graph

1	Computer Science - Bioinf Opti	Mechatronics Engineering	Software Engineering	Systems Design Engineering	Computer Science/Business Opt	Computer Science
2	Computer Science - Bioinf Opti	Mechatronics Engineering	Software Engineering	Computer Engineering	Computer Science/Business Opt	Computer Science
3	Computer Science - Bioinf Opti	Mechatronics Engineering	Systems Design Engineering	Business & Math(Double Degree)	Computer Science/Business Opt	Computer Science
4	Computer Science - Bioinf Opti	Mechatronics Engineering	Systems Design Engineering	Computer Engineering	Computer Science/Business Opt	Computer Science
5	Computer Science - Bioinf Opti	Software Engineering	Systems Design Engineering	Computer Engineering	Computer Science/Business Opt	Computer Science
6	Math/Professional Risk Mgmt	Pure Mathematics	Math/Fin.Analysis & Risk Mgmt	Business & Math(Double Degree)	Math/Chartered Fin. Analyst	
7	Sociology	Legal Studies	English-Literature & Rhetoric	Environment & Business		
8	Science & Business/Env Science	Science & Business	Economics	Environment & Business		
9	Science & Business/Env Science	Science & Business	Science & Business/Biology	Environment & Business		
10	Science & Business/Env Science	Economics	Science & Business/Biology	Environment & Business		
11	Science & Business/Env Science	Economics	Science & Business/Chemistry	Environment & Business		
12	Science-Biotechnology/Econom	Economics	Environment & Business	Science & Business/Env Science		
13	Earth Sci/Hydrogeology Spe	Earth Sciences/Geology Specn	Biology	Environmental Sciences/Ecology		
14	Health Studies	Psychology	Therapeutic Recreation	Rec & Leis St/Business Opt		
15	Health Studies	Rec & Leis St/Business Opt	Kinesiology	Therapeutic Recreation		
16	Chemistry	Biochemistry	Chemical Engineering	Medicinal Chemistry		
17	Earth Sciences/Geology Specn	Environmental Engineering	Environmental Sciences/Ecology	Earth Sci/Hydrogeology Spe		
18	Medicinal Chemistry	Biology	Biochemistry	Chemistry		
19	Statistics	Actuarial Science/Finance Opt	Pure Mathematics	Actuarial Science		
20	Statistics	Actuarial Science/Finance Opt	Statistics for Health	Actuarial Science		
21	Actuarial Science/Finance Opt	Pure Mathematics	Statistics for Health	Actuarial Science		
22	Kinesiology	Social Development Studies	Therapeutic Recreation	Health Studies		
23	English Rhetoric &Prof Writing	English-Literature & Rhetoric	Psychology			
24	Accounting & Financial Mgt	Math/CA Finance Opt	Math/Chartered Fin. Analyst			
25	Geography & Environmental Mgmt	Environment & Resource Studies	Environmental Sciences/Ecology			

References

[1] James Abello, Mauricio GC Resende, and Sandra Sudarsky. Massive quasi-clique detection. In *LATIN 2002: Theoretical Informatics*, pages 598–612. Springer, 2002.

[2] Charu C Aggarwal and Haixun Wang. Graph data management and mining: A survey of algorithms and applications. In *Managing and Mining Graph Data*, pages 13–68. Springer, 2010.

[3] Charu C Aggarwal and Haixun Wang. A survey of clustering algorithms for graph data. In *Managing and mining graph data*, pages 275–301. Springer, 2010.

[4] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. Network flows. Technical report, DTIC Document, 1988.

[5] Bhadra JH Arachchige and Alan Robertson. Business student perceptions of a preferred employer: A study identifying determinants of employer branding. *The IUP Journal of Brand Management*, 8(3):25–46, 2011.

[6] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. Four degrees of separation. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 33–42. ACM, 2012.

[7] Nikhil Bansal, Uriel Feige, Robert Krauthgamer, Konstantin Makarychev, Viswanath Nagarajan, Joseph Seffi, and Roy Schwartz. Min-max graph partitioning and small set expansion. *SIAM Journal on Computing*, 43(2):872–904, 2014.

[8] David Barber. Clique matrices for statistical graph decomposition and parameterising restricted positive definite matrices. *eprint arXiv:1206.3237*, 2012.

[9] Ronald Barnett. Supercomplexity and the curriculum. *Studies in higher education*, 25(3):255–265, 2000.

[10] Ronald Barnett. Learning for an unknown future. *Higher Education Research & Development*, 31(1):65–77, 2012.

[11] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.

[12] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

[13] Benjamin F Blair, Meghan Millea, and Joshua Hammer. The impact of cooperative education on academic performance and compensation of engineering majors. *Journal of Engineering Education*, 93(4):333–338, 2004.

[14] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[15] Phillip Bonacich. Power and centrality: A family of measures. *American journal of sociology*, pages 1170–1182, 1987.

[16] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. *Graph theory with applications*, volume 290. Macmillan London, 1976.

[17] Stephen P Borgatti. Centrality and network flow. *Social networks*, 27(1):55–71, 2005.

[18] Maura Borrego and Jonte Bernhard. The emergence of engineering education research as an internationally connected field of inquiry. *Journal of Engineering Education*, 100(1):14–47, 2011.

[19] Craig W Bowen. A quantitative literature review of cooperative learning effects on high school and college chemistry achievement. *Journal of Chemical education*, 77(1):116, 2000.

[20] LA Braunstein and MK Loken. Benefits of cooperative education for employers. *International handbook for cooperative education*, pages 237–245, 2004.

[21] Lori A Braunstein and William A Stull. Employer benefits of, and attitudes toward post-secondary cooperative education. *Journal of Cooperative Education*, 36(1):7–22, 2001.

[22] Helen Breen and Nerilee Hing. Improving competitiveness through cooperation: Assessing the benefits of cooperative education partnerships in gaming management. *UNLV Gaming Research & Review Journal*, 6(1):5, 2012.

[23] Luiz S. Ochi C. Rodrigo Dias. Efficient evolutionary algorithms for the clustering problem in directed graphs. *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, pages 983 – 990 Vol.2, 2003.

[24] Daniel M Cable and Daniel B Turban. The value of organizational reputation in the recruitment context: A brand-equity perspective. *Journal of Applied Social Psychology*, 33(11):2244–2266, 2003.

[25] Cheryl Cates and Patricia Jones. *Learning outcomes: The educational value of cooperative education*. Cooperative Education Association Columbia, MD, 1999.

[26] Kettil Cedercreutz, Cheryl Cates, Richard Miller, Thomas Newbold, Anita Todd, Hua Zou, et al. Assessment of the effectiveness of engineering and engineering technology curricula in the context of student work performance: A quantitative approach. *Journal of Cooperative Education & Internships*, 45(2), 2011.

[27] Yiling Chen and Frederico Fonseca. A bipartite graph co-clustering approach to ontology mapping, 2004.

[28] Ken Cherven. *Network graph analysis and visualization with Gephi*. Packt Publishing Ltd, 2013.

[29] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[30] R Coll, Karsten Zegwaard, and Dave Hodges. Science and technology stakeholders ranking of graduate competencies part 2: Students perspective. *Asia-Pacific Journal of Cooperative Education*, 3(2):35–44, 2002.

[31] Richard K Coll and Richard Chapman. Evaluating service quality for cooperative education programs. *Asia-Pacific Journal of Cooperative Education*, 1(2):1–12, 2000.

[32] Richard K Coll, Karsten Zegwaard, and Dave Hodges. Science and technology stakeholders ranking of graduate competencies part 1: Employer perspective. *Asia-Pacific Journal of cooperative education*, 3(2):19–28, 2002.

[33] Christopher J Collins. The interactive effects of recruitment practices and product awareness on job seekers' employer knowledge and application behaviors. *Journal of applied psychology*, 92(1):180, 2007.

[34] Christopher J Collins and Jian Han. Exploring applicant pool quantity and quality: The effects of early recruitment practice strategies, corporate advertising, and firm reputation. *Personnel Psychology*, 57(3):685–717, 2004.

[35] Cooperative Education. Procedures for job searches in jobmine and arranging your own job. Accessed on 17 Feb 2015, at `https://uwaterloo.ca/co-operative-education/get-hired/job-search-process-and-procedures/procedures-job-searches-jobmine-and-arranging-your-own-job`.

[36] Kino Coursey and Rada Mihalcea. Topic identification using wikipedia graph centrality. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 117–120. Association for Computational Linguistics, 2009.

[37] Jason Davis. Word cloud generator. Accessed on 15 August 2015, at `https://www.jasondavies.com/wordcloud/`.

[38] Inderjit Dhillon. A fast kernel-based multilevel algorithm for graph clustering. *In Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 629–634, 2005.

[39] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM, 2001.

[40] Robert C Dickeson. *Prioritizing academic programs and services: Reallocating resources to achieve strategic balance, revised and updated*. John Wiley & Sons, 2009.

[41] Francis Donkor, Stanislaus N Nsoh, and Stephen J Mitchual. Assessment of supervised industrial attachment of a technical and vocational teacher education program in ghana. *Asia-Pacific Journal of Cooperative Education*, 10(1):1–17, 2009.

[42] Yon Dourisboure, Filippo Geraci, and Marco Pellegrini. Extraction and classification of dense communities in the web. In *Proceedings of the 16th international conference on World Wide Web*, pages 461–470. ACM, 2007.

[43] CHRIS Eames and CHERYL Cates. Theories of learning in cooperative and work-integrated education. *International handbook for cooperative and work-integrated education: International perspectives of theory, research and practice*, pages 41–52, 2011.

[44] Elaine El-Khawas. Higher education re-formed: Peter scott (ed.): Falmer press, london, 2000. *Higher Education Policy*, 14(1):93–95, 2001.

[45] Ergin Elmacioglu and Dongwon Lee. On six degrees of separation in dblp-db and more. *SIGMOD Rec.*, 34(2):33–40, June 2005.

[46] Zinaida Fadeeva, Yoko Mochizuki, Katja Brundiers, Arnim Wiek, and Charles L Redman. Real-world learning opportunities in sustainability: from classroom into the real world. *International Journal of Sustainability in Higher Education*, 11(4):308–324, 2010.

[47] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM computer communication review*, volume 29, pages 251–262. ACM, 1999.

[48] Sonia Ferns and Kerri Moore. Assessing student outcomes in fieldwork placements: An overview of current practice. *Asia-Pacific Journal of Cooperative Education*, 13(4):207–224, 2012.

[49] Gary William Flake, Steve Lawrence, and C Lee Giles. Efficient identification of web communities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160. ACM, 2000.

[50] Gary William Flake, Steve Lawrence, C Lee Giles, and Frans M Coetzee. Self-organization and identification of web communities. *Computer*, 35(3):66–70, 2002.

[51] World Association for Cooperative & Work-integrated Education. Wace global institutions. Accessed on 30 Jun 2015, at `http://www.waceinc.org/global_institutions.html`.

[52] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[53] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[54] Jack Gault, John Redington, and Tammy Schlager. Undergraduate business internships and career success: are they related? *Journal of marketing education*, 22(1):45–53, 2000.

[55] Vipin Kumar George Karypis. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20:359–392, 1998.

[56] David Gibson, Ravi Kumar, and Andrew Tomkins. Discovering large dense subgraphs in massive graphs. In *Proceedings of the 31st international conference on Very large data bases*, pages 721–732. VLDB Endowment, 2005.

[57] Anna C Gilbert and Kirill Levchenko. Compressing network graphs. In *Proceedings of the LinkKDD workshop at the 10th ACM Conference on KDD*, volume 124, 2004.

[58] Gabriel P Gimenes, Hugo Gualdron, Jose F Rodrigues Jr, and Mario Gazziro. Multimodal graph-based analysis over the dblp repository: critical discoveries and hypotheses. In *Proceedings of the ACM Symposium on Applied Computing*, pages 1–7. ACM Press, 2015.

[59] A Inokuchi H Kashima. Kernels for graph classification. *ICDM Workshop on Active Mining 2002*, 2002.

[60] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, August 2008.

[61] Martin Hayden, Ron Dowell, and Helen Saenger. Insights from an evaluation of a workplace-based course in business for tourism and hospitality students in australia. *Asia-Pacific Journal of Cooperative Education*, 2(2):1–11, 2001.

[62] Jay Hays and Mike Clements. Supervision in work experience for learning programs. In *Proceedings of the 17th World Conference on Cooperative and Work-Integrated Education (WACE)*, 2011.

[63] Anthony J Hesketh. Recruiting an elite? employers' perceptions of graduate education and training. *Journal of education and work*, 13(3):245–271, 2000.

[64] Dave Hodges and Noel Burchell. Business graduate competencies: Employers views on importance and performance. *Asia-Pacific Journal of Cooperative Education*, 4(2):16–22, 2003.

[65] John Hopcroft, Omar Khan, Brian Kulis, and Bart Selman. Natural communities in large linked networks. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 541–546. ACM, 2003.

[66] H. A. Kestler J. M. Kraus, G. Palm. On the robustness of semi-supervised hierarchical graph clustering in functional genomics, 2007.

[67] Yuheng Jiang, Wai Yin Sally Lee, and Lukasz Golab. Analyzing student and employer satisfaction with cooperative education through multiple data sources. *Asia-Pacific Journal of Cooperative Education*, In press.

[68] Jonas. finding the best trade-off point on a curve. Accessed on 17 Feb 2015, at `http://stackoverflow.com/questions/2018178/finding-the-best-trade-off-point-on-a-curve`.

[69] Marco Saerens Pierre Dupont Jrme Callut, Kevin Franoisse. Semi-supervised classification from discriminative random walks. *Machine Learning and Knowledge Discovery in Databases*, 2008.

[70] David R Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999.

[71] David Kember, Amber Ho, and Celina Hong. The importance of establishing relevance in motivating student learning. *Active learning in higher education*, 9(3):249–263, 2008.

[72] Jan H Kietzmann, Kristopher Hermkens, Ian P McCarthy, and Bruno S Silvestre. Social media? get serious! understanding the functional building blocks of social media. *Business horizons*, 54(3):241–251, 2011.

[73] Suh-Ryung Kim, Jung Yeun Lee, Boram Park, and Yoshio Sano. The competition hypergraphs of doubly partial orders. *Discrete Applied Mathematics*, 165:185–191, 2014.

[74] Ilkka Kivimäki, Alexander Panchenko, Adrien Dessy, Dries Verdegem, Pascal Francq, Cédrick Fairon, Hugues Bersini, and Marco Saerens. A graph-based approach to skill extraction from text. *Graph-Based Methods for Natural Language Processing*, page 79, 2013.

[75] Jon M Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S Tomkins. The web as a graph: measurements, models, and methods. In *Computing and combinatorics*, pages 1–17. Springer, 1999.

[76] Vaishnavi Krishnamurthy, Michalis Faloutsos, Marek Chrobak, Li Lao, J-H Cui, and Allon G Percus. Reducing large internet topologies for faster simulations. In *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, pages 328–341. Springer, 2005.

[77] Renaud Lambiotte, J-C Delvenne, and Mauricio Barahona. Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint arXiv:0812.1770*, 2008.

[78] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2006.

[79] Harry Leung. The impact of information quality of job descriptions on an applicant's decision to pursue a job. 2007.

[80] M Li and B Lam. Cooperative learning, 2005.

[81] Pasquale Lops, Marco de Gemmis, Giovanni Semeraro, Fedelucio Narducci, and Cataldo Musto. Leveraging the linkedin social network data for extracting content-based user profiles. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 293–296. ACM, 2011.

[82] László Lovász. *Large networks and graph limits*, volume 60. American Mathematical Soc., 2012.

[83] G.D. Fatat Michael R. Berthold. High performance subgraph mining in molecular compounds. *HPCC 2005*, pages 866–877, 2005.

[84] Annie Moletsane. Work integrated learning (wil) stakeholder expectations in the hospitality industry. WACE Advancing Cooperative & Work-Integrated Education, 2011.

[85] M Nascimento, J Sander, and J Pound. Analysis of SIGMOD's Co-Authorship Graph. *ACM SIGMOD Record*, 32:2002–2004, 2003.

[86] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, Feb 2004.

[87] Mark EJ Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, 2004.

[88] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

[89] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[90] Ontario Ministry of Education. Earning more credits through workplace experience. Accessed on 30 Jun 2015, at https://www.edu.gov.on.ca/morestudentsuccess/coop.html.

[91] University of Waterloo. Mathematics: Programs. Accessed on 30 Jun 2015, at https://uwaterloo.ca/math/future-undergraduates/programs.

[92] University of Waterloo. Programs by faculty. Accessed on 30 Jun 2015, at https://uwaterloo.ca/find-out-more/programs-faculty.

[93] Co operative Education at University of Waterloo. Our programs. Accessed on 30 Jun 2015, at https://uwaterloo.ca/co-operative-education/about-co-operative-education/our-programs.

[94] Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.

[95] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[96] Srinivasan Parthasarathy, Shirish Tatikonda, and Duygu Ucar. A survey of graph mining techniques for biological datasets. In *Managing and mining graph data*, pages 547–580. Springer, 2010.

[97] Jeffrey Xu Yu Peixiang Zhao. Mining closed frequent free trees in graph databases. *Proceeding of Database Systems for Advance Application 2007*, pages 91–102, 2007.

[98] Joseph A Raelin, Margaret B Bailey, Jerry Hamann, Leslie K Pendleton, Jonathan Raelin, Rachelle Reisberg, and David Whitman. The effect of cooperative education on change in self-efficacy among undergraduate students: Introducing work self-efficacy. *Journal of Cooperative Education and Internships*, 45(2):17–35, 2011.

[99] Davood Rafiei. Effectively visualizing large networks through sampling. In *Visualization, 2005. VIS 05. IEEE*, pages 375–382. IEEE, 2005.

[100] Elizabeth Rainsbury, David Leslie Hodges, Noel Burchell, and Mark C Lay. Ranking workplace competencies: Student and graduate perceptions. 2002.

[101] Edwin Ralph, Keith Walker, and Randy Wimmer. Practicum-education experiences: Post-interns' views. *International Journal of Engineering Education*, 25(1):122–130, 2009.

[102] Sidney Redner. How popular is your paper? an empirical study of the citation distribution. *The European Physical Journal B-Condensed Matter and Complex Systems*, 4(2):131–134, 1998.

[103] S U Rehman, a U Khan, and S Fong. Graph mining: A survey of graph mining techniques. In *Digital Information Management (ICDIM), 2012 Seventh International Conference on*, pages 88–92, 2012.

[104] Matthew A Russell. *Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More*. " O'Reilly Media, Inc.", 2013.

[105] Thomas Schank and Dorothea Wagner. Finding, counting and listing all triangles in large graphs, an experimental study. In *Experimental and Efficient Algorithms*, pages 606–609. Springer, 2005.

[106] Last M. Bunke H. Kandel A. Schenker, A. Classification of web documents using a graph model. In *2013 12th International Conference on Document Analysis and Recognition*, page 240, 2003.

[107] Senate Undergraduate Council. Items for approval at october 8, 2013 senate undergraduate council. Accessed on 17 Feb 2015, at `https://uwaterloo.ca/secretariat-general-counsel/sites/ca.secretariat-general-counsel/files/uploads/files/suc_agenda_package_2013-10-08_0.pdf`.

[108] D Stengel, DR Young, P Chaffe-Stengel, and RM Harper. Assessing the academic and workplace skills of undergraduate business interns. *Journal of Cooperative Education & Internships*, 44(18):13–22, 2010.

[109] Lei Tang and Huan Liu. Graph mining applications to social network analysis. In *Managing and Mining Graph Data*, pages 487–513. Springer, 2010.

[110] Ilya B. Muchnik Thang V. Le, Casimir A. Kulikowski. Coring method for clustering a graph. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1 – 4, 2008.

[111] Glenn R Thiel and Nell T Hartley. Cooperative education: A natural synergy between business and academia. *SAM Advanced Management Journal*, 62(3):19, 1997.

[112] Takenao Ohkawa Tomonobu Ozaki. *Mining Correlated Subgraphs in Graph Databases*. Advances in Knowledge Discovery and Data Mining, 2008.

[113] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011.

[114] Charles S Ungerleider. *Evaluation of the Ontario Ministry of Education's Student Success/learning to 18 Strategy: Final Report*. Canadian Council on Learning, 2008.

[115] Duncan J Watts. *Six degrees: The science of a connected age*. WW Norton & Company, 2004.

[116] Miriam Weisz and Sandra Smith. Critical changes for successful cooperative education. In *Higher education in a changing world. Proceedings of the 28th HERDSA Annual Conference*, pages 605–616, 2005.

[117] Arnim Wiek, Lauren Withycombe, and Charles L Redman. Key competencies in sustainability: a reference framework for academic program development. *Sustainability Science*, 6(2):203–218, 2011.

[118] Alan Wilson. Strategy and management for university development. *Higher education re-formed*, pages 29–44, 2000.

[119] Alan Wilson. *Knowledge power: interdisciplinary education for a complex world*. Routledge, 2010.

[120] Zhijun Yin, Manish Gupta, Tim Weninger, and Jiawei Han. Linkrec: a unified framework for link recommendation with user attributes and graph structure. In *Proceedings of the 19th international conference on World wide web*, pages 1211–1212. ACM, 2010.

[121] Effandi Zakaria et al. Promoting cooperative learning in science and mathematics education: A malaysian perspective. *Coleccion Digital Eudoxus*, (22), 2009.

[122] K Zegwaard and Dave Hodges. Science and technology stakeholders ranking of graduate competencies. part 3: Graduate perspective. *Asia-Pacific Journal of Cooperative Education*, 4(2):23–35, 2003.