AN ANALYTICS BASED ARCHITECTURE AND METHODOLOGY FOR COLLABORATIVE TIMETABLING IN HIGHER EDUCATION

by

Carlos Alberto Sánchez

Masters of Information Sciences, University of Pittsburgh, 2003

Masters of Public Policy and Management, University of Pittsburgh, 1999

Magister en Administración (MBA), Universidad de Los Andes, 1994

Ingeniero Mecánico (B.Sc. Mechanical Engineering), Universidad de Los Andes, 1988

Submitted to the Graduate Faculty of

School of Information Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2015

UNIVERSITY OF PITTSBURGH

SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

Carlos Alberto Sánchez

It was defended on

December 2nd, 2015

And approved by

Dissertation Advisor: Stephen Hirtle, Ph.D., Professor, School of Information Sciences

Marek Druzdzel, Ph.D., Associate Professor, School of Information Sciences

Jerrold May, Ph.D., Professor, Joseph Katz Graduate School of Business

Paul Munro, Ph.D., Associate Professor, School of Information Sciences

Vladimir Zadorozhny, Ph.D., Associate Professor, School of information Sciences

AN ANALYTICS BASED ARCHITECTURE AND METHODOLOGY FOR COLLABORATIVE TIMETABLING IN HIGHER EDUCATION

Carlos A. Sánchez, Ph.D.

University of Pittsburgh, 2015

Class scheduling in higher education, also known as "timetabling", is a complex process that involves many people across an institution for several months every year, and literature on the topic has been rapidly evolving over the last 15 years. We propose architecture and methodology to enable the implementation of systems that can help users gain insight on non-trivial existing and emerging enrollment patterns that need to be considered for planning purposes, and to facilitate collaborative timetabling activities. University of Pittsburgh data on undergraduate enrollments during six recent fall terms is used to illustrate the proposed ideas. Core components are specified by: *First*, modeling the problem using Association Rule Analysis where the sets of courses that individual students take in an academic term are treated as transactions. This renders combinations of courses called itemsets. A new backtracking algorithm called MASAI is proposed to determine the ***ma***ximum number of ***s***eats ***a***vailable per ***i***temset. This corresponds to the identification of itemsets of interest as in the case at hand course itemsets with no seats available are primary targets. MASAI is a novel approach to the identification of itemsets of interest that uses information that is not available in transactional data to determine the maximum number of seats possible in each itemset. *Second*, in order to facilitate deeper analyses that consider the relationships between course itemsets, the problem is modeled as a multi-mode graph that incorporates information obtained with the Association Rule Analysis and MASAI. A Generalized Clique Percolation Method (GCPM) is proposed to enable the identification of overlapping and hierarchical communities in graphs/networks. GCPM is used to identify communities in the multi-mode graph, enabling the discovery of non-trivial enrollment patterns, and the identification of scheduling practices that limit the enrollment options for students. *Third*, the elements that would form the core of a socially translucent environment that is based on the previous components are discussed. This collaborative environment is intended to provide scheduling authorities with access to shared information on enrollment patterns and how decisions on scheduling of courses in their departments impact the overall institution's schedule and the enrollment options for students.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I express my gratitude to my advisor and dissertation committee chair, Professor Stephen Hirtle for his support and guidance over the years as I advanced through the PhD program and my dissertation.

I would like to thank the members of my dissertation committee, Professors Marek Druzdzel, Jerrold May, Paul Munro, and Vladimir Zadorozhny, whose comments and suggestions were of great help. I also want to express my gratitude to Professor Juan Manfredi who is the Vice Provost for Undergraduate Studies at the University of Pittsburgh, and to my colleague Mr. John Knox for their motivation, helpful ideas, opinions and critiques during the development of my dissertation.

My love and gratitude to my wife as without her support and dedication I would not have been able to complete my dissertation. My love and gratitude to my children as every hour that I dedicated to my studies was an hour that I did not dedicate to them.

I want to express my gratitude to my brother, whom I deeply love and admire. Through my life, he has been a role model of resilience and determination, and has always been there for me. My love and gratitude to my aunts for their unconditional love, support and encouragement through my life.

Above all, I want to express my love and gratitude to my parents. Without their unconditional love, dedication, and great effort to provide me with the best possible education that they could afford, I would not had been able to even start dreaming of having the life that I have had.

1.0 INTRODUCTION

One of the most complex business processes in higher education institutions is class scheduling, also known in the literature as "timetabling in higher education." At a higher education institution the timetabling process typically involves many people across multiple academic and administrative units for several weeks or months each year. Its outcome, a schedule of classes, has a direct impact on the access of students to the classes they need in order to complete their degrees on time, and on the institution's ability to effectively and efficiently use its human resources and facilities in alignment with strategic goals. Despite the potential for an efficient and effective timetabling process to enhance the competitive advantage of higher education institutions, and the theoretical advances on some of the core aspects of the problem, there have been just a handful of known implementations of those advances and there are still numerous open challenges.

Timetabling is considered a difficult problem from the theoretical and practical perspectives and there has been a wealth of research on the topic with early works and implementations starting in the late 1950s (Barraclough, 1965; Holzman & Turkes, 1964; Murphy, Sutter, & Laboratories, 1966; Sherman, 1958). Coincidentally, one of the pioneers of timetabling research as an application of operational research, Dr. Albert Holzman (1921-1985) was Professor and Chairman of the Department of Industrial Engineering at the University of Pittsburgh (Pitt). Since the early 2000s, there has been a noticeable increase in the volume of published works on timetabling. The increased research activity on the topic appears to be associated with two international timetabling competitions organized by the European Metaheuristic Network in 2002 (ITC 2002) and the International Conference on the Practice and Theory of Automated Timetabling PATAT in 2007 (Di Gaspero, McCollum, & Schaerf, 2007; Lewis, Paechter, & McCollum, 2007; McCollum, McMullan, Burke, Parkes, & Qu, 2007; McCollum et al., 2010).

So far, the timetabling community has focused mostly on the specification of timetabling scenarios and development of optimization algorithms to solve the resources allocation problem associated with a single academic term timetabling. More precisely, research has focused on specifying and solving the problem of generating class schedules for single academic terms given: A set of courses with a defined number of sections per course, the number of seats per section, a set of classrooms, a set of instructors, and a set of hard and soft constraints associated mostly with times, classrooms, and instructors. These efforts have enabled the systematic testing and evaluation of proposed algorithms based on agreed upon benchmarks. However, published research on timetabling has not yet explicitly considered the strategic and operational goals of higher education institutions as they relate to the primary mission of providing the best educational experience for students (Bonutti, De Cesco, Di Gaspero, & Schaerf, 2012; Kristiansen & Stidsen, 2013; McCollum, 2007; Schaerf, 1999).

Most higher education institutions operate on federated governance structures with schools and departments having a great deal of independence built-in to support academic freedom. Within that framework, a common approach to timetabling in U.S. universities is for academic units (i.e. schools, departments, and programs) to prepare and offer their class schedules in an independent fashion with little or no coordination between them. Students then enroll in the offered sections across academic units considering the requirements of their programs of study, interests, and offerings in the schedule of classes. In most cases, students do not advance in their programs in curriculum-like synchronization with their peers.

Large universities normally have tens of thousands of students registered in hundreds of programs across schools and departments that offer from hundreds to thousands of sections per term, frequently with multiple sections of the same course offered at multiple schedules. For instance, when considering only undergraduate students and sections, Pitt's Pittsburgh campus enrolls approximately 19,000 students; they take about 90,000 seats in 3,500 undergraduate sections each fall or spring term (counting only lectures, seminars, practicums and workshops). During the fall term, approximately 80% of undergraduate students enrolled at Pitt's Pittsburgh campus take classes on three or more subjects with 56% taking four or five subjects.

Within the described environments, a development that further highlights the importance of considering cross-enrollment patterns for timetabling, is that students increasingly enroll in multiple academic majors or programs. Pitt and Tepper (2012) report that "Double majoring has

become an important trend, especially at some of the most selective schools in America. While there has been a slight increase in double majoring on average across all colleges and universities, we see a steep increase (more than ten percent) at the most selective colleges, with many colleges seeing the ranks of double majors swelling to over thirty to forty percent of all graduates" (Pitt & Tepper, 2012). "At Vanderbilt University, the number of double majors has risen to nearly 40% of all students. At UC-Davis the number of double majors jumped 50% in five years; it has doubled at MIT since 1993. At Tufts, one-third of the students have a double major; at Georgetown, 23% (an increase of 60% since 1996); at Washington University, 42% of students in 2002 selected two majors (up from 28% five years earlier); and at Brown, 40%." (Pitt & Tepper, 2012).

Literature on timetabling in higher education has not yet considered the complexities associated with the described environments. Works on the topic assume that the required number of seats and sections for each term are given and do not explicitly consider that students take classes across departments and schools and that enrollment patterns change over time. When timetabling systems are used, the current approach is to take the requirements and constraints individually provided by departmental users as a given, and then to use optimization algorithms to search for optimal scheduling solutions that meet the provided requirements and constraints. However, optimization algorithms are neither designed nor intended to identify and solve inefficiencies embedded in constraints passed to them. Current solutions do not provide users with information that would help them to identify those inefficiencies and potentially enable them to formulate better informed constraints and/or change overall aspects of the institution's timetabling policies and practices when opportune. Furthermore, current solutions do not provide tools to help higher education administrators to gain insights into non-trivial enrollment patterns and areas where collaborative scheduling could be beneficial for immediate timetabling activities and for planning purposes.

The timetabling research community recognizes the referred gaps in published works that call for the advancement of the timetabling field toward solutions that are accepted and actually implemented in higher education institutions. New solutions need to provide scheduling authorities, who are the domain experts, with shared information that helps them to better understand enrollment patterns and, when available, specify better informed constraints for optimization algorithms. That is, rather than providing "black box" solutions, it is important to

develop decision support systems that engage administrators and faculty in charge of scheduling classes, with the goal of helping them develop better scheduling practices. There are identified needs like offering students maximum flexibility of choice when selecting courses to take (McCollum, 2007); improving measurability and reproducibility of solutions to timetabling problems (Schaerf & Di Gaspero, 2007); considering the inefficiencies embedded in the input constraints that are provided to the sophisticated optimization algorithms that the research community has developed; and leveraging the existing corpus of knowledge that exists in the field and at the institutional levels (De Causmaecker & Berghe, 2012).

There are also identified areas of improvement in higher education coming from outside the timetabling community that have the potential to bring benefits to institutions and students. These areas include better course scheduling and sequencing. For instance, the cost of building, maintaining and operating facilities is among the top items at the budget of higher education institutions (Blanchette, 2010; Kirshstein & Wellman, 2012). However, one of the main factors that permit the status quo in scheduling is low classroom utilization, which is reported to be around 30% at some universities in the United Kingdom (Fink, 2002; Geller, 2004) and about 56% in the United States (Braithwaite et al., 2012).

There are initial advances on some of the referred directions that include interactive design of timetables allowing individual users to specify the problem constraints and later on intervene in the creation of the solutions (Piechowiak, Ma, & Mandiau, 2005; Rudová, Müller, & Murray, 2011; Zeising & Jablonski, 2012). There are proposals that are based on users' detailed knowledge of an institution's academic programs to account for schedule conflicts (Zeising & Jablonski, 2012). There are approaches that use the knowledge of expert users to help transform the curriculum model into the enrollment model (Müller & Rudová, 2012). There are also approaches that survey faculty on courses that they consider should be offered in non-conflicting schedules (Wehrer & Yellen, 2013). However, in the absence of mechanisms to prevent competitive behavior derived from the federated governance and the culture of academic freedom that are the hallmarks in higher education institutions, the referred approaches lead to the embedding of inefficiencies in the required constraints that no optimization algorithm is going to be able to identify and solve. In fact, proposals for interactive systems report that users manipulate solutions produced by optimization algorithms to mimic the solution that they wanted

in the first place without much consideration of enrollment patterns, the impact on students, or the optimal use of human resources and facilities (Murray, Müller, & Rudová, 2007).

1.1 OBJECTIVES AND SCOPE

This dissertation proposes *an analytics based architecture and methodology for the design and implementation of collaborative timetabling systems in higher education*. While leveraging the wealth of research that has been produced on timetabling in higher education, this work expands the scope of the domain in directions that are frequently mentioned in the literature. It is intended to address core gaps that exist between research on timetabling and the needs of higher education institutions rather than to further developing timetabling optimization algorithms. The new architecture and methodology are intended to solve the problem of identifying non-trivial enrollment practices and patterns that cause schedule bottlenecks and that have the potential to be of value for planning purposes. The new architectural components are also intended to support and promote collaborative timetabling efforts in the usually decentralized higher education organizational environments. While the proposed architecture leverages research on association rule analysis (also known as market basket analysis), and identification of overlapping and hierarchical communities in networks, it also makes contributions of general application on those topics as detailed ahead (Agrawal, Imieliński, & Swami, 1993; Borgelt, 2012; Fortunato, 2010; Lancichinetti, Fortunato, & Kertész, 2009; Zaki & Ogihara, 1998).

In the decentralized organizational environments that are prevalent in higher education, no single person has the complete information or authority that are required to make decisions leading to an optimal schedule of classes for the institution. Thus, if the effort is structured as a group decision task, it can be classified as a conjunctive task (Steiner, 1972). In that case, successful decisions can only be achieved when all the group members maximize their efforts. In theory, an optimal decision cannot be achieved if at least one member of the group fails to contribute to the decision task. There is also a systematic relationship between patterns of communication and decision quality in a Group Decision Support System (GDSS) environment that significantly improve decision quality in conjunctive tasks (Lam, 1997). Thus, in principle a collaborative timetabling system has the potential to be beneficial.

The new architecture and methodology are intended to form the base for the design of collaborative timetabling systems. Those systems would facilitate the discovery of non-trivial enrollment patterns that can be of interest, and help foster collaboration among scheduling authorities based on a mutual understanding of those patterns. A system implemented using the proposed architecture, would provide users with shared information on current and historical enrollments on courses under their purview, cross-enrollments of interest, and on the impact of their decisions on the quality of the institution's course schedules. The provided information would help users to gain shared non-trivial insights on existing and emerging enrollment patterns that can be used for planning purposes, and to specify better informed constraints to be passed to optimization algorithms. Pitt's, Pittsburgh Campus is used as a case study to illustrate the presented ideas with real enrollment data from six recent fall academic terms. Specific architectural components are presented in detail and results from prototype implementations that use Pitt enrollment data are discussed.

The proposed architecture includes components that use and make contributions to Association Rule Analysis and network analysis. *First,* a new approach for the identification of itemsets of interest in association rules is presented. *Second,* a new generalized clique percolation method for the identification of overlapping and hierarchical communities in networks (hereinafter interchangeably referred to as graphs) is presented. Although the discussion focuses on timetabling on higher education, both approaches have general applications as illustrated in the references provided in the relevant sections.

A group of architectural components is intended to identify the combinations of courses that individual students enroll in per term, and that are offered with schedules and/or enrollment capacities that limit their enrollment options. This is achieved by modeling the problem as an Association Rule Analysis that explicitly considers that students take classes across departments and schools, and without requiring any detailed knowledge about the programs offered at the institution. The Association Rule Analysis renders a set of combinations of courses called course itemsets or itemsets. In order to identify the itemsets of interest, a new algorithm that determines the ***ma***ximum number of ***s***eats ***a***vailable per ***i***temset called "MASAI" is proposed. Within this context, course itemsets of primary interest are those that have reached a capacity limit or potentially those that are under enrolled below predefined thresholds. For completeness, there is a section dedicated to the complexity analysis of the problem that MASAI is designed to solve.

MASAI advances the topic of identification of itemsets of interest in Association Rule Analysis. There is a voluminous literature on association rules that proposes methodologies to identify itemsets and rules of interest based on the information existing in the transaction data sets. MASAI departs from that approach and incorporates information that is not directly available in transactional data used in Association Rule Analysis. For the particular case of timetabling, the natural limitations to enrollments in sets of courses are derived from enrolment limits in individual sections offered and/or from conflicting schedules. Thus, MASAI uses information on seats offered and schedules at the level of individual sections of courses to identify the maximum possible enrollment capacity on each course itemset identified with the Association Rule Analysis. A prototype implementation of MASAI on ORACLE PLSQL (Feuerstein & Pribyl, 2005) is used on the Pitt data set enabling longitudinal analyses of results over the six terms of enrollment data under study. This provides a first pass on the identification of combinations of courses of interest and enrollment patterns.

Another group of architectural components enables a deeper analysis of enrollment patterns, and the identification of groups of courses that could potentially benefit from changes in scheduling practices and/or collaborative scheduling. This is achieved by modeling the problem as a multi-mode weighted graph and using the information derived from the Association Rule Analysis and MASAI results to facilitate the community identification process in the resulting graph. The proposed approach leverages the theoretical link that exists between Association Rule Analysis and graph analysis; namely, that closed itemsets resulting from Association Rule Analysis are by definition cliques in graph analysis (Zaki & Ogihara, 1998). This conceptual bridge along with the modeling of the problem as a multi-mode weighted graph leads to a Generalized Clique Percolation Method (GCPM) for the identification of overlapping and hierarchical communities in graphs. In contrast to the standard clique percolation method (CPM) that considers only cliques of equal size (*k*-cliques), GCPM identifies cliques of different sizes that form communities (*multi-k*-cliques) (Derényi, Palla, & Vicsek, 2005; T. S. Evans, 2010; Fortunato, 2010; Fu, Kang, Zhicun, Lansheng, & Jing, 2014; Lancichinetti et al., 2009; Palla, Derényi, Farkas, & Vicsek, 2005). The NEO4J graph database is used to implement a prototype version of the referred multi-mode graph and GCPM (Miller, 2013; Webber, 2012).

The two referred groups of components would serve as backbone for the implementation of an environment that enables collaborative timetabling activities. The results of the community

analyses provide the basis to form working groups of users that might benefit from collaborative work on course timetabling activities or analysis of enrollment patterns. In order to help achieve collaboration in the usually decentralized academic and administrative environment of a higher education institution, the timetabling environment would be designed using the core precepts of *social translucence* (i.e. visibility, awareness, and accountability) (Erickson & Kellogg, 2000). The basic framework for the collaborative timetabling environment is discussed, leaving a discussion on the design of a fully-fledged system for future work.

Although the focus of this dissertation is on higher education, timetabling has applications in other areas where the goal is to optimize the scheduling or allocation of limited resources in predefined timeslots. For instance, there are applications on convention planning, scheduling rosters of on duty medical personnel, and athletic or game tournaments (Burke, De Causmaecker, Berghe, & Van Landeghem, 2004; Burke, Kendall, et al., 2004; Moody, 2011). Applications using operational research have existed for decades in several areas, i.e. traffic control, production lines, scheduling for trains, airplanes, buses, financial portfolio optimization, etc. (Thompson & Thore, 1996).

The identification of overlapping and hierarchical communities is of importance in numerous disciplines where systems and/or groups of people are often represented as networks, and literature on the topic is still evolving (Fortunato, 2010; Kramer, Dutkowski, Yu, Bafna, & Ideker, 2014; Lancichinetti et al., 2009; Palla et al., 2005; Paul, Anand, & Anand, 2015; Pennacchioli, Coscia, & Pedreschi, 2014)

1.2 ORGANIZATION OF THE DOCUMENT

The rest of the document is organized as follows.

Section 2 reviews literature on timetabling in higher education, focusing on past and current implementations of timetabling systems and emerging approaches to the problem. It concludes with a discussion on the current typical architecture of timetabling systems in higher education. Literature review on other topics is discussed in the sections where those topics are presented.

Section 3 presents an overview of the proposed architecture and methodology. Following sections focus on specific aspects of the architecture.

Section 4 discusses the group of architectural elements that deal with the identification of combinations of courses of interest. This includes modeling the problem as an Association Rule Analysis, a relational schema to support data processing across components, the presentation of an algorithm to determine the maximum number of seats available per course itemset MASAI, and a discussion of results obtained with this group of components.

Section 5 focuses on the identification of overlapping and hierarchical communities of courses using graph analysis. This includes modeling the problem as a multi-mode graph, a generalized clique percolation method (GCPM), a prototype implementation of the multi-mode graph using NEO4J, and analyses of the resulting graph including the proposal of a metric called Enrollment Weighted Degree Centrality (EWDC), and metrics on Clique Overlap and Clique Weighted Overlap.

Section 6 presents analyses of selected scheduling cases that illustrate the combined use the components and methodology discussed in the previous two sections.

Section 7 discusses the taxonomy of decision groups task structures, how timetabling fits into it, and how a translucent environment would foster collaborative timetabling activities in higher education. The framework for a socially translucent environment that uses information derived from the components presented in sections four and five is discussed. Implementation of the framework is left for future work.

Section 8 presents conclusions, including the main contributions of this work, and discusses areas of future research on the topic of collaborative timetabling, and benchmarking of GCPM with data from other disciplines.

Appendix A discusses enrollment data and practices at Pitt's, Pittsburgh campus. It is used as study environment to illustrate the ideas proposed in this dissertation in the case of a large higher education institution.

Appendix B provides results of one of the analysis discussed in Section 4.4. More precisely, it includes a list of courses that satisfy general education requirements that cannot be taken together due to schedule conflicts during the six academic terms under analysis.

Appendix C includes the NEO4J scripts used to create objects and load data into the multi-mode graph. Appendixes D and E present statistics on the multi-mode graph.

2.0 RELATED WORK ON TIMETABLING IN HIGHER EDUCATION

The two international higher education timetabling competitions held in 2002 and 2007 have contributed to the establishment of systematic testing and evaluation procedures using benchmarks and scenarios derived from real world applications. Advances on the topic have rendered a rich collection of optimization algorithms that can handle complex timetabling problems and are flexible enough to perform on different scenarios. This was demonstrated at the 2007 International Timetabling Competition where Müller's algorithm succeeded as a finalist in the three competition tracks and was the winner in two of the them (Di Gaspero et al., 2007; Lewis et al., 2007; McCollum et al., 2007; McCollum et al., 2010; Müller, 2009)

Just before the 2007 Timetabling Competition, Schaerf and Di Gaspero (2007) discussed *measurability* and *reproducibility* of solutions to timetabling problems. They see them as critical elements to facilitate the coalescing of the timetabling efforts into a well-established research community. They also proposed the creation of a web based Problem Management System (PMS) to help researchers contribute their content within a standardized framework, i.e. addition of results and instances, management and analyses of instances generation, data translation, visualizations and organization of on-line competitions. (Schaerf & Di Gaspero, 2007). More recently, Bonutti and colleagues have aimed to close the gaps between theory and practice that McCollum identified in 2006-07. They seek to create models for timetabling that more closely reflect the realities of timetabling practice and look for ways to apply the results of research to real applications (Bonutti et al., 2012) .

There are discussions on the directions that timetabling research needs to advance in order to better reflect the needs of higher education institutions. These include offering students maximum flexibility of choice when selecting courses to take, providing flexibility to administrative and instructional staff while using the human resources capacity efficiently and effectively, and optimizing the use of existing and planned facilities (De Causmaecker & Berghe, 2012; Fernandes, Pereira, & Barbosa, 2015; McCollum, 2007).

There are relatively few cases of implementations of timetabling systems in higher education. This section presents a summary of the most known implementations and emerging

approaches to the problem. The section closes with a description of the typical architecture of timetabling systems that is based on the literature review.

2.1 AN INTERACTIVE COURSE TIMETABLING SYSTEM AT ROLLINS COLLEGE, UNITED STATES

A timetabling system was designed and implemented for the Science Division at the Rollins College in Florida, USA to support the scheduling of approximately 100 classes per term (Yellen & Wehrer, 2013). This approach has three interesting aspects that relate to the ideas presented in this dissertation. First, they make an attempt to consider the reduction of schedule conflicts by surveying faculty on courses that they consider should be offered in non-conflicting schedules. Then, they model the collected information as constraints in their system. Clear limitations of this approach are that there is no guarantee that all undesirable schedule conflicts would be identified, and that there is no identification of the combinations of sections that limit the enrollment opportunities for students. Furthermore, although the identification of schedule conflicts to avoid through a faculty survey might work in a small college, it is not likely that at a large institution any person would be able to provide an exhaustive identification of the schedule conflicts to avoid based on their knowledge of the institution.

A second aspect of interest in Yellen and Wehrer's work is that they revisit the use of the graph coloring methodology for schedule optimization, which has been used in the past (Carter, 2001; de Werra, 1985; D De Werra, 1997; Dominique de Werra, 1997). They model the problem as a graph with each node representing a course, the edges representing conflicts to avoid, and the colors time slots. The optimization problem is then to color each vertex in a way that adjacent vertices have non-overlapping colors (non-conflicting timeslots). Thus, they use graph analysis as an optimization tool. That, as opposed to our use of a multi-mode weighted graph for identification of non-trivial enrollment patterns and communities of courses of interest.

The third aspect of interest in Yellen and Wehrer's work is that they propose an interactive system to improve the schedules produced by the optimization procedure. However, the interactivity is limited to individual users working with the system administrator and using a visualization tool to modify the produced schedules.

2.2 UNITIME AT THE UNIVERSITY OF PURDUE, UNITED STATES

UNITIME is a comprehensive scheduling system that has been implemented at Purdue University and is available under the open source GNU license (UNITIME). Purdue University is a large higher education institution with approximately 39,000 students. In a typical term the class schedule includes about 9,000 classes in 570 teaching spaces and about 259,000 individual student class requests are satisfied (Müller, 2009; Müller & Murray, 2010; Müller & Rudová, 2012; Murray et al., 2007; Rudová et al., 2011).

UNITIME decomposes the timetabling problem into a series of sub-problems. This decomposition allows the system to handle the fact that at a large university students take classes across several departments and that there is a combination of local and global management of resources, i.e.: faculty time and rooms. The sub-problems are: A centrally timetabled large lecture problem, individually timetabled departmental problems, and a centrally timetabled computer laboratory problem. The large lecture problem includes sections with students from different programs using a relatively lower number of large rooms and is solved first. The departmental problem is solved second considering the output of the large lecture problem. The laboratory problem is solved last considering the output of the previous two problems.

At Purdue students have the ability to preregister for classes, and about 50% of them do. Thus, their timetabling problem can be considered as a combination of the post-enrollment and the curriculum based enrollment problems as defined in the 2007 timetabling competition. Preregistration information is used to plan for class capacities and their schedules, in combination with projections on enrollments in individual classes provided by users. In order to serve the needs of students who preregister, a student sectioning procedure is used (Müller & Murray, 2010).

The need to perform student sectioning led to the modeling of the course structures in a way that represent their complexity while allowing the use of optimization algorithms designed for timetabling. More precisely, one of the simplifications used in optimization algorithms that permits the handling of the timetabling problem is that they do not normally consider that individual instructional offerings can be composed of several related sections. For instance, a large introductory Biology Science class (instructional offering) normally includes the lecture, several recitations, and laboratory sections where students have to be registered. In another case,

different sections of the same course offered by different faculty might have different requirements in terms of laboratories or computer aided recitations. This means that in some large courses there might be tens or even hundreds of classes associated with an individual instructional offering. Timetabling has to be done at the level of individual classes that compose each instructional offering. To solve the problem, in UNITIME individual instructional offerings define a parent child hierarchical relationship that includes the components of the offering. Individual hard constraints are added to the optimization problem to control and/or prevent conflicts among sections that compose the instructional offering (Murray et al., 2007).

UNITIME's conceptual design is oriented to provide a system that assists departments and scheduling authorities in their task rather than replacing them with an automated system. To that end, departmental users and central authority schedulers are allowed to enter the desired section capacities as well as hard and soft constraints required for each course. Then, when the optimization algorithm finds a solution, users are allowed to modify it before committing to a final solution. Although this approach facilitates acceptance and use of the system, the authors report the evident presence of competitive behavior for preferred times and rooms leading to non-optimal schedules being finally implemented.

Although optimization algorithms can find optimal solutions to a given problem, the number and quality of preferences that users specify in the form of hard and soft constraints restrict the space of possible solutions beforehand. Furthermore, after solutions are reached, modifications in the interactive part of the process increase the number of conflicts as users try to make the system reach the solution that they are used to: "In particular, the large increase in student conflicts between the computed best solution and the final solution committed by the schedule manager in the large lecture problem is largely attributable to adjustments to accommodate faculty time preferences. These had been the primary criteria in manually building timetables since data on student conflicts were not available to be considered in the past" (Murray et al., 2007).

2.3 AN OPEN EXTENSIBLE ARCHITECTURE AT THE UNIVERSITY OF SHERBROOKE, CANADA

Rubio and Muñoz (2004) discuss and early effort to develop open and extensible architecture to implement timetable production systems for courses and exams. Along the same lines than other researchers, they note that most of the work on this area has focused on development of algorithms and mathematical formulations to deal with the problem of timetable construction. Conversely, not much effort had been invested at the time on the overall process of timetable production (Rubio & Munoz, 2004).

They propose a timetabling system based on a standard three-tier architecture including an interface for data input-output, a business logic layer that includes business rules and incorporates a timetabling module, and a data persistence layer that takes care of the physical storage of data. It is based on an information flow that includes the data sources, timetable production software, and results output through reports or a web page. Data sources include, basic data on students, instructors and resources from an existing data warehouse; and period data including the offerings and seats (or students' registration for the case of pre-registration) for specific term and timetable data, which is the final output.

2.4 EDT2004 AT THE INSTITUTE OF SCIENCES AND TECHNIQUES OF VALENCIENNES, FRANCE

An open interactive timetabling tool called EDT2004 was developed at the Institute of Sciences and Techniques of Valenciennes (ISTV) in France. EDT2004 is designed to help users build timetables by providing guidance through an interactive interface. Their stated approach is to be focused on the users as opposed to the problem or the algorithms available to solve it (Piechowiak et al., 2005).

The argument presented for the referred approach is fourfold: *First*, graphical tools are easy to use but do not provide help to solve conflicts in timetables. *Second*, fully automated tools can find optimal timetables but require powerful machines. *Third*, when constraints lead to the impossibility of a clash-free timetable automated tools do not provide explanations for the lack

of solutions. *Fourth*, the quality of automatically produced timetables depends on the exhaustiveness of the constraints, which at a large university is difficult or impossible to achieve. Thus, their approach is based on the expertise of users at the university. The system provides guidance based on a hierarchical model of resources available combined with hard and soft constraints.

EDT2004 defines sets of physical and preference constraints, which mostly correspond to hard and soft constraints as defined in the literature, i.e.: while physical constraints cannot be modified by users, preference constraints may be violated leading to timetables of "lower quality." Although the metric to assess quality of timetables is not precisely defined in the paper, it refers to constraints that are "used to express what a 'good' timetable should be for the students and for teachers." (Piechowiak et al., 2005).

The system defines three types of users: Designers (pedagogical managers), analyzers (administrative managers) and consultors (teachers and students). First, designers build timetables in a collaborative fashion based on meetings and discussions. Then, analyzers take care of room assignments. If users prefer, the system provides help with room allocation using optimization algorithms.

At the time of their publication, they were looking to advance their system with the inclusion of capabilities to help users to express constraints and consider them in an interactive fashion. Their stated goal was to create a system that combines automated and interactive generation of timetables, which appears to be a similar direction as the one that Müller and Barták had already explored as explained in the following section and that is at the core of UNITIME as discussed in Section 2.2 (UNITIME).

2.5 INTERACTIVE TIMETABLING SYSTEM AT CHARLES UNIVERSITY, CZECH REPUBLIC

An interactive timetabling system was implemented at the Faculty of Mathematics and physics at Charles University in the Czech Republic. The problem consisted of scheduling 746 lectures. Resources are known and described as including among others, 479 teachers and 41 classrooms at three different locations. It is assumed that the number of students who will attend the lectures

is known beforehand. Physical and human resources available are specified along with a list of hard and soft constraints (Müller & Barták, 2002).

The idea behind the proposed approach is to allow users to intervene at any step during the creation of the timetables to modify the outcome and further processing. That is, the system interface allows users to see how the timetabling is being built and let users intervene during the process by changing tasks at runtime. Their system uses a combination of a local search algorithm with a backtracking algorithm. Although a system is reported to have been implemented, there is no discussion on the impact of the intervention of users in the quality of the solutions other than describing that the required constraints were met.

The proposed algorithm uses interactive scheduling in an iterative fashion. There are two basic data structures including a set of activities that are not scheduled and a partial feasible solution. At each iteration step, the algorithm tries to improve the partial schedule that it already has towards a feasible schedule. Users can interrupt the algorithm after any iteration through a graphical user interface, and decide to take the current solution even if it is not complete, or manually allocate some of the activities that have not yet been scheduled. Users can also modify the constraints or add new activities during the intervention.

2.6 SCHEDULEXPERT AT THE CORNELL UNIVERSITY SCHOOL OF HOTEL ADMINISTRATION, UNITED STATES

A system called “SchedulExpert” was implemented at the Cornell University School of Hotel Administration in the early 2000s. At the time, this school offered 200 classes taught by 60 faculty members in 21 classrooms. The project started after a review of the school showed that ineffective course scheduling was making it difficult for students to complete their programs on time, causing dissatisfaction among faculty and resulting in inefficient use of classroom facilities. For instance, some required courses and electives in the same areas were being offered at the same time, on the same day, and at times dictated by the preferences of faculty, who were unaware of the impact of their decisions (Hinkin & Thompson, 2002).

SchedulExpert helped solve the referred issues and brought down the time required to prepare the schedules from weeks to a few hours. It demonstrated that is possible to gain faculty

acceptance and support even when their preferences are not necessarily the top priority. The system was further developed as commercial software that became popular in small schools and was later on licensed to SAP America.

2.7 COURSE AND EXAMINATION TIMETABLING AT ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS, GREECE

In the early 2000s, A PC-based computer system was designed and implemented to aid with the construction of course and examination timetables at the Athens University of Economics and Business. There are six departments at the University, each one with three or more specializations. The scheduling is done for about 180 courses each semester and the optimization module of the system was built using Integer Programming. To facilitate the construction of the system some university streams were selected to be free of conflicts. A university stream is a set of compulsory an elective courses that students follow through eight semesters. Thus, the classes to offer, number of seats to offer and scheduling restrictions were given. In 2004, they followed with a distributed version of the system that allowed users at departments to enter the required sections and faculty restrictions for each term. There is not interaction among users that would facilitate collaborative scheduling (M Dimopoulou & Miliotis, 2001; Maria Dimopoulou & Miliotis, 2004).

2.8 TIMETABLING AND STUDENT SCHEDULING SYSTEM AT THE UNIVERSITY OF WATERLOO, CANADA

A comprehensive course timetabling and student scheduling system was in place at the University of Waterloo between 1979 and 1985. The system was demand driven --Post-Enrollment-- in the sense that students first chose their courses and the system tried to design the best schedule to satisfy the selections. In his 2001 account, Carter reports that while initially the project expectation was that the success of the system would depend on the quality of the

schedule produced by the algorithms, they quickly discovered that what departmental administrators valued the most was the ability to make real time on-line changes to the course schedules. An illuminating observation is that timetabling is a highly political process with no hope of incorporating the complete spectrum of preferences and alternatives in an automatic quantitative system. The system was abandoned after 15 years due to its lack of portability to "even within Waterloo" (Carter, 2001).

2.9 TIMETABLING SYSTEM FOR THE ANDERSON SCHOOL OF MANAGEMENT AT UCLA, UNITED STATES

In the mid-1990s there was an implementation of a smaller scale timetabling system for the Anderson School of Management at UCLA. An explicit measure of success in this implementation was that "a timetable is 'good' when it meets most of the teacher's preferences." The design of this system was based on the premise that teachers are assigned to courses and that the number of courses and sections to schedule are known beforehand. A published report indicates that there were gains in scheduling time --which went from weeks to hours--, the data gathering process and the quality of the final schedule (Stallaert, 1997).

2.10 EMERGING APPROACHES TO TIMETABLING

Emergent lines of development in timetabling include the incorporation of knowledge of courses prerequisites with the goal of reducing the search space for optimal solutions, and the use of autonomous agents to negotiate conflicts when building schedules.

Zeising and Jablonski (2012) discuss a prototype that they tested at the University of Bayreuth in Germany. They propose an interactive timetabling system that allows a scheduler to visualize intermediate solutions coming from the optimization procedure, modify them and trigger a new optimization. Their main contribution is that they reduce the search space for the optimization algorithms by considering the structure of regulations (equivalent to curriculums in

the U.S.) to identify courses that individual students would not be taking together in the same term and courses that could potentially be taken together, i.e. courses that are linked through a prerequisite or co-requisite relationship. An important aspect of this contribution is that it appears to be the first published work that explicitly attempts to consider schedule conflicts from the point of view of the students. Most other works, including the design of the timetabling competitions, focus on the conflicts of instructor schedules and rooms (Zeising & Jablonski, 2012).

An assumption that facilitates the referred approach is that students are registered only in one regulation at a time, i.e.: there are no students with multiple majors. Information to build the conflict graphs is derived from detailed knowledge of the academic programs at the University of Bayreuth. The outcome is a set of session conflicts where conflicting sessions must not overlap over time. This set of session conflicts is then added to the list of constraints that is included into the selected optimization algorithm.

There are other developments that use existing architectures and methodologies to develop multi-agent systems (MAS) for scheduling. In both cases, the resources are time slots and the actors are teachers, groups of students and rooms. Each actor is represented by agents that encapsulate their characteristics, availability and restrictions (Mathieu & Verrons, 2006; Oprea, 2007). Oprea splits the process in two stages: *First*, Course timetabling scheduling at the level of Faculty (school) including allocation of course day and time. *Second*, course rooms allocation. Verrons and Mathieu take an integrated approach that allocates courses, days, times and rooms simultaneously. In both cases, it is assumed that the number of classes and seats to offer is known beforehand.

The MAS approach to timetabling looks for solutions that would satisfy the requirements of teachers while keeping them private. That is, when a conflict is encountered, a predefined negotiation protocol between agents ensues. If no solution is reached the agent communicates with its owner who might privately relax a constraint or suggest alternatives. Due to the possibility of negotiation protocols being terminated without a successful outcome, there is no guarantee that all required sections will be scheduled or that solutions will be optimal.

2.11 CURRENT TYPICAL ARCHITECTURE OF TIMETABLING SYSTEMS IN HIGHER EDUCATION

This section presents the most typical architecture found in timetabling systems in higher education. It is based on the findings discussed in the literature review in previous sections, and is intended to provide the reader with a conceptual schema of the elements most frequently found in current timetabling systems architecture. It provides a reference for comparison with the ideas presented in Section 3.

Current Timetabling systems discussed in the literature can be described with the general architectural elements shown in Figure 1 below. They include:

- A group of components includes data sources, data preparation (ETL processes), and predefined processes to specify and incorporate structural constraints (e.g. predefined available facilities for a term).
- At the core of a timetabling system is an optimization algorithm, of which the most advanced in the timetabling domain are interactive backtracking algorithms. As discussed in the previous sections, those algorithms enable users to modify restrictions and/or solutions during processing.
- An interface to enable users to interact with the system. This includes entry of initial and modified constraints during processing. At this time, state of the art architectures and systems that enable interactivity do so only for individual users. They have to make scheduling decisions based on information external to the system and provide hard and soft constraints.

Current architectures and system implementations do not provide users with support to help them identify non-trivial enrollment patterns for planning and scheduling purposes and to formulate better informed constraints to be passed to optimization algorithms.

Current systems that enable interactivity do not support collaborative scheduling. The goal of this dissertation is then to propose new architectural elements and methodology that enable the referred activities.

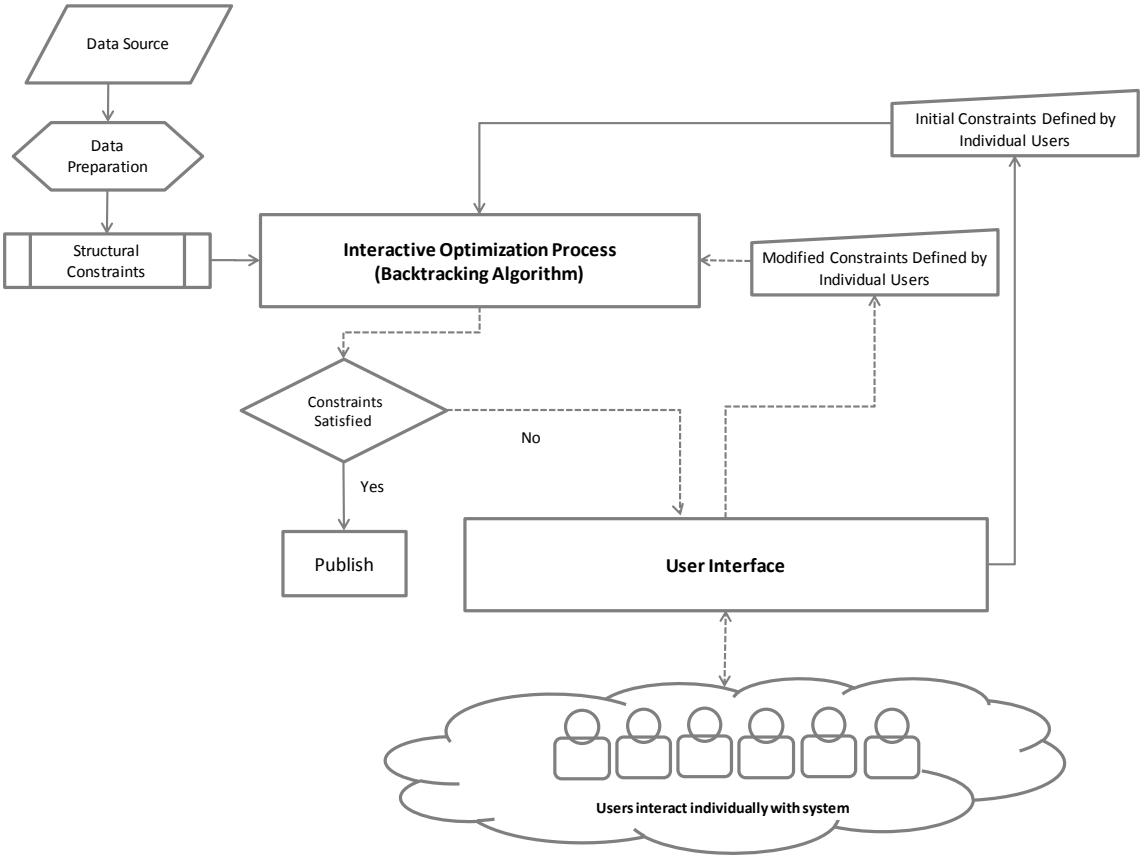


Figure 1 Current typical architecture of timetabling systems in higher education

3.0 OVERVIEW OF PROPOSED ARCHITECTURE AND METHODOLOGY

This section provides and overview of the proposed architecture and methodology. It includes a description of the conceptual approach to the problem, a summary of each architectural component, and how they are organized and presented through the document. Before delving further into the topic, a couple of naming conventions are presented.

- A *course identifier* has three components: the school or academic unit that owns the course, the academic subject of the course, and its catalog number. For instance the identifier (ARTSC, MATH, 0220) or ARTSC_MATH_0220 refers to a course offered by the School of Arts and Sciences, with subject Mathematics and catalog number 0220.
- A *class number* is used to distinguish multiple sections of a given course. Each section of a course is identified by a class number (labelled as CLASS_NBR).

If we are to explicitly consider the enrollment patterns across course sections, courses, and academic units in support of better informed timetabling activities, then it is necessary to: *First,* identify all the unique combinations of courses that students take or are able to take per term as well as those combinations that are not possible due to schedule conflicts but that could be of interest. This is done using historical enrollment data from recent terms, and data from ongoing enrollments in a current term. *Second,* identify the courses and combinations of courses of primary interest per term, which are those with section offerings that limit the enrollment options for students. *Third*, identify non-trivial enrollment patterns. *Fourth,* provide scheduling authorities with tools that enable exploration and visualization of enrollment patterns; and when necessary enable them to work in a collaborative manner to produce better class schedules. *Fifth*, enable the analyses of enrollments in the current term with the goal of providing scheduling authorities with ongoing enrollment data, and help them to make well informed scheduling changes during the open enrollment period. *Sixth,* develop components that are compatible among them and with existing institutional systems; and if available, permit interaction with existing optimization algorithms.

We start with some of the available typical architectural components as shown in Figure 2 and propose new components leading to a new architecture that considers the items described

above. More precisely, we start with a data source, a data preparation process that is unique to each data source, a set of structural constraints that is unique to a specific implementation, and one of the optimization algorithms that the timetabling community has developed. From that starting point, this work presents a full new set of components and methodology that integrate into a whole architecture to support collaborative timetabling systems in higher education. Furthermore, from the discussion ahead, it will be clear that the new architectural components can be of value by providing insights on enrollment patterns, and helping to identify practices that have a negative effect on the quality of schedules.

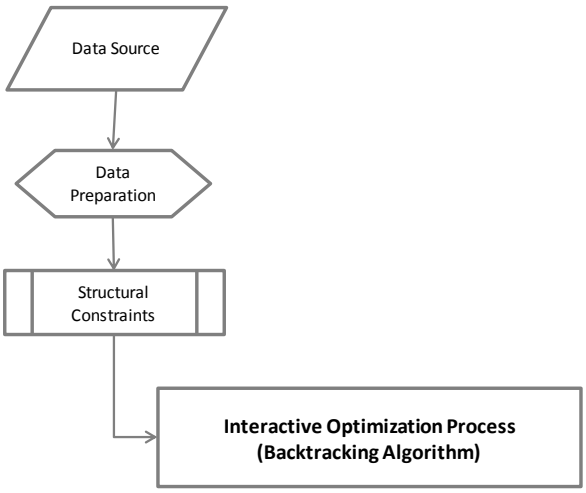


Figure 2 Stage 0 Architectural Diagram: Basic available architectural components

The following paragraphs present a summary of the proposed architectural components as they are presented in the document. Following sections are dedicated to discuss each component in detail along with results from prototype implementations. After every group of components is discussed, it is added as a new stage to the basic architecture.

Section 4 focuses on the identification of combinations of courses of interest as follows. Section 4.1 discusses the exhaustive identification of unique course combinations that students enroll in during a term. This is done by modeling the problem as an Association Rule Analysis. As summary, a transaction is defined as the group of courses that an individual student enrolls in

during a term (e.g. MATH 0200, ENGLISH 0220 and ECONOMICS 0100, where the label refers to the academic subject and the number to the catalog/level of the course). Each course is an item in the transaction and the group of courses in the transaction is a course set or itemset. Having the data modeled in the described form enables the direct use of Association Rule Analysis and algorithms for that purpose that are well-known and implemented in multiple software packages (Agrawal et al., 1993; Agrawal & Srikant, 1994; Borgelt, 2012; Krajca, Outrata, & Vychodil, 2011; Srikant, Vu, & Agrawal, 1997).

Among others, the output of the Association Rule Analysis indicates how many students enrolled in each course set. As an illustration, one course set could be MATH 0200, ENGLISH 0220 and ECONOMICS 0100 with 100 students. In the next step, it is then necessary to identify if 100 students enrolled in that particular course set because no more students were interested (i.e. there is capacity left) or because an enrollment limit was reached due to schedule conflicts or number of seats offered (i.e. there are no more seats possible in the itemset). Those are course sets of interest as they include courses with sections that potentially limit the enrollment options for students. To identify them, it is necessary to analyze all the course sets at the level of individual sections along with their schedules as the information needed to identify the effective capacity of each course set is not included in the transactions data.

Section 4.2 discusses a de-normalized relational schema that facilitates the following: Data processing across components, the operations of the algorithm referred in the following paragraph, data transferring across architectural components, and longitudinal analyses of historical enrollment data.

In Section 4.3 a backtracking algorithm called MASAI is proposed to determine the maximum number of seats available per itemset. It considers actual enrollments, enrollment limits, and schedules at the level of individual sections on each course set. There are sub-sections that discuss the complexity of the problem at hand, results obtained with a prototype implementation of MASAI, and a sample case analysis.

Section 4.4 discusses the methodology to identify couples of courses that that cannot be taken together due to schedule conflicts or that limit the available options to students for other reasons. Couples of courses with schedule conflicts cannot be identified using the Association Rule Analysis as they do not show up in transaction records of courses that students enroll in during a term. This problem corresponds to the identification of negative association rules (Koh

& Pears, 2007; Rani, Srinivas, Reddy, & Govardhan, 2011; Wu, Zhang, & Zhang, 2004; Yuan, Buckles, Yuan, & Zhang, 2002).

Section 4.5 presents Stage I of the proposed architecture by adding the components discussed so far to the basic typical architecture (Stage 0).

Section 5 focuses on the identification of overlapping and hierarchical communities of courses using a multi-mode graph analysis. This is needed because the Association Rule Analysis does not provide detailed explicit information on the relationship among combinations of courses of interest.

In Section 5.1, the problem is modeled as multi-mode graph. This approach leverages the theoretical links that exist between association rules and graph analysis, and enables the use of mature network analysis methodologies. The combined use of association rules and graph analysis enables the identification and visualization of enrollment patterns that are not possible with current timetabling methodologies.

Section 5.2 presents an overview of the community identification problem. Section 5.3 discusses the current Clique Percolation Method (CPM). Section 5.4 presents a generalized Clique Percolation Method (GCPM). It enables the identification of overlapping and hierarchical communities in graphs. Section 5.5 discusses a prototype implementation of the multi-mode graph using the NEO4J graph database system. Section 5.6 discusses the implementation of GCPM in NEO4J. Section 5.7 presents analyses of the graph database, GCPM results, propose graph metrics (e.g. Enrollment Weighted Degree Centrality), and presents examples using Clique Overlap and Weighted Overlap metrics. Section 5.8 presents Stage II of the proposed architecture and adds it to the components discussed so far.

Section 6 presents analyses of selected cases obtained from Pitt's enrollment data. The analyses combine the results of the Association Rule Analysis, MASAI and GCPM.

Section 7 discusses the framework for a translucent environment that would be based on the architectural components previously discussed. It includes a discussion on Decision Support Group Task Structures in Section 7.1, and Social Translucence in Section 7.2. The presentation sets the ground for continued work on the development of a fully-fledged framework that would enable the implementation of a collaborative timetabling system designed from the ground up to support social translucence. Section 7.3 presents the complete proposed architecture by adding Stage III including components of the socially translucent environment.

Figure 3 below provides a high level overview illustrating the three main blocks of components (i.e. Stage I, Stage II and Stage III) that are discussed and how they add to Stage 0.

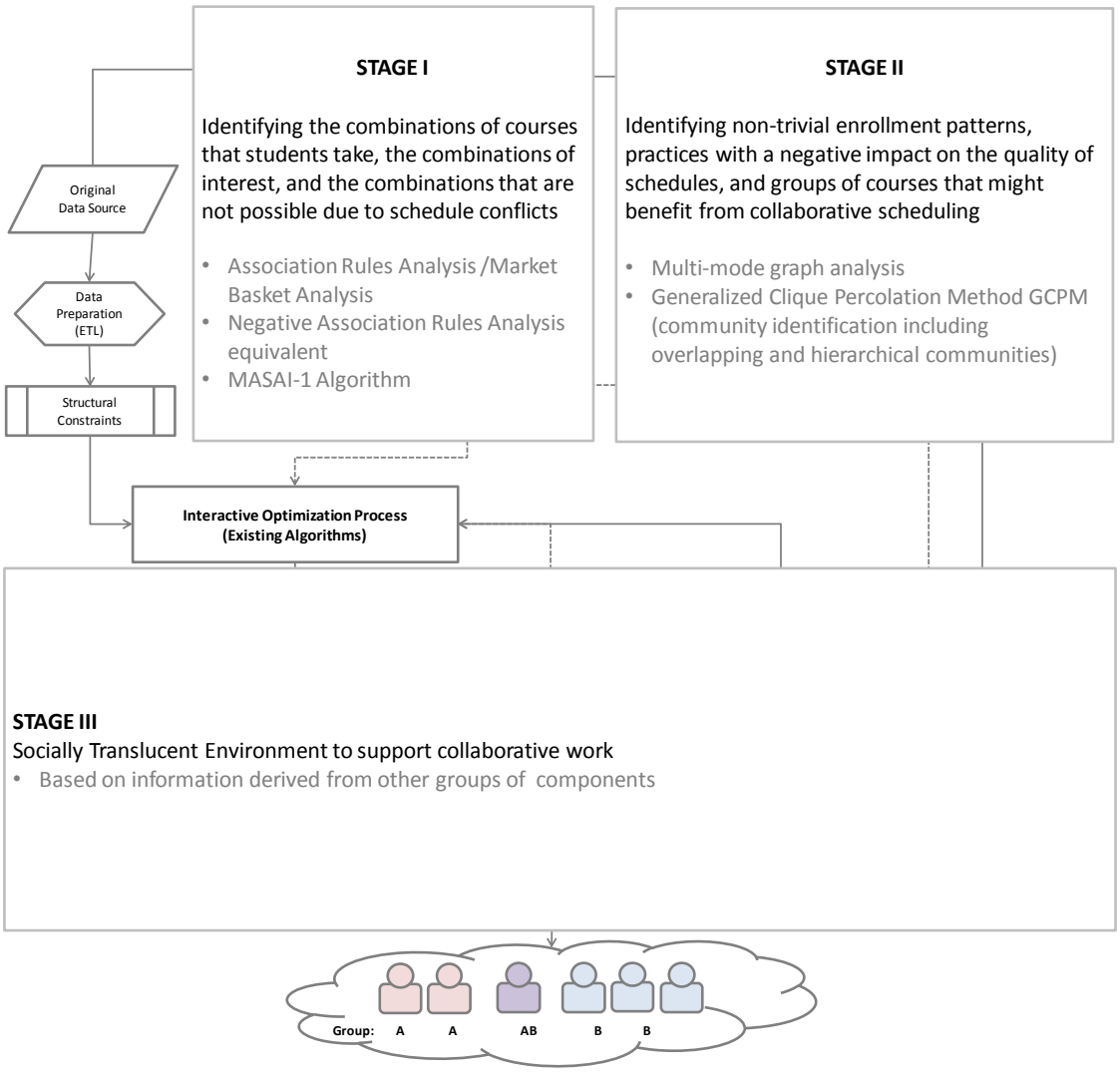


Figure 3 Overview Diagram of Proposed Architecture

4.0 IDENTIFYING COMBINATIONS OF COURSES OF INTEREST

4.1 ASSOCIATION RULE ANALYSIS

If we want to maximize the course enrollment options available to students, it stands to reason that the first step is to use historical data on recent terms to identify the combination of courses that students take from the available options. To that end, we start by modeling the problem as an Association Rule Analysis.

Following is a specification of the association rules problem adapted from the definition of Han, Kamber and Pei (2006). A course c_i is an item and a set of courses $C = \{c_1, c_2, \ldots, c_n\}$ is an itemset. Itemsets of size n are called *n-itemsets*. The set of courses that a student enrolls in during a term is a transaction t. Each transaction is a set of items such as that $t \subseteq I$, where I is the set of all courses offered (or in consideration) during a term. The set of all transactions per term is T, where $t \subset T$ (Han, Kamber, & Pei, 2006).

If A and B are itemsets: A transaction t contains A if and only if $A \subseteq t$. An association rule is an implication of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, and $A \cap B = \emptyset$. The percentage of transactions in T that contain $A \cup B$ is the support s for the rule $A \Rightarrow B$. The percentage of transactions in T containing A that also contain B is the confidence c of the rule $A \Rightarrow B$. In short,

$$\text{support } (A \Rightarrow B) = \textit{Probability of } A \cup B = P(A \cup B)$$
$$\text{confidence } (A \Rightarrow B) = \textit{Probability of } B \textit{ given } A = P(B/A)$$

Rules that satisfy both a given minimum support threshold and a given minimum confidence threshold are called *strong rules*. An itemset that satisfies a given minimum support threshold is a *frequent* itemset.

A well-known challenge in mining of frequent itemsets is that the results can include large numbers of itemsets, especially when the support thresholds are low. That is the case with the data sets generated from course enrollment data. In all academic terms in the study data set, approximately 40% of courses enroll a total of less than 20 students in all sections, and approximately 73% of courses enroll less than 50 students. About 73% of students take between

three and five subjects per term. These characteristics produce a transactions data set where the vast majority of transactions have very low support. More precisely, an average of 96% of transactions have less than 0.5% support, and the maximum support is under 10% for all the terms under consideration. Another factor that contributes to the generation of a large number of frequent itemsets is that, in Association Rule Analysis if an itemset is frequent, each of its subsets is also frequent (e.g. In a frequent itemset of three courses, all subsets of two courses are also frequent).

While the vast majority of transactions have and extremely low support, those transactions need to be considered. *First*, in higher education, scheduling of small courses is as important as scheduling of large courses. *Second,* we need to identify which of the transactions with a very low support might be of interest (i.e. they might have low support because a larger number of students cannot enroll in those course itemsets). Thus, the approach taken at this stage is to set support and confidence thresholds at 0% to get all course sets (i.e. all course sets are considered frequent) and then use the approach described ahead to find out itemsets of interests without using support or confidence measures. That means that we are using the association rules methodology only to identify all course itemsets at zero support and zero confidence levels.

4.1.1 Closed Itemsets and Closed Frequent Itemsets

If every item of set *I* is contained in set *Y,* but there is at least one item of *Y* that is not in *I*, then *I* is a subset of *Y*. That is, $I \subset Y$ and *Y* is a super-set of *I*. An itemset *I* in a data set *D* is closed if no superset of *I* has the same support as *I*. Additionally, *I* is a *closed frequent itemset* in *D* if I is *closed* and *frequent*. In the case at hand, where we are setting the minimum support threshold at 0%, all *closed itemsets* are *closed frequent itemsets*.

Closed frequent itemsets are used in order to produce smaller association rules sets and reduce redundancy while maintaining the same information contained in the whole set of frequent itemsets (Han et al., 2006). Two implementations of association rules algorithms were tried; namely SPSS Modeler (Mikut & Reischl, 2011) and Christian Borgelt's implementation (Borgelt, 2012). The latter is preferred in this case as it provides the option to obtain closed frequent itemsets.

4.1.2 Results of Association Rule Analysis

Table 1 ahead shows a summary of figures of interest derived from the data analysis for six recent fall terms at Pitt's Pittsburgh campus. Data includes enrollments in undergraduate courses taken for credit. As the set of courses that a student enrolls in during a term is a transaction, then the number of transactions is the same than the number of students enrolled in courses in the data set. As each student takes one seat on each enrolled course, then the average number of courses that each student enrolls per term is the number of seats divided by the number of transactions.

The last two columns to the right of Table 1 below show the number of frequent itemsets and closed frequent itemsets at 0% support and 0% confidence. The results confirm a substantial reduction in the number of itemsets that need to be processed in subsequent steps when using closed frequent itemsets.

Table 1 Summary of Association Rule Analysis for six recent fall terms at Pitt-Pittsburgh

Academic Term	Archive Period	Undergraduate Courses	Transactions	Seats	Avg. Courses per Student	Frequent Itemsets	Closed Frequent Itemsets
Sept. to Dec. 2008	2091	1,488	17,003	82,195	4.8	76,480	42,611
Sept. to Dec. 2009	2101	1,465	17,657	85,355	4.8	76,936	43,897
Sept. to Dec. 2010	2111	1,484	18,054	87,743	4.9	74,389	45,595
Sept. to Dec. 2011	2121	1,525	18,065	88,567	4.9	75,908	46,591
Sept. to Dec. 2012	2131	1,546	18,092	89,496	4.9	77,328	47,145
Sept. to Dec. 2013	2141	1,538	18,277	90,771	5.0	80,945	47,479

Table 2 below shows the count of all closed course itemsets by enrollment and archive period. Results show that the 86.1% of closed itemsets have enrolments under five students, with more than 63% having one or two students. This is expected given the discussion above on the fact that 96% of transactions have less than 0.5% support. These results also support the design decision to select a support threshold of zero in order to capture all itemsets and then identify those of interests using information that is not provided by the Association Rule Analysis alone. The selection of any support level above zero to filter out association rules would potentially lead to the loss of valuable information.

Table 2 Count of closed course itemsets by enrollment and archive period

	Count of all closed itemsets by enrolled students per academic term								
Enrollment per Itemset	**2091**	**2101**	**2111**	**2121**	**2131**	**2141**	**Total**	**% / Total**	**Cumulative %**
1	13,915	14,333	14,603	14,717	14,834	14,867	87,269	31.9%	31.9%
2	13,395	13,842	14,241	14,380	14,945	14,825	85,628	31.3%	63.3%
3	5,317	5,403	5,688	5,943	5,811	6,142	34,304	12.6%	75.8%
4	2,663	2,729	3,022	3,039	3,085	3,106	17,644	6.5%	82.3%
5	1,623	1,665	1,734	1,852	1,848	1,821	10,543	3.9%	86.1%
6	1,047	1,068	1,137	1,196	1,236	1,223	6,907	2.5%	88.6%
7	723	775	798	836	869	857	4,858	1.8%	90.4%
8	521	534	577	631	622	590	3,475	1.3%	91.7%
9	417	428	466	501	463	512	2,787	1.0%	92.7%
10 to 19	1,632	1,721	1,852	1,889	1,870	1,981	10,945	4.0%	96.7%
20 or more	1,358	1,399	1,477	1,607	1,562	1,555	8,958	3.3%	**100.0%**
Total	**42,611**	**43,897**	**45,595**	**46,591**	**47,145**	**47,479**	**273,318**	**100.0%**	

Table 3 below shows the distribution of closed itemsets based on the number of courses per itemset. Itemsets with two or three courses make up for an average of 59% of all closed itemsets over the six terms under consideration. An average of 89% of all itemsets have 5 or less courses.

Table 3 Distribution of closed itemsets per number of courses per itemset

Number of Courses per Itemset	**2091**	**2101**	**2111**	**2121**	**2131**	**2141**
2	39.07%	39.05%	38.70%	38.36%	38.28%	37.56%
3	19.65%	19.54%	20.32%	20.39%	20.82%	21.13%
4	12.93%	13.15%	12.84%	13.08%	12.73%	12.84%
5	18.04%	18.00%	17.45%	17.16%	17.01%	16.85%
6	8.75%	8.72%	9.00%	9.15%	9.23%	9.58%
7	1.36%	1.35%	1.48%	1.64%	1.66%	1.73%
8	0.18%	0.18%	0.19%	0.19%	0.26%	0.29%
9	0.01%	0.01%	0.02%	0.02%	0.01%	0.02%
Total	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**

Table 4 below shows information on the distribution of itemsets by the number of terms were they are present over the six terms under analysis. In the table, the column "Demand" refers to the sum of enrolled students per itemset for the terms when the itemset is present. Results indicate that a minority of itemsets drive the majority of demand. More precisely, while 10.3% of itemsets are present for three or more years, they correspond to 68% of the total demand.

To clarify, in Table 4 the sum of "Terms present" times the "Count of Distinct Itemsets" equals the total number of itemsets present over six terms (i.e. ∑ Terms present * Count of Distinct Itemsets) = Total number of itemsets over six terms). More precisely, from Table 2: 42,611 + 43,897 + 45,595 + 46,591 + 47,145 + 47,479 = 273,318; and from Table 4: 6*4,116 + 5*3,174 + 4*4,908 + 3*7,622 + 2*17,458 + 155,338 = 273,318.

Table 4 Distribution of itemsets by number of terms present

Terms present	Count of Distinct Itemsets	% / Total	Cumulative %	Demand	% / Total	Cumulative %
6	4,116	2.14%	2.14%	497,215	40.30%	40.30%
5	3,174	1.65%	3.78%	115,471	9.36%	49.65%
4	4,908	2.55%	6.33%	129,333	10.48%	60.13%
3	7,622	3.96%	10.29%	96,804	7.85%	67.98%
2	17,458	9.06%	19.35%	123,775	10.03%	78.01%
1	155,338	80.65%	**100.00%**	271,333	21.99%	**100.00%**
Total	**192,616**	**100.00%**		**1,233,931**	**100.00%**	

Table 5 below shows the results of the Association Rule Analysis for the top 20 closed frequent itemsets for academic term 2141 (Sept. to Dec. 2013) in terms of students enrolled in the itemsets. For instance, 505 students enrolled in sections of the courses ARTSC_CHEM_0310 and ARTSC_CHEM_0330 (ITEMSET_ID 47259) in term 2141.

Table 5 Top 20 Closed itemsets by enrollments for academic term 2141 (Sep. to Dec. 2013)

ITEMSET_ID	ARCHIVE_PERIOD	CLOSED ITEMSETS	ENROLLED
47429	2141	ARTSC_BIOSC_0050 **ARTSC_BIOSC_0150**	940
47476	2141	ARTSC_CHEM_0110 ARTSC_FP_0001	631
47462	2141	**ARTSC_BIOSC_0150** ARTSC_CHEM_0110	590
47473	2141	ARTSC_ENGCMP_0200 ARTSC_FP_0001	545
47449	2141	ARTSC_BIOSC_0050 ARTSC_CHEM_0110	516
47466	2141	**ARTSC_BIOSC_0150** ARTSC_FP_0001	509
47259	2141	ARTSC_CHEM_0310 ARTSC_CHEM_0330	505
47479	2141	ARTSC_FP_0001 ARTSC_PSY_0010	497
47438	2141	ARTSC_BIOSC_0050 **ARTSC_BIOSC_0150** ARTSC_CHEM_0110	493
47289	2141	ARTSC_MATH_0220 ARTSC_PHYS_0174	469
46927	2141	ENGR_ENGR_0011 ENGR_ENGR_0081	466
47109	2141	ARTSC_PHYS_0174 ENGR_ENGR_0081	442
47452	2141	ARTSC_BIOSC_0050 ARTSC_FP_0001	437
47442	2141	ARTSC_BIOSC_0050 **ARTSC_BIOSC_0150** ARTSC_FP_0001	433
47469	2141	ARTSC_CHEM_0110 ARTSC_ENGCMP_0200	428
46944	2141	ARTSC_PHYS_0174 ENGR_ENGR_0011 ENGR_ENGR_0081	408
47475	2141	ARTSC_ENGCMP_0200 ARTSC_PSY_0010	398
46428	2141	ARTSC_CHEM_0960 ENGR_ENGR_0081	371
47478	2141	ARTSC_CHEM_0110 ARTSC_PSY_0010	360
47422	2141	ARTSC_CHEM_0110 ARTSC_MATH_0220	358

So far, the analysis of Pitt's enrollment data indicates that:

- For the six fall terms under consideration, the number of closed itemsets per term averages 45,553 with a standard deviation of just 1,935, and there is a slight linear upward trend with the number of closed itemsets going from 42,611 to 47,479 over the six fall terms. These figures coincide with the increase in enrollments and number of sections offered over the same period of time (not shown in tables).
- The vast majority of itemsets have five or less courses (89%)
- The vast majority of itemsets enroll five or less students (86%). Furthermore, more than 63% of itemsets enroll one or two students. This is expected given that 96% of transactions have less than 0.5% support, which results from approximately 40% of courses enrolling less than 20 students in all sections; and approximately 73% of courses enrolling less than 50 students.

This result highlights the importance of considering the combination of courses that students enroll as most of them have what could be labelled as a "unique experience" in terms their specific combination of courses for a term.

- While 10.3% of itemsets are present for three or more years, they correspond to 68% of the total demand. This result, in combination with the previous one suggests that there is a relatively small dominant set of course itemsets that most students enroll, along with a large number of itemsets with smaller numbers of students enrolled.

The previous observations highlight several lines of inquiry that have not been considered in timetabling in higher education and that will be further explored ahead. They include:

- Is the number of students enrolled in each itemset the maximum possible or are there seats left available? For instance, did 505 students enrolled in ARTSC_CHEM_0310 and ARTSC_CHEM_0330 because no more students were interested in taking both courses together or because it was not possible for more students to enroll in both during the same academic term? In the first case, there is extra capacity in the itemset. In the second case, a capacity limit has been reached due to enrollment limits in sections or to schedule conflicts. In the latter case, decision makers would need consider if a higher number of effective seats or a different scheduling in this particular combination of subjects might need to be offered as it is a *schedule bottleneck* that reduces the enrollment options for students.

  This aspect of the problem corresponds to the identification of association rules or itemsets of interests. There is a voluminous literature on the topic of identifying association rules of interest that focuses on different ways of processing the data at hand in the transaction data sets. There is also literature that shows that those methods have limitations and that different methods render different association rules of interests on the same data set (Raeder & Chawla, 2011). The approach proposed in Section 4.3 to identify the course itemsets of interests enhances the transaction data sets, which are at the level of courses, by considering the individual sections offered on each course that is part of an itemset. The algorithm MASAI presented in Section 4.3 uses that extra information to identify the closed itemsets of courses that have reached the maximum number of seats possible.

- The Association Rule Analysis reveals closed frequent itemsets, which are derived from actual enrollments. Are there combinations of courses that might be of interest and are not possible to enroll due to schedule conflicts and are thus not appearing in transactions?

  This aspect of the problem corresponds to the identification of negative association rules. It appears that, due to its complexity, it has been less explored in the literature than the identification of positive association rules (Mani, 2012; Wu et al., 2004; Yuan et al., 2002). In the case at hand, we need to identify courses that do not appear together in transactions, which can happen due to three reasons: *First*, schedule conflicts; *second*, that students are not interested in taking the courses together, and *third* that courses frequently have pre-requisites. Section 4.4, presents the proposed approach to handle this aspect of the problem.

- Courses appear in multiple itemsets. For instance, ARTSC_BIOSC_0150 is highlighted in Table 5 appearing in five of the top 20 itemsets by enrollments. Are there offerings of courses that are of more importance than others in limiting or expanding the enrollment options for students?

  The Association Rule Analysis does not provide information regarding which courses or groups of courses appearing in multiple itemsets might be the most important in determining the quality of the whole schedule of classes. It does not provide information that would facilitate the understanding of complex enrollment patterns. The multi-mode graph analysis and community identification methodology proposed in Section 5 focus on these aspects of the problem.

4.2 RELATIONAL SCHEMA TO SUPPORT DATA PROCESSING ACROSS COMPONENTS

The denormalized relational schema illustrated in Figure 4 ahead is proposed to facilitate storing and processing of data and results. A prototype version has been implemented in an ORACLE 11g relational database and includes the following tables:

- TRANSACTIONS: The first processing step is to extract enrollment data from the institution's system of record (e.g. data warehouse). Data is loaded into a relational database table that contains the transactions for the academic terms under consideration. The details of the Extraction, Transformation and Loading (ETL) process depends on the implementation specifics of the system of record.

  As a summary of the definitions presented in the previous section, a transaction is the set of courses that a student enrolls in during a term. A course is identified by the school or academic unit that offers it, a subject of study and a catalog number. A section is an instance of a course offered in an academic term. Transactions are uploaded into table TRANSACTIONS in the relational schema. It enables an efficient way to explore the data and pass it to association rules algorithms.

  Each record in TRANSACTIONS identifies the registration of a student in one course for one term. A transaction is then the set of records with the same transaction identifier (TRANSACTION_ID) and academic term identifier (ARCHIVE_PERIOD). Table 6 below shows a sample of three transactions for one of the fall terms under study. In the table, the two records with ARCHIVE_PERIOD = 2141 and TRANSACTION_ID =2 show that during archive period 2141, the student with ID #####29 enrolled in the courses Chemistry 0110 section with class number 10332, and History 1614 section with class number 21155, both offered by the Dietrich School of Arts and Sciences (ARTSC_CHEM_0100 and ARTSC_HIST_1614).

Table 6 Sample of three transactions

ARCHIVE_PERIOD	STUDENT_ID	TRANSACTION_ID	COURSE	CLASS_NBR
2141	#####69	1	ARTSC_MATH_0220	17362
2141	#####29	2	ARTSC_CHEM_0110	10332
2141	#####29	2	ARTSC_HIST_1614	21155
2141	#####47	3	NURS_NUR_1221	20265
2141	#####47	3	NURS_NUR_1227	20267
2141	#####47	3	NURS_NUR_1233	20268

- UGRD_CLOSED_ITEMSETS_ST: This is a staging table. Data from the table TRANSACTIONS is used as input to the software used for Association Rule Analysis. In this case we used Borgelt's implementation, which renders flat files with the closed frequent itemsets (Borgelt, 2012). Results are then loaded into the staging table UGRD_CLOSED_ITEMSETS_ST in one record per itemset. The prefix "UGRD" indicates that the table contains data on undergraduate sections; the same applies to the naming convention on other tables. The table fields include an itemset identifier, term (ARCHIVE_PERIOD), the number of students enrolled in the itemset and a field for each course in the itemset.
- UGRD_CLOSED_ITEMSETS: This table stores results produced by the algorithm MASAI, discussed in Section 4.3 ahead. MASAI computes the maximum number of seats possible on each closed itemset. This table includes fields coming from the staging table: itemset identifier fields (ITEMSET_ID, and ARCHIVE_PERIOD), and the itemset course in a single string form (ITEMSET). It also provides fields to store the maximum capacity possible on each itemset at the beginning and end of the enrollment season, and the number of iterations required to obtain those figures.
- UGRD_SECTIONS: Contains detail data on undergraduate sections offered for the terms under consideration. These data includes section identifiers (e.g. ARCHIVE_PERIOD for academic term, course subject, catalog number, course title, section identifier, etc.), schedule information (e.g. days, times, number of meetings per week, etc.), capacity information (enrollment cap and enrollment total), and other information (e.g. classroom).
- UGRD_ITEMSET_COURSES: This table enables the implementation of the many-to-many relationship between closed itemsets and sections. More precisely, one closed itemset has two or more courses, and a course –and all its sections—can be in multiple closed itemsets.
- UGRD_ITEMSET_SECTIONS: Is a materialized view created by joining the previous three tables. It contains the fields and data at the level of individual sections and itemsets that are needed to compute the maximum capacity that is possible on each itemset. The latter is performed using the algorithm discussed in the following section.

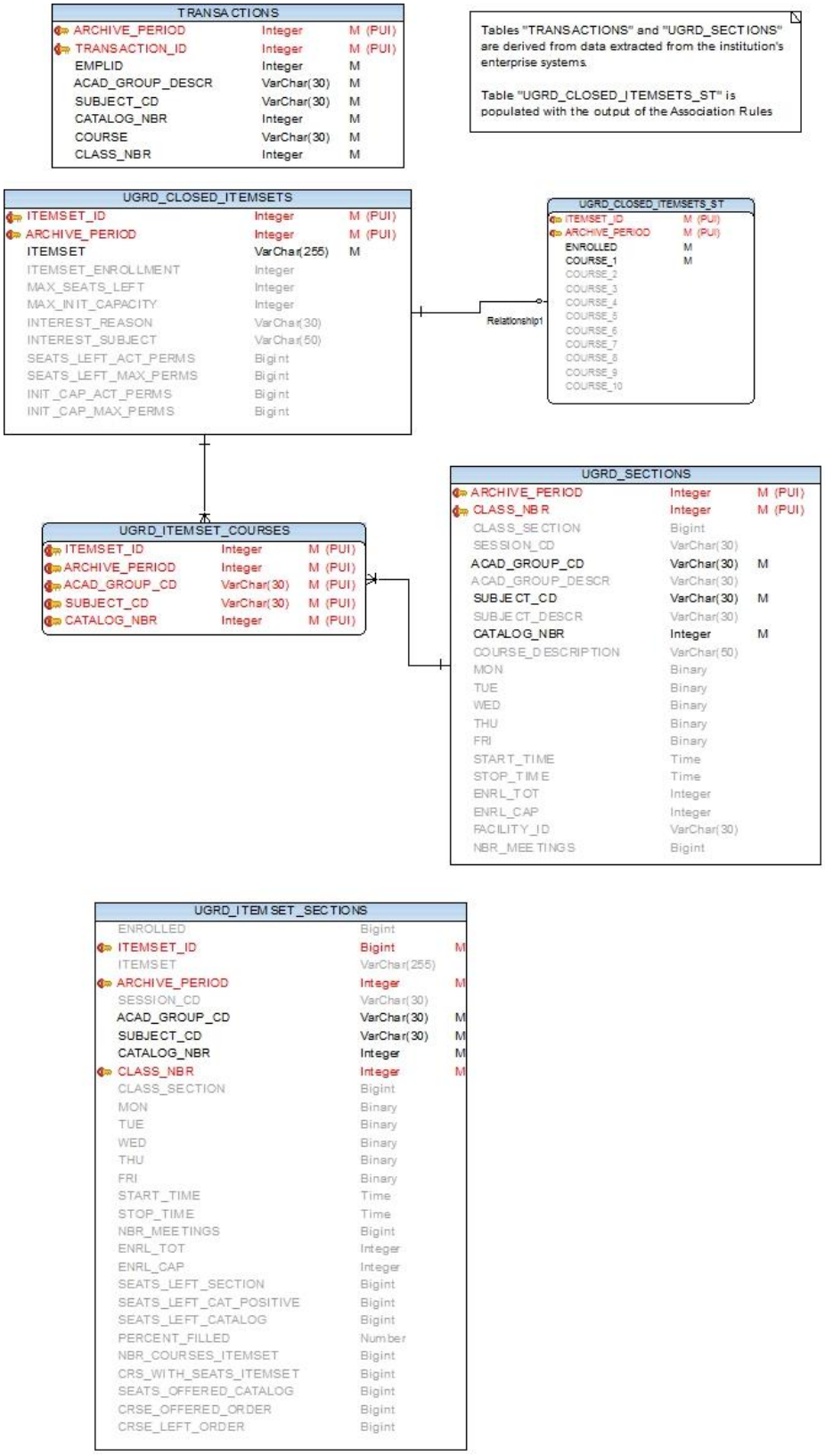


Figure 4 Denormalized relational schema to support data processing

4.3 MASAI: AN ALGORITHM TO DETERMINE THE MAXIMUM AVAILABLE SEATS PER COURSE ITEMSET

This section presents the MASAI algorithm including sub-sections on complexity analysis, the details of the algorithm, results obtained with a prototype implementation, and a sample case analysis.

4.3.1 Complexity Analysis

This sub-section discusses the complexity associated with determining the maximum number of seats possible in each closed itemset found per term. The following sub-sections present an algorithm that performs that task and elaborate on the results using a prototype implementation in PL/SQL. Pitt's real enrollment data is used as study case.

The association rules algorithm renders itemsets at the level of courses along with the actual enrollment in each itemset per term. For instance, during term 2111 (Sept. to Dec. 2010) a closed itemset is (ARTSC_CHEM_0960, ARTSC_MATH_0220, ARTSC_PHYS_0174). The output of the Association Rule Analysis indicates that 247 students enrolled in these three courses during term 2111. Would have it been possible for more students to enroll in those three courses?

For tem 2111 there were 10 sections of course ARTSC_CHEM_0960 offered at different weekly schedules. There were 16 sections of course ARTSC_MATH_0220 and seven sections of course ARTS_PHYS_0174. In order to determine the maximum number of seats possible in the itemset, it is necessary to consider the number of seats offered in individual sections of each course, their schedules, and the possible combinations of sections of each one of the three courses that could be taken together in this itemset in non-conflicting schedules.

The data set used for this work includes six terms of undergraduate enrollment data at Pitt's Pittsburgh campus during the fall terms between calendar years 2008 and 2013. The Association Rule Analysis identifies approximately 46,000 closed itemsets per term at the level of courses. A course is defined by the fields School, Subject and Catalog Number, i.e. ENGR_CHE_0100 refers to a course offered at the School of Engineering (ENGR) with subject

Chemistry (CHE) and Catalog Number 0100. In order to find the itemsets of interest, it is necessary to first identify the maximum number of seats offered and taken in each closed itemset. To that goal, we need to explore all the closed itemsets at the level of individual sections of each course, considering actual enrollments, enrollment limits and schedules.

Table 7 below helps to illustrate the idea with the case of the itemset including three courses offered by the Swanson School of Engineering offered in the fall of 2012 (ARCHIVE_PERIOD = 2131). These courses are Foundations of Chemical Engineering "ENGR_CHE_0100", Foundations of Chemistry Laboratory "ENGR_CHE_0101", and Probability and Statistics for Engineers 1 "ENGR_ENGR_0020". In this simple case, it is straightforward to see that there was a maximum of 70 effective seats possible in the itemset at the beginning of the term based on the offered seats (column ENRL CAP). More precisely, assuming that all seats available are taken by students enrolling only in this itemset, then a maximum of 70 students would had been able to enroll in sections of the three courses in the itemset. After the enrollment period is closed (add/drop), six students were enrolled in the itemset (not shown in Table 7), and no more seats in the itemset were available. That is, even though there were still seats available in sections of each of the three courses (column "Seats Left Section"), it was not possible for more students to register in the three of them (Sánchez, 2014).

Table 7 Enrollments in individual sections of a sample course itemset with no seats left available

Course Number	ACAD GROUP CD	SUBJECT CD	CATALOG NBR	CLASS NBR	Days	Start Time	Stop Time	Seats Left Section	Seats Left Catalog	Percent Filled	ENRL CAP
1	ENGR	ENGR	0020	14327	M W	16:00	17:15	-1	10	101.4	70
1	ENGR	ENGR	0020	14443	T H	9:30	10:45	0	10	100.0	70
1	ENGR	ENGR	0020	14566	T H	9:30	10:45	11	10	84.3	70
2	ENGR	CHE	0100	14484	M W F	8:00	9:50	9	18	86.2	65
2	ENGR	CHE	0100	23971	M W F	8:00	9:50	9	18	86.2	65
3	ENGR	CHE	0101	14485	H	8:00	9:50	12	25	81.5	65
3	ENGR	CHE	0101	23963	T	8:00	9:50	13	25	80.0	65

In the worst case scenario (Standish, 1995), to find the maximum number of seats offered it is necessary to explore all the sets of three sections of courses in the itemset. For the sample itemset, 12 sets of three sections need to be explored (3*2*2). Then, for each of the 12 sets of three sections, it is necessary to check all the combinations of two sections for schedule collisions and capacity. The reason is that in order to be possible to enroll in a set of *m* sections, each combination of two sections in the set needs to be free of schedule conflicts. For a set of three sections, there are three possible combinations of two sections (binomial coefficient (3 2)). Thus, for the sample shown in Table 7, in the worst case scenario 36 iterations are required to find out the maximum number of seats possible in the itemset (3 * 2 * 2 * binomial coefficient (3 2)).

In the general case, if there are *m* courses in an itemset, with each course having *k* sections offered where *k* is a positive integer (i.e. k_j is the number of sections in course m_j), then the worst case scenario for the number of iterations that need to be performed to find out the maximum number of seats possible in an itemset is

$$I = k_1 * k_2 * \ldots\ldots * k_m * \text{binomial coefficient } (m\ 2) = (k_1 * k_2 * \ldots\ldots * k_m) * (m*(m-1)/2)$$

The computation of I has a complexity $O(k_{max}^{m})$, where k_{max} is the maximum k in an itemset. The total number of iterations that need to be performed to process all the itemsets in a term is the sum of the total iterations per itemset, and thus have the same complexity

Figure 5 below shows a histogram on the $\log_{10}$ of the worst case scenario for the number of iterations required to find the maximum number of seats per itemset at the beginning of the enrollment period (initial capacity or initial seats) for the fall term of 2013 (archive period 2141 in Pitt's notation). Other terms in the data set being used follow the same extremely skewed distribution pattern, and thus for brevity those histograms are omitted. In the fall of 2013 there were 47,479 closed itemsets resulting from the Association Rule Analysis. 22,405 (47.2%) of those itemsets would require 50 or less iterations to find the maximum number of seats. Only 24 (0.05%) itemsets make up for 50% of the required iterations. In the worst case scenario, the total number of required iterations to determine the total seats capacity in all itemsets at the beginning of the enrollment period would be 1.348E+11.

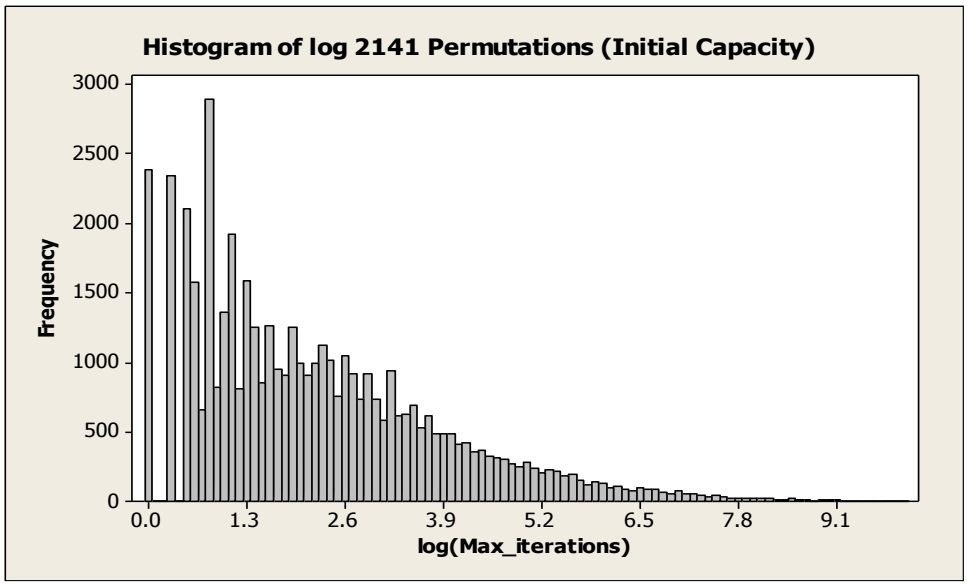


Figure 5 Histogram of the worst case scenario for number of iterations required to find the maximum number of initial seats per course itemset for a sample term (log10)

Figure 6 below shows a histogram on the $\log_{10}$ of the worst case scenario for the number of iterations required to find the maximum number of seats per itemset at the end of the enrollment period (capacity left or seats left available) for the fall term of 2013 (archive period 2141 in Pitt's notation). As in the previous case, other terms in the data set being used follow the same extremely skewed distribution pattern, and thus for brevity the histograms for other academic terms are omitted. In this case, the set of itemsets include those remaining after filtering out all sections with no seats left available. This renders a set for processing that includes 23,153 itemsets to be processed for the fall of 2013. 15,570 (67%) of those itemsets would require 50 or less iterations to find the maximum number of seats available in the itemset. Only 19 (0.1%) of those itemsets make up for 50% of the total number of iterations required in the worst case scenario. In this scenario, the total number of required iterations to determine the total seats capacity in all itemsets for the term would be 2.89E+09.

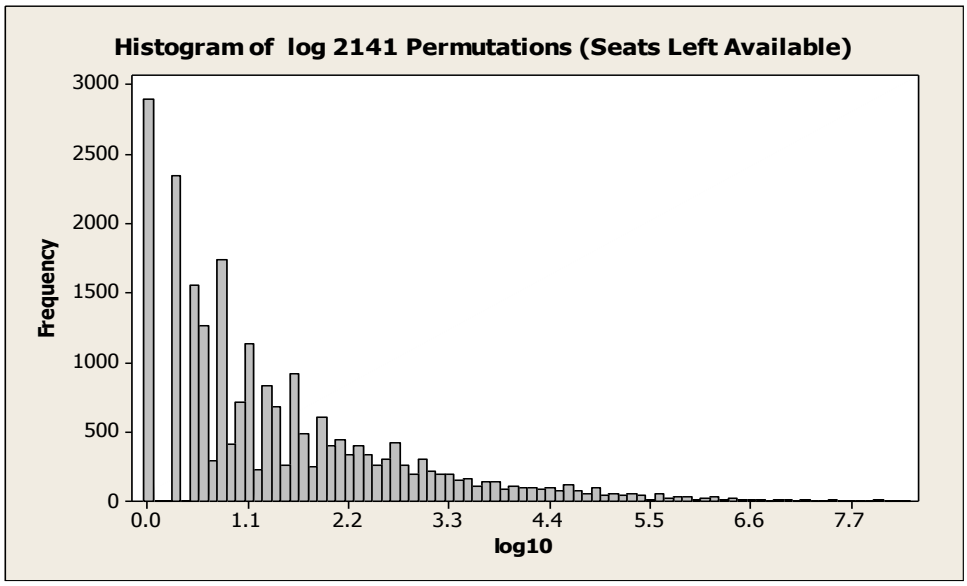


Figure 6 Histogram of the worst case scenario for the number of iterations required to find maximum number of seats left available per course itemsets for a sample term (log10)

4.3.2 MASAI Algorithm

A backtracking algorithm has been developed to find the ***ma***ximum number of ***s***eats ***a***vailable per ***i***temset. It is called MASAI. A prototype implementation of MASAI was done in ORACLE's PL/SQL and used to test results with Pitt's actual enrollments data set.

MASAI is designed to take input from a relational database table or flat file using the schema described in section 4.2 and write results back to it. In the current set up, it records results in one of the tables in the same schema, which facilitates further processing discussed in sections ahead. MASAI can receive as input all itemsets, a partial number of itemsets, or one itemset from a term, and compute the number of maximum possible seats in each one of itemsets provided as input.

Processing proceeds one itemset at a time and assumes that all seats available in each section are available for the itemset being processed. That is, it does not consider that seats in a section can be taken in multiple itemsets. If we were to relax the referred assumption and device a more complex approach that somehow splits the seats available across multiple itemsets, then the available capacity for the itemset under consideration would be lower than assuming that all

seats are available for the specific itemset under consideration. Thus, as we are looking for course itemsets with no seats left available, this represents the conservative approach. A consideration is that for the case of the computation of initial capacities per itemset, figures are overestimated as at the beginning of the enrollment period all sections have all seats available.

MASAI could be used in several scenarios within the proposed architecture including:

- Processing of all closed itemsets at the beginning of a term to determine the maximum offered capacities per course itemset.
- Processing of all closed itemsets at the end of the enrollment period of a term, or when enrollments are still open to determine the maximum number of seats left available per course itemset at any time.
- Processing of one or more itemsets provided as input to determine the maximum number of seats possible. In this scenario, and with a user interface, the algorithm could interactively support users on the design of timetables. It could also work in tandem with an existing optimization algorithm. In the latter case, if minimum capacities for certain itemsets are specified as constraints, the optimization algorithm could pass requests to MASAI for verification of constraint thresholds satisfaction during processing.
- Processing of any itemset provided as input in the required format. This enables the identification of combinations of courses that cannot be enrolled together in a term as discussed in Section 4.4 ahead. More precisely, closed itemsets are obtained using enrollment data. Thus, they do not provide data on combinations of courses that cannot be taken together due to schedule conflicts as no student is able to enroll in those combinations (i.e. there are no transactions on those combinations of courses). In this scenario, the input could include sets of courses of interest. Those with a resulting maximum capacity of zero would be the itemsets that are not possible. For instance, input could include tuples sets of courses that apply for General Education Requirements, or include the set of all courses with equivalent academic level in a department or multiple departments (e.g. all introductory engineering courses along with all introductory courses in the department of mathematics).

MASAI processes all the closed itemsets per term in polynomial time. It does so using an approach that substantially reduces the number of iterations that need to be checked per itemset in order to obtain the maximum number of seats possible. The proposed approach combines:

- An initial filtering of itemsets that are extreme outliers in terms of expected number of iterations required when computing initial maximum capacities. In the prototype implementation this step consisted of filtering out itemsets with an expected number of iterations beyond 10 standard deviations from the average. It resulted in the removal of an average of 23 itemsets per term when computing initial capacities. This step filters out itemsets that would require in average 61% of all iterations in the worst case scenario, while removing only 0.05% of the itemsets.

    - In the case of the computation of the maximum number of seats left available at the end of the enrollment period, no itemsets were filtered during pre-processing. The reason is that in the worst case scenario for this case, the expected number of iterations is two orders or magnitude lower than in the case of the computation of initial capacities.

- For the case of computation of capacities per itemset at the end of the enrollment period, itemsets where one or more of the courses have no seats left available in any of its sections are excluded. In these cases the number of seats left in the itemset is zero and thus there is no need for further processing.
- Sections with no seats left available or oversubscribed are removed when computing the number of seats left available per itemset at the end of the enrollment period. That is, a course with multiple sections might have seats left in some of them at the end of the enrollment period. In that case, sections with no seats left can be filtered out from further processing while keeping in the set only the sections that still have seats available.
- The backtracking mechanism implemented in the algorithm described ahead. It processes the sections remaining in the input set while reducing the number of required iterations where possible, by skipping iterations that won't contribute to the final result.
- A timer implemented in the algorithm that can be set to stop processing itemsets beyond a pre-specified time limit. For the case of the Pitt enrollment data, after several rounds of

testing in a development environment, the timer was set to 20 seconds for the computation of maximum initial seats possible per itemset, and 10 seconds for the case of maximum seats available per itemset at the end of the enrollment period.

The materialized view UGRD_ITEMSET_SECTIONS described in Section 4.2 is used to generate the record set that is passed as input to MASAI. In the PL/SQL implementation, after the steps described in the first three bullets above are performed, using standard relational database operations, a cursor is used to read the remaining records. The input data set has one record per section and includes extra fields that the algorithm uses for its operation as detailed ahead. Table 8 below shows a sample input record using the itemset shown in Table 7 (i.e. ARCHIVE_PERIOD: 2131, ITEMSET_ID: 33374). Sections with no seats left (i.e. the number of seats left in the section is zero or negative in the case of oversubscribed sections), or itemsets where one of the courses has all its sections full are filtered out from the input records as they do not contribute to the computation.

Table 8 Sample of input record for MASAI

COURSE	SECTION	ARCHIVE_PERIOD	ITEMSET_ID	CRSE_NBR	CRSES_IN_ITEMSET	SEATS_BALANCE_SECT	SEATS_BALANCE_CAT	MON	TUES	WED	THURS	FRI	START_REF	STOP_REF
ENGR_CHE_0101	1	2131	33374	1	3	12	25	N	N	N	Y	N	0.333333	0.409722
ENGR_CHE_0101	2	2131	33374	1	3	13	25	N	Y	N	N	N	0.333333	0.409722
ENGR_CHE_0100	1	2131	33374	2	3	9	18	Y	N	Y	N	Y	0.333333	0.409722
ENGR_CHE_0100	2	2131	33374	2	3	9	18	Y	N	Y	N	Y	0.333333	0.409722
ENGR_CHE_0020	1	2131	33374	3	3	11	10	N	Y	N	Y	N	0.395833	0.447917

In addition to the identifiers, the following fields are included in the input:

- CRSE_NBR: Courses in each itemsets are sequentially numbered and included in ascending order.
- CRSES_IN_ITEMSET: Total number of courses in the itemset.
- SEATS_BALANCE_SECT: Balance of seats left available in the section. MASAI starts with the provided value and then decreases that value as it evaluates possible combinations of

sections that could be taken within the itemset. Input is provided in ascending order to guarantee that sections with the lowest number of seats available are processed first, as they are the most restrictive.

- SEATS_BALANCE_CAT: Balance of seats left in the course. That is, the total number of seats left in all sections offered in the course. MASAI starts with the initial provided value and then decreases it during processing as it evaluates possible combinations of sections that could be taken within the itemset. Input is provided in ascending order to guarantee that courses with the lowest number of total seats left across its sections are processed first.
- MON, TUES, WED, THURS, FRI: Provide Yes/No values identifying if a section is offered on a day of the week.
- START_REF, STOP_REF: Start and stop daily time of weekly sessions for each section. Figures are captured in decimal format to facilitate processing.

Output data is recorded in selected fields of table UGRD_CLOSED_ITEMSETS in the relational schema. The main output is the maximum seats possible per itemset. There are other fields that are used to record additional output that can be used for further analysis (e.g. number of iterations required per itemset, and processing time per itemset).

The main idea is to process all possible combinations of sections per itemset and add the possible seats in each combination to a running total per itemset. Processing per itemset starts with the sections with the lowest capacity available as they are the most restrictive, and uses a backtracking mechanism to skip combinations that do not lead to an increase of the maximum capacity (e.g. If there are no more seats available in the current section or the section that will come in the next iteration then skip it). During iterations the seats available in sections are decreased as possible combinations are considered. When no seats are left available in a section, then it is skipped when processing other combinations that include the section.

Using as illustration the small case shown in Table 5, we need to iterate through the combinations of five sections in three courses (i.e. course number one has two sections, course number two has two sections, and course number three has one section). At each iteration step we take one section from each course for further processing. This is shown in the pseudo code segment below, which does not yet show the backtracking mechanism. For the sample case, at every iteration step of the inner-most loop we obtain a set of three sections for further

processing. In the general case, when processing an itemset with *k* courses, we need to iterate through *k* loops. At every step of the inner-most loop we obtain a *k*-tuple of sections for further processing.

```
j1 = 1;
while  j1 <= number of sections in course number 1 loop
        j2 = 1;
        while  j2 <= number of sections in course number 2 loop
                j3 = 1;
                while  j3 <= number of sections in course number 3 loop
       --- Further processing for sets of three sections at this iteration step;
                    j3 = j3 +1;
                end loop;
                j2 = j2 +1;
        end loop;
        j1 = j1 +1;
end loop;
```

Sets of sections obtained at each iteration step of the inner-most loop are processed considering every combination of two sections to identify if they have schedule conflicts. The reasoning is that it is possible for students to enroll in all sections in a set of sections of any size, only if it is possible for them to enroll in every couple of sections in the set. For instance, it is possible to enroll in a set of three sections only if it is possible to enroll in every combination of two sections in the set. For the sample case at hand, when $j_1 = j_2 = j_3 = 1$, the first section of each course is taken to form a set for processing. Then, for each set of three sections, every combination of two sections needs to be checked for schedule collisions.

The process at this stage involves two steps: *First*, check if the two sections being processed have sessions scheduled to meet on the same day. If not, then there is no conflict; the two sections could be taken together, and processing continues. *Second*, if the two sections have sessions scheduled on the same day, then is necessary to check if they are offered at colliding times. If not, then the two sections could be taken together. The pseudo code for this part of the process is shown below.

```
for p in 1…k loop  --k is the number of courses in the set
  for q in 1….k loop
    if p < q then -- matrix is symmetric. Need to check only half of the cases
      if section(p) is offered at the same day than section (q) then
        if  (    section(p) start time >= section(q) stop time
            OR section(p) stop time <= section (q) start time) then
              do nothing ---no collision;
        else increase collision counter; --schedule collision
        end if ;
       end if;
      increase iterations counter;
     end if;
   end loop; --end q loop
  end loop; -- end p loop
```

After the previous step determines that a set of two sections does not have schedule conflicts and that there are still seats available in both sections; then a variable that holds the maximum capacity is increased by the balance of seats available in the section with the lowest number of seats available. The balance of seats available in the sections under consideration is then reduced accordingly.

Following is a discussion of the complete algorithm that identifies the maximum number of seats available per itemset per term. In the presentation ahead, the term “bundle” is used instead of the term “itemset”; the reason is that MASAI can process any set of courses (bundles) as opposed to only the itemsets produced by the Association Rule Analysis.

The core of the algorithm is

```
1 BEGIN
2  max_time = 10;  --set max time for individual itemset processing. e.g. 10 seconds
3  read_term_bundles; -- read data set
4  for i in 1…term_bundles.count loop --bundles in the input data set are processed one at a time
5         split_bundle(i);
6         process_bundle(i);
7         update_ugrd_bundles_tbl(i);
8 end loop;
9 END;
```

Line two sets a maximum processing time per bundle. It prevents the algorithm to run for times that can be unacceptably long in case of bundles that still require large number of combinations to determine the maximum number of possible seats after the filtering and backtracking steps.

Line three includes a call to the procedure "read_term_bundles" that reads the data set. This is a standard procedure that reads data from a database table with the fields required per term (a cursor in PL/SQL) as previously discussed.

- Each record contains information on a section of a course in a closed itemset as produced by Association Rule Analysis or a bundle of courses of interest.
- While reading data, the procedure populates an auxiliary in-memory table with bundle_id and number of courses in each itemset (or bundle). These data are used later for processing.

Lines four to eight include a loop that reads a single bundle per iteration and includes calls to several procedures as follows:

- Line five includes a call to procedure "split_bundle(i)". This procedure splits the bundle records into individual in-memory tables where each table contains the records for an individual course in the bundle. This facilitates processing of all combination of sections in all courses in the bundle to find out maximum capacities.
- Line six includes a call to function "process_bundle(i)", which returns the maximum number of seats possible in the bundle. During processing it makes repeated calls to a procedure called "get_max_bdl()".
- Line seven includes a call to procedure "update_ugrd_bundles_tbl(i)". This procedure writes the results back to the database table UGRD_CLOSED_ITEMSETS.

Following is an explanation of the procedure "split_bundle(i)" and the function "process_bundle(i)" using the case of a bundle with three courses. The procedure "update_ugrd_bundles_tbl(i)" is not discussed as it is a standard table update. The PL/SQL

prototype implementation used to test results with actual enrollment data can handle bundles with up to ten courses and no limit in the number of sections per course.

process_bundle(i)

--In the case of a bundle with three courses there are three tables, each one contains the records for a course in the bundle (i.e. bundle_table1, bundle_table2 and bundle_table3). More precisely, each table has the records for all the sections in one course

max_seats_bdl = 0; *--initialize global variable that is counter for maximum seats in bundle*
term_bundles(i).permutations = 0; *--initialize count of permutations in array that holds processing data*
BEGIN
j_1 = 1; *--initialize loop counter*
<<outer>>
WHILE j_1 <= bundle_table1.count LOOP *--LOOP 1: loop through table1 records*
--//////////////
--Look forward: jump to next record if there are no seats in the current one and the current record is not the last record.
IF there are no seats left in table1.section AND this is not the last record THEN
$j_1 = j_1 + 1$; *--skip one record*
ELSIF there are no seats left in table1.section AND this is the last record THEN
exit;
END IF;
--//////////////////
j_2 = 1; *--initialize counter for next loop*
WHILE j_2 <= bundle_table2.count LOOP *–LOOP 2: loop through table 2 records*
--//////////////
--Look backwards: Check if there are seats left in section from table in previous loop.
IF there are not seats left in table1.section THEN
exit;
END IF; *– no seats left in section from table in previous loop*
--//////////
---Look forward: jump to next record if there are no seats in the current one and the current record is not the last record.
IF there are no seats left in table2.section AND this is not the last record THEN
$j_2 = j_2 + 1$; *--skip one record*
ELSIF there are no seats left in table2.section AND this is the last record THEN
exit;
END IF;
--//////////
j_3 =1; *--initialize counter for next loop*
WHILE j_3 <= bundle_table3.count LOOP *---LOOP 3: loop through table3 records*
--//////////////////
– Look backwards to check if there are seats left in sections from tables in previous loops

```
    IF there are not seats left in table1.section OR there are no seats left in table2.section THEN
      exit; –no seats left
    END IF;
    --//////////////////
   --Look forward: jump to next record if there are no seats in the current one and current
   record is not the last record.
   IF there are no seats left in table3.section AND this is not the last record THEN
     j3 = j3 +1;   --skip one record
   ELSIF there are no seats left in table3.section AND this is the last record THEN
     exit;
   END IF;
   --//////////////////////
  IF time processing < maximum time allowed THEN
     –Have not yet reached max allocated time per itemset
     --call function that computes the maximum number of seats left in current combination
     -- First integer argument refers to number of courses in bundle.
     -- Current function accepts up to ten courses in bundle (i.e. j1 to j10).
      get_max_bdl( 3, j1, j2, j3, 0, 0, 0, 0, 0, 0, 0, i) ;
   ELSE return -1; -- allocated processing time was exceeded;
   END IF;
   --///////////////
    j3 = j3 +1;
  END LOOP; --LOOP 3
  --///////////////
    j2 = j2 +1;
 END LOOP; --LOOP 2
 --///////////////
    j1 = j1 +1;
END LOOP outer;  --LOOP 1;
--///////////////
Return max_seats_bundle;
END;  --End process_bundle

-//////////////////////////////////////////////////////
get_max_bundle( n, k1, k2, k3, k4, k4, k6, k7, k8, k9, k10, i)
min_cap = 0;  --initialize minimum capacity
coll = 0; –initialize schedule collision counter
--///////////////
BEGIN
--copy record from each Ki section into table bdl_core_tbl (bundle core table).
-- while copying records get the minimum capacity available in a section in this group of sections
--Implementation includes the following five lines for each ki where 1<=i<=10
IF Ki > 0 THEN
  bdl_core_tbl ( bundle_tablei(ki).crse_nbr ) = bdl_tbli(ki)
  IF balance of seats in this section < min_cap THEN
     min_cap = balance of seats this section;
```

```
  END IF;
 --//////////////
Loop through the sections in the group checking every 2-tuple. Only half the tuples need to be checked due to symmetry
FOR p in 1….n LOOP
  FOR q in 1…n LOOP
   IF p < q THEN  --matrix is symmetrical, need to check only half of it
    -- A schedule collision occurs when sections are offered at the same day and time.
    -- First, check if there is a day collision in the sections tuple
          IF (      (bdl_core_tbl(p).mon = 'Y' AND bdl_core_tbl(q).mon = 'Y')
              OR  (bdl_core_tbl(p).tues = 'Y' AND bdl_core_tbl(q).tues = 'Y')
              OR  (bdl_core_tbl(p).wed = 'Y' AND bdl_core_tbl(q).wed = 'Y')
              OR  (bdl_core_tbl(p).wed = 'Y' AND bdl_core_tbl(q).wed = 'Y')
              OR  (bdl_core_tbl(p).fri   = 'Y' AND bdl_core_tbl(q).fri = 'Y') )
          THEN  --if section tuple is offered on the same day then need to check times
                IF     bdl_core_tbl(p).start_time >= bdl_core_tbl(q).stop_time
                  OR bdl_core_tbl(p).stop_time <= bdl_core_tbl(q).start_time
                THEN  null;   --no collision
                ELSE   coll = coll + 1; -- increase collision counter
                END IF;
           ELSE null; --no collision
           END IF;
       term_bundles(i).permutations = term_bundles(i).permutations +1; --increase iterations
       counter
    END IF;
END LOOP; --end q loop
END LOOP; --end p loop
--//////////////////////////////
-- Now, it is necessary to update the balance of seats in records
IF coll = 0 AND min_cap > 0
-- If there are no collisions and there are seats available in sections with lowest number of seats available in the group
  IF ki > 0 THEN
    --Do this for all Ki >0 in multiple if statements (could be done in a for loop as well)
     --subtract the minimum number of seats available in section group from the balance of seats
     in this section.
    bdl_tbl_i.seats_balance_sect = bdl_tbl_i.seats_balance_sect – min_cap;
    --subtract the minimum number of seats left available in catalog group from the balance of
     seats in this section.
     bdl_tbl_i.seats_balance_cat  = bdl_tbl_i.seats_balance_cat - min_cap;
 --////////////////
           Update variable holding maximum number of seats in bundle
             max_seats_bdl = max_seats_bdl + min_cap;
  END IF;
END IF;
END;
```

4.3.3 Results Obtained with a MASAI Prototype Implementation

Enrollment data from Pitt's Pittsburgh campus for six recent fall terms are used as case study to illustrate the results obtained with a prototype version of MASAI. Tables 9 and 10 below show the results obtained regarding the number of iterations required to find the maximum number of seats available per closed itemset per term. Table 9 shows the figures after the enrollment period has ended. Table 10 shows the figures at the beginning of each term.

In Tables 9 and 10 below, the worst case scenario refers to the maximum total number of iterations that would be required to compute the maximum number of seats available per term as discussed in Section 4.3.1. The tables also show the actual number of iterations that MASAI performs to compute the same figures. In Table 9, the figures are substantially lower than in Table 10 as at the end of the enrollment period is not necessary to consider the sections that are filled to capacity thereby reducing the required number of combinations that need to be processed. The tables also show the percentage of the actual number of iterations performed over the number of iterations required in the worst case scenario. For the case of the enrollment data under study, the processing approach and backtracking mechanisms implemented in MASAI require an average of 3.88% and 1.09% of the number of iterations expected in the worst case scenarios. These results indicate that the proposed approach is viable in support of a timetabling system as the actual computation time is bounded.

Table 9 Number of iterations required to find the maximum number of seats left available in closed itemsets per term

(Capacity Left : Seats Available After Enrollment Period has Ended)

ARCHIVE_PERIOD	Actual Iterations MASAI-1	Worst Case Scenario	% Actual / Worst Case
2091	42,266,618	1.019E+09	4.15%
2101	6,781,687	1.006E+08	6.74%
2111	76,347,203	1.408E+09	5.42%
2121	42,169,230	1.959E+09	2.15%
2131	93,570,126	3.113E+09	3.01%
2141	110,977,945	2.890E+09	3.84%

Table 10 Number of iterations required to find the maximum initial number of seats available in closed itemsets per term

(Initial Capacity: Seats Available at Beginning of Enrollment Period)

ARCHIVE_PERIOD	Actual Iterations MASAI-1	Worst Case Scenario	% Actual / Worst Case
2091	1,349,799,045	1.332E+11	1.01%
2101	1,196,259,184	7.428E+10	1.61%
2111	1,690,160,025	1.957E+11	0.86%
2121	1,569,320,710	1.360E+11	1.15%
2131	1,765,052,138	2.568E+11	0.69%
2141	1,664,737,699	1.348E+11	1.23%

Table 11 below shows the count of all closed itemsets by the number of seats left available at the end of the enrollment period for each of the six terms being considered. Results show that, in the case of Pitt, for every academic term under study, there are between 50% and 64% of closed course itemsets with not seats available at the end of the enrollment period. A result not shown in the table indicates that in average there is only 1% of itemsets with no seats available that have capacity left in all courses in the itemset. That is, in the case study at hand the limitations on enrollments appear to be derived mostly from capacity limits rather than from schedule conflicts. Conversely, there are between 36% and 50% of itemsets with seats available, and between 8.6% and 16.2% of itemsets with more than six seats left available. This results suggests that there might be opportunities for improvement in both directions.

Table 11 Distribution of itemsets by number of seats left available at the end of the enrollment period

Seats Left	2091	2101	2111	2121	2131	2141	2091	2101	2111	2121	2131	2141
-1 (*)	5		13	6	21	25	0.01%	0.00%	0.03%	0.01%	0.04%	0.05%
0	24,898	27,902	25,574	23,466	28,084	25,070	58.43%	63.56%	56.09%	50.37%	59.57%	52.80%
1	4,799	4,388	6,463	6,850	5,908	7,024	11.26%	10.00%	14.17%	14.70%	12.53%	14.79%
2	2,739	3,010	3,658	3,070	2,575	3,465	6.43%	6.86%	8.02%	6.59%	5.46%	7.30%
3	1,154	2,362	2,995	2,790	1,905	2,211	2.71%	5.38%	6.57%	5.99%	4.04%	4.66%
4	1,222	1,161	971	2,032	1,340	880	2.87%	2.64%	2.13%	4.36%	2.84%	1.85%
5	1,190	1,281	1,031	812	937	1,115	2.79%	2.92%	2.26%	1.74%	1.99%	2.35%
6 or more	6,604	3,793	4,890	7,565	6,375	7,689	15.50%	8.64%	10.72%	16.24%	13.52%	16.19%
Total	**42,611**	**43,897**	**45,595**	**46,591**	**47,145**	**47,479**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**

(*) Not completely processed: timer limit exceeded

Table 12 below shows the distribution of itemsets by the number of seats available at the beginning of the enrollment period per term (i.e. initial capacity). The capacity per itemset is computed assuming that all seats in every section of every course in the itemset are available for the itemset under consideration. In practice, as courses appear in multiple itemsets, the actual available capacity is split among them. Thus, the computed maximum capacity per itemset overestimates the total that will be effectively available during the enrollment period.

The capacity that is ultimately available per itemset depends on the speed at which sections are filled up during the enrollment period. Thus, a better indicator of capacity per itemset is the number of seats left at the end of the enrollment period as opposed to the capacity at the beginning of the enrollment period. The longitudinal analyses over the six terms of data presented ahead show that it is possible to identify itemsets that frequently run out of capacity even though at the beginning of the term it appears that there is plenty of seats capacity available. That analysis is further enriched with the multi-mode graph analysis presented in Section 5. In a production environment, one can envision that a daily run would be necessary to continuously monitor itemsets that are known to be of interest, and to identify all other itemsets that are running out of capacity. This would enable adjustments to the schedules when opportune and viable.

Table 12 Distribution of itemsets by number of seats available at the beginning of the enrollment period

Max Initial Capacity	2091	2101	2111	2121	2131	2141	Total	%/Total	Cumulative %
-1 (*)	195	175	304	290	345	326	1,635	0.60%	
0 to 9	1,138	1,015	367	394	352	409	3,675	1.34%	1.34%
10 to 24	5,315	5,412	5,359	5,458	5,185	5,324	32,053	11.73%	13.07%
25 to 49	10,793	10,728	11,212	11,058	12,165	11,421	67,377	24.65%	37.72%
50 to 99	10,406	11,786	12,337	12,518	12,438	12,944	72,429	26.50%	64.22%
100 to 199	7,841	8,399	8,919	9,300	8,608	9,269	52,336	19.15%	83.37%
200 to 299	4,727	3,723	4,192	4,662	4,616	4,761	26,681	9.76%	93.13%
300 or more	2,196	2,659	2,905	2,911	3,436	3,025	17,132	6.27%	99.40%
Total	**42,611**	**43,897**	**45,595**	**46,591**	**47,145**	**47,479**	**273,318**	**100.00%**	

(*) Not completely processed: Timer limit exceeded

Table 13 below shows the distribution of itemsets with no seats left available at the end of the enrollment period, by the number of terms present throughout the six terms under study (i.e. items in line Seats Left = 0 in Table 11). Results indicate again that a minority of itemsets concentrate the majority of demand over the six terms. While 5.11% of itemsets with no seats left were present for three or more years, they correspond to 46.18% of the demand for itemsets with no seats left.

To clarify, in Table 13 the sum of terms present times the count of distinct items equals the total number of itemsets present over six terms. That is, $\sum$ Terms Present * Count of Distinct Itemsets) = Total number of itemsets with not seats left over six terms. More precisely, in Table 11, from line for zero seats left: 24,898 + 27,902 + 25,574 + 23,466 + 28,084 + 25,070 = 154,994, and in table 13: 6*337 + 5*892 + 4*1,696 + 3*3,602 + 2*9,759 + 1*111,404 = 154,994.

Table 13 Distribution of course itemsets with no seats left available by number of terms present

Terms present	Count of Distinct Itemsets	% / total	Cumulative %	Demand	% / total	Cumulative %
6	337	0.26%	0.26%	35,865	6.10%	6.10%
5	892	0.70%	0.96%	87,756	14.93%	21.03%
4	1,696	1.33%	2.29%	65,266	11.10%	32.14%
3	3,602	2.82%	5.11%	82,542	14.04%	46.18%
2	9,759	7.64%	12.75%	98,223	16.71%	62.90%
1	111,404	87.25%	**100.00%**	218,074	37.10%	**100.00%**
Total	**127,690**	**100.00%**		**587,726**	**100.00%**	

Table 14 below shows the count of closed course itemsets with no seats left by enrollment and archive period. The distribution of itemsets shows that like in the case of table 2, which shows the equivalent information for all itemsets, the vast majority of itemsets with not seats left after the enrollment period has ended, have five or less students (89%). Furthermore, itemsets with one or two enrolled students make up for 70% of the itemsets with no seats left available versus 63.3% for the case of all itemsets (shown in Table 2).

Table 14 Count of closed course itemsets with no seats left by enrollment and archive period

	Count of closed course itmesets with no seats left available per archive period								
Enrolled per Itemset	**2091**	**2101**	**2111**	**2121**	**2131**	**2141**	**Total**	**% / Total**	
1	10,622	11,692	10,943	10,166	11,531	10,528	65,482	42.2%	
2	6,828	7,875	7,307	6,380	7,938	6,727	43,055	27.8%	↑
3	2,633	2,980	2,724	2,497	2,939	2,691	16,464	10.6%	88.9%
4	1,286	1,472	1,342	1,178	1,510	1,334	8,122	5.2%	↓
5	764	849	724	727	873	806	4,743	3.1%	
6	519	570	487	460	589	511	3,136	2.0%	
7	347	360	326	334	440	352	2,159	1.4%	
8	231	274	244	245	302	246	1,542	1.0%	
9	207	224	205	204	231	239	1,310	0.8%	
10	164	172	145	151	181	168	981	0.6%	
11	122	146	118	108	150	150	794	0.5%	
12	96	107	99	88	154	123	667	0.4%	↑
13	74	99	92	79	98	85	527	0.3%	11.1%
14	71	63	67	71	85	73	430	0.3%	↓
15	77	72	62	45	64	73	393	0.3%	
16	44	73	57	33	56	66	329	0.2%	
17	51	60	43	33	46	53	286	0.2%	
18	40	39	35	54	57	47	272	0.2%	
19	47	52	36	36	39	39	249	0.2%	
20	43	36	18	33	37	44	211	0.1%	
20+	632	687	500	544	764	715	3,842	2.5%	
Total	**24,898**	**27,902**	**25,574**	**23,466**	**28,084**	**25,070**	**154,994**	**100.0%**	

Table 15 shows the top 30 course itemsets by total enrollment over the six terms under analysis. Table 16 shows the top 30 course itemsets by total enrollment and with not seats left over the same terms. From the tables, it is also immediately apparent that courses are present across multiple itemsets and that there are patterns that are present over time at least in the itemsets with the highest enrollments.

Table 15 Top 30 closed course itemsets by total enrollment in the six fall terms under analysis

Line #	Closed Course Itemset	Enrollment by Archive Period					
		2091	2101	2111	2121	2131	2141
1	ARTSC_BIOSC_0050 ARTSC_BIOSC_0150	814	922	961	958	868	940
2	ARTSC_CHEM_0110 ARTSC_FP_0001	488	588	586	585	565	631
3	ARTSC_BIOSC_0150 ARTSC_CHEM_0110	526	530	584	594	556	590
4	ARTSC_CHEM_0310 ARTSC_CHEM_0330	551	571	576	538	502	505
5	ARTSC_BIOSC_0050 ARTSC_CHEM_0110	507	519	551	563	480	516
6	ARTSC_FP_0001 ARTSC_PSY_0010	527	580	503	482	431	497
7	ARTSC_BIOSC_0050 ARTSC_BIOSC_0150 ARTSC_CHEM_0110	480	492	534	541	457	493
8	ARTSC_BIOSC_0150 ARTSC_FP_0001	396	440	480	474	475	509
9	ARTSC_ENGCMP_0200 ARTSC_FP_0001	405	371	425	416	426	545
10	ARTSC_BIOSC_0050 ARTSC_FP_0001	361	419	441	447	405	437
11	ARTSC_MATH_0220 ARTSC_PHYS_0174	346	360	429	430	445	469
12	ARTSC_BIOSC_0050 ARTSC_BIOSC_0150 ARTSC_FP_0001	348	409	438	435	393	433
13	ENGR_ENGR_0011 ENGR_ENGR_0081	362	372	379	413	453	466
14	ARTSC_ENGCMP_0200 ARTSC_PSY_0010	344	336	435	446	354	398
15	ARTSC_PHYS_0174 ENGR_ENGR_0081	321	340	359	386	436	442
16	ARTSC_PHYS_0174 ENGR_ENGR_0011 ENGR_ENGR_0081	299	301	325	358	395	408
17	ARTSC_CHEM_0960 ENGR_ENGR_0081	296	316	329	308	346	371
18	ARTSC_ECON_0100 ARTSC_PSY_0010	370	363	380	349	280	215
19	ARTSC_CHEM_0110 ARTSC_PSY_0010	250	338	312	346	317	360
20	ARTSC_CHEM_0110 ARTSC_MATH_0220	218	289	316	335	329	358
21	ARTSC_CHEM_0960 ENGR_ENGR_0011 ENGR_ENGR_0081	269	285	301	295	327	335
22	ARTSC_MATH_0220 ENGR_ENGR_0081	280	276	294	282	326	301
23	ARTSC_MATH_0220 ENGR_ENGR_0011 ENGR_ENGR_0081	279	273	289	277	322	299
24	ARTSC_BIOSC_0150 ARTSC_CHEM_0110 ARTSC_FP_0001	270	279	308	307	278	288
25	ARTSC_MATH_0220 ARTSC_PHYS_0174 ENGR_ENGR_0081	274	268	291	279	316	297
26	ARTSC_MATH_0220 ARTSC_PHYS_0174 ENGR_ENGR_0011 ENGR_ENGR_0081	273	265	286	274	312	296
27	ARTSC_CHEM_0960 ARTSC_PHYS_0174 ENGR_ENGR_0081	240	251	280	263	299	318
28	ARTSC_CHEM_0110 ARTSC_ENGCMP_0200	190	184	270	268	280	428
29	ARTSC_BIOSC_0050 ARTSC_CHEM_0110 ARTSC_FP_0001	257	269	288	290	238	247
30	ARTSC_CHEM_0960 ARTSC_PHYS_0174 ENGR_ENGR_0011 ENGR_ENGR_0081	230	234	260	253	286	296

Table 16 Top 30 closed course itemsets with no seats left at the end of the enrollment period, by enrollment per term and present six terms

Line #	Closed Course Itemsets with no Seats Left at the End of Enrollment Period	Enrollment by Archive Period					
		2091	2101	2111	2121	2131	2141
1	ARTSC_CHEM_0310 ARTSC_CHEM_0330	551	571	576	538	502	505
2	ARTSC_CHEM_0110 ARTSC_PHYS_0174	75	118	161	195	200	198
3	ARTSC_CHEM_0330 ARTSC_PHYS_0110	68	65	71	90	77	70
4	ARTSC_CHEM_0110 ARTSC_NROSCI_0080	58	77	95	62	55	62
5	ARTSC_CHEM_0330 ARTSC_MATH_0240	80	96	39	45	48	46
6	ARTSC_BIOSC_0350 ARTSC_CHEM_0330	54	36	50	60	79	74
7	ARTSC_BIOSC_0050 ARTSC_CHEM_0120	51	50	58	48	49	50
8	ARTSC_BIOSC_0050 ARTSC_CHEM_0330	65	78	65	28	41	25
9	ARTSC_CHEM_0120 ARTSC_MATH_0220	32	31	33	40	56	58
10	ARTSC_BIOSC_0160 ARTSC_CHEM_0110	42	27	40	39	45	53
11	ARTSC_BIOSC_0150 ARTSC_CHEM_0120	29	47	36	38	47	46
12	ARTSC_CHEM_0330 ARTSC_PSY_0010	43	48	41	39	24	42
13	ARTSC_CHEM_0120 ARTSC_PHYS_0110	37	27	37	42	39	51
14	ARTSC_CHEM_0330 ARTSC_COMMRC_0520	40	27	48	29	40	31
15	ARTSC_CHEM_0330 ARTSC_STAT_0200	25	41	44	36	27	38
16	ARTSC_BIOSC_0370 ARTSC_CHEM_0330	24	28	23	36	41	39
17	ARTSC_MUSIC_0411 ARTSC_MUSIC_0412	31	30	29	25	22	26
18	ARTSC_CHEM_0330 ARTSC_PSY_0310	24	24	39	14	27	31
19	ARTSC_CHEM_0120 ARTSC_PSY_0010	26	19	24	23	28	36
20	ARTSC_CHEM_0330 ARTSC_NROSCI_1250	28	26	17	31	29	22
21	ARTSC_CHEM_0330 ARTSC_MATH_0220	27	24	27	22	22	25
22	ARTSC_CHEM_0310 ARTSC_CHEM_1720	21	29	25	34	19	16
23	ARTSC_CHEM_0120 ARTSC_MATH_0230	29	19	21	20	24	25
24	ARTSC_CHEM_0330 ARTSC_THEA_0830	29	11	18	24	26	21
25	ARTSC_CHEM_0120 ARTSC_PSY_0310	11	14	14	26	26	34
26	ARTSC_CHEM_0310 ARTSC_CHEM_0330 ARTSC_CHEM_1720	20	26	21	29	16	12
27	ARTSC_CHEM_0330 ARTSC_MATH_0230	24	21	14	13	21	30
28	ARTSC_CHEM_1720 ARTSC_PHYS_0110	14	13	24	23	19	25
29	ARTSC_CHEM_0120 ARTSC_ENGCMP_0200	21	10	11	15	23	37
30	ARTSC_ANTH_0536 ARTSC_CHEM_0110	14	27	15	21	22	14

As a summary of results so far, for the particular case of Pitt's enrollment data:

- Every term, there is a substantial percentage (between 50% and 64%) of closed course itemsets with not seats available at the end of the enrollment period. In the opposite view, every term there is a substantial percentage (between 36% and 50%) of closed course itemsets with seats available at the end of the enrollment period (Table 11).
- While a minority of course itemsets with not seats left available at the end of the enrollment period (5.1%) are present for three or more years, they command 46.2% of the demand for those itemsets (Table 13).
- 89% of itemsets with no seats left after the enrollment period has ended have five or less students enrolled, and 70% of them have one or two students enrolled (Table 14).
- Course offerings and schedules in one department or school have an impact on the enrollment options for students in other departments and/or schools.
- There might be course itemsets of interest at any level of enrollment and not only in the extremes.
- There are apparent patterns in the combinations of courses in itemsets that recur term over term.
- Even though, at this point we have discussed only one set of components of the proposed architecture, it enables a first pass on the identification of enrollment patterns of interest without the need to have detail administrative and/or pedagogical knowledge about the course offerings. This is illustrated in the following sub-section with the example of the itemset "ARTSC_CHEM_0310 ARTSC_CHEM_0330". The methodology and tools presented in sections ahead enable deeper levels of analysis.

4.3.4 Sample Case Analysis Using Results from MASAI

To illustrate the results so far with a specific example, we use the only course itemset that is present in Tables 15 and 16, which is "ARTSC_CHEM_0310 ARTSC_CHEM_0330". This course itemset has the overall fourth largest enrollment, and the largest enrollment in the set of itemsets with no seats left available at the end of the term. Additionally, as these two courses are

listed as co-requisites, it would be of interest to explore at a deeper level why is not possible for more students to take the two courses in the same term.

Table 17 below shows the balance of seats in each of the two courses in the itemset "ARTSC_CHEM_0310 ARTSC_CHEM_0330" over the six terms under study (i.e. Total of Enrollment Capacity minus Actual Enrollment in all sections of each one of the two courses per term). From the figures in Table 17, it is clear that the Organic Chemistry Laboratory (ARTSC_CHEM_0330) is severely oversubscribed every term and is the source of the enrollment bottleneck in the itemset. More precisely, for the six fall terms between 2091 and 2141 there were 13 sections of the Organic Chemistry Laboratory offered per term (not shown in the tables). All of them were oversubscribed every term. As the number of offered seats in Organic Chemistry 1 appears to have been increased to 1000 per term, the number of offered seats in the co-requisite Organic Chemistry Laboratory 1 has remained constant. Thus, increasing the deficit of seats in the latter.

Table 17 Total Seats Offered Minus Total Enrollment per Academic Term in Course Itemset "ARTSC_CHEM_0310 ARTSC_CHEM_0330"

ORGANIC CHEMISTRY 1 (ARTSC_CHEM_0310)	2091	2101	2111	2121	2131	2141
Enrollment Capaciy	710	710	900	1,000	1,000	1,000
Actual Enrollment	729	773	878	938	954	934
Balance of Seats	**(19)**	**(63)**	**22**	**62**	**46**	**66**

ORGANIC CHEMISTRY LABORATORY 1 (ARTSC_CHEM_0330)	2091	2101	2111	2121	2131	2141
Enrollment Capacity	480	480	480	480	480	480
Actual Enrollment	593	627	641	653	676	672
Balance of Seats	**(113)**	**(147)**	**(161)**	**(173)**	**(196)**	**(192)**

Table 18 below shows the complete weekly utilization schedule of the laboratory rooms where Organic Chemistry Laboratory 1 was offered in academic term 2141. Without having detailed domain and administrative knowledge of this offering, it would appear that the

specialized facilities that are required to offer this chemistry laboratory are used almost to their full capacity from Monday to Friday.

It is of interest to note that a regular listing of oversubscribed courses would have included the Organic Chemistry Laboratory. However, it can be argued that a simple list of oversubscribed courses does not make immediately apparent the impact on the combination of courses that a substantial number of students can take, derived from not being able to enroll in Organic Chemistry and its co-requisite laboratory in the same term, as it happens for every one of the six fall terms under analysis.

Table 18 Complete Weekly Utilization Schedule for laboratory rooms where ARTSC_CHEM_0330 is offered – Academic Term 2141

Facility ID	Start	Stop	MON	TUES	WED	THURS	FRI	SAT
CHVRN00135	8:00	8:50				X	X	
CHVRN00135	9:00	9:50	X	X	X	X	X	
CHVRN00135	10:00	10:50		X		X	X	
CHVRN00135	11:00	11:50	X	X	X	X	X	
CHVRN00135	12:00	12:50		X		X	X	
CHVRN00135	13:00	13:50	X	X	X	X	X	
CHVRN00135	14:00	14:50	X	X	X	X	X	
CHVRN00135	15:00	15:50	X	X	X	X	X	
CHVRN00135	16:00	16:50	X	X	X	X	X	
CHVRN00135	17:00	17:50						
CHVRN00135	18:00	18:50	X	X	X			
CHVRN00135	19:00	19:50	X	X	X	X		
CHVRN00135	20:00	20:50	X	X	X	X		
CHVRN00135	21:00	21:50	X	X	X			

Facility ID	Start	Stop	MON	TUES	WED	THURS	FRI	SAT
EBERL00206	8:00	8:50				X	X	
EBERL00206	9:00	9:50	X	X	X	X	X	
EBERL00206	10:00	10:50		X		X	X	
EBERL00206	11:00	11:50	X	X	X	X	X	
EBERL00206	12:00	12:50		X		X	X	
EBERL00206	13:00	13:50	X	X	X	X	X	
EBERL00206	14:00	14:50	X	X	X	X	X	
EBERL00206	15:00	15:50	X	X	X	X	X	
EBERL00206	16:00	16:50	X	X	X	X	X	
EBERL00206	17:00	17:50						
EBERL00206	18:00	18:50	X	X	X			
EBERL00206	19:00	19:50	X	X	X	X		
EBERL00206	20:00	20:50	X	X	X	X		
EBERL00206	21:00	21:50	X	X	X			

4.4 USING MASAI TO IDENTIFY COURSES THAT CANNOT BE TAKEN TOGETHER DUE TO SCHEDULE CONFLICTS

So far, we have focused on identifying course itemsets of interest based on combinations of courses that students take together in an academic term. In this section the focus is on courses that cannot be taken together in a term due to schedule conflicts, and that therefore do not appear in transaction data sets. This problem is known in Association Rule Analysis as the identification of negative association rules (Mani, 2012).

The segment of MASAI's procedure GET_MAX_BUNDLE that checks for schedule collisions in couples of courses sections is used to implement a PLSQL procedure that process any set of courses provided as input in the record format shown in Table 19 below. The output is a data set where every record is a couple of sections and a flag indicating if a schedule collision exists as illustrated in the sample shown in Table 20 (i.e. a collision flag =1 indicates that a schedule collision exists in the course tuple). The resulting data set is easily processed to identify the courses in the input data set that cannot be taken together (i.e. courses where all their sections have schedule collisions).

Table 19 Sample input record for identification of courses that cannot be taken together due to schedule conflicts

ARCHIVE_PERIOD	2141
SESSION_CODE	AT
ACAD_GROUP_CD	ARTSC
SUBJECT_CD	AFRCNA
CATALOG_NBR	0031
CLASS_NBR	10646
CLASS_SECTION	1070
DESCR	INTRODUCTION TO AFRCNA STUDIES
MON	N
TUES	Y
WED	N
THURS	Y
FRI	N
STARTT	12/30/1899 11:00:00 AM
STOPT	12/30/1899 12:15:00 PM

Table 20 Sample output of procedure that identifies courses with sections offered at conflicting schedules

Archive Period	Class Number 1	Class Section 1	Class Number 2	Class Section 2	Collision Flag	Course Tuple
2141	27270	1030	16166	1080	1	ARTSC_SOC_1119 SIS_INFSCI_0010
2141	28770	1200	16166	1080	0	ARTSC_SOC_1488 SIS_INFSCI_0010
2141	29019	1230	16166	1080	0	ARTSC_CS_0401 SIS_INFSCI_0010
2141	22428	1200	16166	1080	1	ARTSC_CS_0401 SIS_INFSCI_0010
2141	22447	1300	16166	1080	1	ARTSC_CS_0007 SIS_INFSCI_0010
2141	10173	1070	16166	1080	1	ARTSC_CS_0004 SIS_INFSCI_0010

The described methodology enables the identification of all couples of courses that cannot be taken together in a term due to schedule conflicts. In order to determine which of those couples of courses might be of interest, it is necessary to consider the structure of programs and course requirements at a particular institution. For instance, it would not be of interest to know that a basic mathematics course has conflicting schedule with an advanced physics course as nobody would, could or should take them at the same time.

A reasonable approach that can be taken in this case is to determine groups of courses of interest for analysis. For instance, the following three cases would be of interest at the level of schools or departments as they limit the enrollment options for students: *First*, identification of courses that are offered in multiple sections at the same schedule over several terms. *Second*, identification of courses of equivalent academic level that have no pre-requisite constraints and are offered at the same schedule over several terms. *Third*, in the set of all courses that meet general education requirements, it would be of interest to identify courses that cannot be taken together due to schedule conflicts. The following section discusses results obtained when processing sets of courses in the referred cases.

4.4.1 Results and Sample Cases of Courses that cannot be Taken Together in an Academic Term

Pitt course offerings throughout the six fall data terms under analysis, corresponding to the three cases discussed above where processed. The results are as follows.

First Case: Identification of courses that are offered in multiple sections at the same schedule

In the Pitt enrollments data set there are 56 courses that are offered for one or more terms with all sections of the same course at the same schedule (i.e. same days and times). Table 21 below shows the sub-set of 10 courses with multiple sections offered at same schedule per term for four or more of the last six fall terms. Offering multiple sections of the same course at the same time, at prime time slots, for several days of the week limits the enrollment options for students. Unless reasons exists for these schedules, it might be better to offer those sections at different schedules. This is corroborated with the examples presented in Section 6.

Table 21 Courses with multiple sections offered at same schedule per term for four or more of the last six fall terms

		Sections Offered per Term								
Course	**Course Titlle**	**2091**	**2101**	**2111**	**2121**	**2131**	**2141**	**Days**	**Start**	**Stop**
ARTSC_GER_0002	Elementary German 2	2	2	2	2	2	2	MTWHF	10:00	10:50
ARTSC_GER_0004	Intermediate German 2	2	2	2	2	2	2	M W F	12:00	12:50
ARTSC_JPNSE_0001	First Year Japanese 1	2	2	2	2	2	2	T H	11:00	11:50
ENGR_ENGR_1050	Product Realization (*)	5	5	5	5	5	5	H	17:45	20:10
SOCWK_SOCWRK_1024	Practicum Seminar and LAB 1	3	2	3	3	2	3	M	13:00	15:50
ARTSC_MATH_1530	Advanced Calculus 1	2	1	2	2	2	2	M W F	12:00	12:50
ARTSC_PHYS_0111	Introduction to Physics 2	1	2	2	2	2	2	M W F	13:00	13:50
EDUC_PSYED_1028	Psychology in Education	2	2	2	2	1	1	T	13:00	15:50
EDUC_PSYED_1036	Developmental Practicum Seminar 1	2	2	2	2	1	1	T	13:00	15:50
ENGR_ENGR_0012	DEVELOPMNTL MEANG CULTL DISTN	2	2	2	2	1	3 (**)	M	13:00	15:50

* By Design: Students at different programs in Engineering enroll in different sections of the same course

** Two of the three sections offered in the same schedule

Second Case: Identification of courses of equivalent academic level that have no pre-requisites constraints and are offered at the same schedule over several terms

Table 22 below shows a sample of results including two Information Science courses with no excluding pre-requisites that appear as scheduled at the same time over the six fall terms under consideration. Thus, preventing the ability of students to enroll in both of them in the same term.

The data set under study does not include pre-requisites and co-requisites information. Thus, further processing and analyses of results on these cases was not attempted.

Table 22 School of Information Science courses that cannot be taken together due to schedule conflicts

Archive Period	Course	Class Number	Course Title	Days	Start	Stop
2091	SIS_INFSCI_1024	27936	INFORMATION SYSTEMS ANALYSIS	T	18:00	20:30
2091	SIS_INFSCI_1068	27978	GEOGRAPHIC INFORMATION SYSTEMS	T	18:00	20:50
2101	SIS_INFSCI_1024	25940	INFORMATION SYSTEMS ANALYSIS	T	18:00	20:30
2101	SIS_INFSCI_1068	25978	GEOGRAPHIC INFORMATION SYSTEMS	T	18:00	20:50
2111	SIS_INFSCI_1024	24888	INFORMATION SYSTEMS ANALYSIS	T	18:00	20:30
2111	SIS_INFSCI_1068	24918	GEOGRAPHIC INFORMATION SYSTEMS	T	18:00	20:50
2121	SIS_INFSCI_1024	16945	INFORMATION SYSTEMS ANALYSIS	T	18:00	20:30
2121	SIS_INFSCI_1068	16959	GEOGRAPHIC INFORMATION SYSTEMS	T	18:00	20:50
2131	SIS_INFSCI_1024	16672	ANALYSIS OF INFORMTN SYSTEMS	T	18:00	20:30
2131	SIS_INFSCI_1068	16683	GEOGRAPHIC INFORMATION SYSTEMS	T	18:00	20:50
2141	SIS_INFSCI_1024	16190	ANALYSIS OF INFORMTN SYSTEMS	T	18:00	20:30
2141	SIS_INFSCI_1068	16199	GEOGRAPHIC INFORMATION SYSTEMS	T	18:00	20:50

Third Case: In the set of all courses that meet general education requirements, it would be of interest to identify courses that cannot be taken together due to schedule conflicts.

Courses that satisfy general education requirements (GENED[1]) are offered across departments and schools and most students complete them within the first two years of their programs. Table 23 below shows figures on GENED course offerings at Pitt over the six fall terms under analysis, including the percentage of couples of these courses that cannot be taken together per term due to solely to schedule conflicts. In average, 4% of the total combinations of couples of courses per term have schedule conflicts. An analysis including pre-requisites was not attempted due to the lack of required data available in the authorized data set. However, most of these courses are low level and do not have pre-requisites.

Table 23 Count of general education courses that cannot be taken together due to schedule conflicts

Archive Period →	2091	2101	2111	2121	2131	2141
GENED Courses Offered	376	422	392	395	423	469
Possible Combinations of Two Courses	70,500	88,831	76,636	77,815	89,253	109,746
Combinations of Two Courses Not Possible due to Schedule Conflicts	3,328	3,586	2,969	2,844	3,683	4,090
Percentage of Combinations Not Possible over Possible Combinations	4.72%	4.04%	3.87%	3.65%	4.13%	3.73%

Table 24 below shows the results of the longitudinal analysis on GENED couples of courses offered in conflicting schedules. Out of 15,681 couples of courses that have schedule conflicts in any of the six fall terms under analysis, 83 have schedule conflicts in all six terms corresponding to 0.535 of the total. Furthermore, 6.88% of these course couples have schedule conflicts over three or more of the six fall terms; 18.97% have schedule conflicts over two or more of the six terms.

[1] Dietrich School of Arts and Sciences General Education Requirements: http://www.as.pitt.edu/fac/teaching/general-requirements

Table 24 Longitudinal analysis of general education courses that cannot be taken together

Number of Terms	Couples of Courses	% / Total	Cummulative %
6	83	0.53%	0.53%
5	136	0.87%	1.40%
4	245	1.56%	2.96%
3	615	3.92%	6.88%
2	1,895	12.08%	18.97%
1	12,707	81.03%	**100.00%**
Total	**15,681**	**100.00%**	

In the particular case of Pitt, the percentage of GENED courses that cannot be taken together appears to be low. However, given that students complete most of their GENED courses requirements over the first two years of their programs, these results mean that for most students those combinations of courses are never possible in practice, as by the third year of their programs they have moved on to complete other requirements. Appendix B includes a list of couples of GENED courses that are not possible to enroll together for any of the six terms under analysis.

4.5 STAGE I ARCHITECTURAL DIAGRAM

Figure 7 below shows Stage I of the proposed architecture. The components discussed above would be compatible with the basic components that are used in current timetabling architectures. There is a data preparation process from the original data source to feed the discussed relational schema. Then there is the identification of closed itemsets that reads data from the relational schema and feeds the results back into a table in that schema. The proposed algorithm MASAI finds the maximum capacity of each closed itemset by using data fields at the level of individual sections in each itemset as discussed above. It could also receive input from an interactive optimization algorithm to test for maximum capacities in course itemsets when the algorithm needs to check for capacity constraints thresholds.

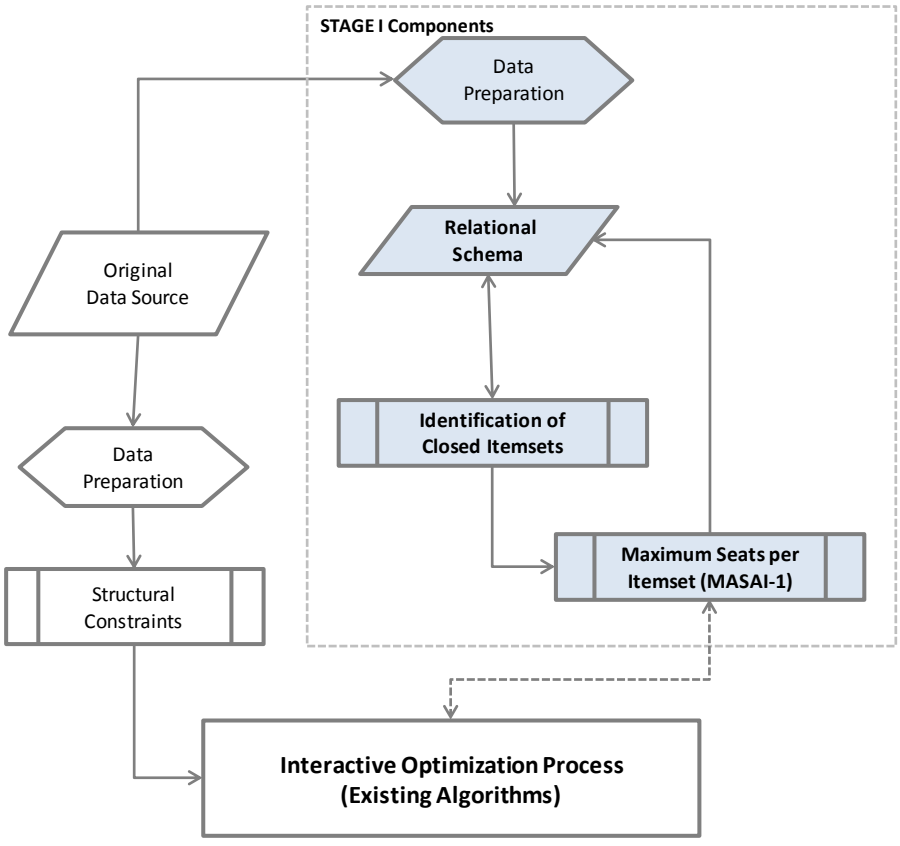


Figure 7 Stage I architectural diagram

5.0 IDENTIFYING OVERLAPPING AND HIERARCHICAL COMMUNITIES OF COURSES USING MULTI-MODE GRAPH ANALYSIS

The Stage I architectural components discussed in the previous sections solve the problem of computing the maximum seats possible in all combinations of courses that are closed itemsets. That is, those components identify course itemsets that are schedule bottlenecks that prevent higher enrollments, and limit the options for students to take the combination of classes that they want or need. However, there is a core aspect of the timetabling problem that is not captured by the association rules or the itemset capacity analysis. That is, most courses are present in multiple itemsets. The sample of closed itemsets shown in Table 5 illustrates this, with the most salient examples in the sample being BIOSC 0050 and 0150, which are present in most of the closed itemset shown in the table. It is then not enough to know which course itemsets are generating schedule bottlenecks. It is also necessary to understand the relationship between those course itemsets to determine which are the most critical and which groups of courses appear to have the potential to benefit from changes in scheduling practices and/or collaborative scheduling. Furthermore, for multiple purposes associated with effective and efficient planning in higher education, it is important to help administrators identify non-trivial enrollment patterns.

In order to facilitate the identification of non-trivial enrollment patterns and provide architectural components to support the associated analyses, the problem is modeled as a multi-mode weighted graph as discussed ahead. Figure 8 below provides an initial illustration of the graph core mode. In the example shown in the figure, courses are a class of vertex in the graph, and the edge attributes include the number of students taking the courses at each end of an edge.

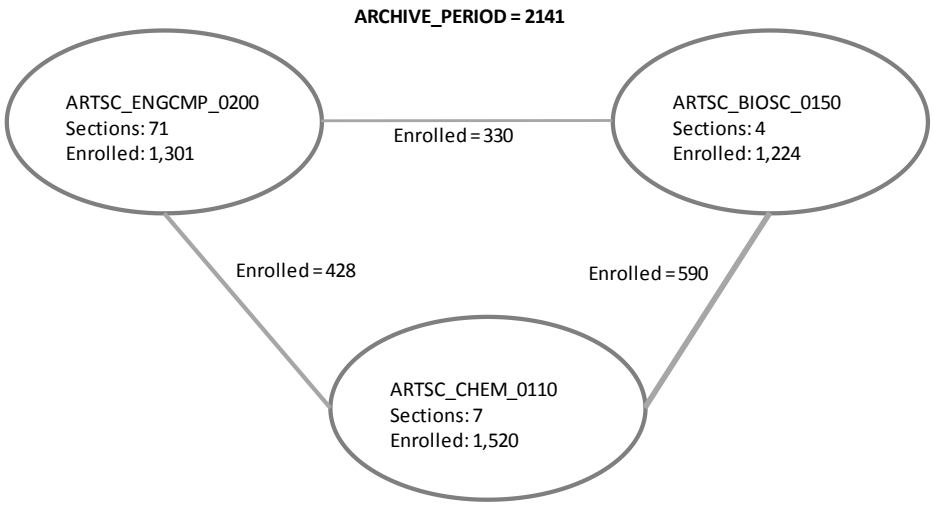


Figure 8: Graph showing three courses as nodes, and edges as enrollments between each couple of courses

5.1 MODELING THE TIMETABLING PROBLEM AS A MULTI-MODE GRAPH

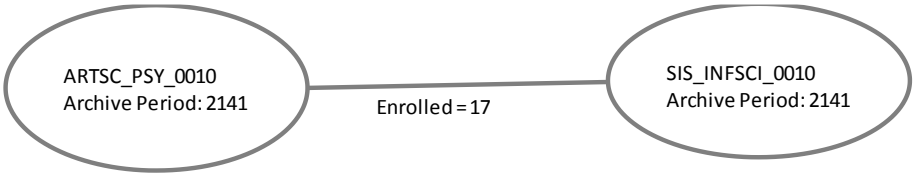
Modeling the timetabling problem as a multi-mode graph has three important advantages. *First*, it enables the direct use of mature developments and software on graphs/network analysis. Of specific interest for the purpose here is the identification of communities in the resulting graph. Those communities include the courses that might benefit from changes in scheduling practices and/or collaborative scheduling. They can also provide valuable information to identify non-trivial existing and/or emerging enrollment patterns. *Second*, communities of courses can be directly associated with scheduling authorities, who then could work collaboratively in an environment designed for that purpose. This would enable them to identify and, if opportune and viable, modify scheduling practices that have a negative impact on the quality of schedules. This would also enable administrators to determine better informed hard and soft constraints to be passed to optimization algorithms. *Third*, the graph approach enables visualizations that facilitate the understanding of complex relations in the enrollments of multiple courses across departments and schools.

The use of the graph approach to model the problem provides more than a conceptual artifact. Zaki and colleagues, showed that there are deep theoretical links between frequent

itemsets and bipartite graph cliques. They leveraged those links to develop algorithms (*Clique* and *MaxClique*) for fast discovery of association rules that are based on the identification of maximal hypergraph cliques (Zaki, Parthasarathy, Ogihara, & Li, 1997). More precisely, the input for association mining consists of sets of items grouped in transactions where each item can be present in multiple transactions. This input can also be modeled as a bipartite graph with two distinct vertex sets, namely transactions and items. *The problem of enumerating all frequent itemsets corresponds to the task of enumerating all constrained bipartite cliques.* A clique is a complete graph, where every pair of vertices are connected by an edge. Finding the maximum bipartite clique (i.e. the largest bipartite clique in a bipartite graph) is an NP complete problem (Zaki & Ogihara, 1998).

In order to support the community identification process, and later on the analysis of results and a graphic user interface, we propose a multi-mode weighted, labelled and attributed graph. We also leverage the link between frequent itemsets and graph cliques in the opposite direction than Zaki and colleagues did. That is, we start with the identification of closed itemsets, which then become cliques in the multi-mode graph. The graph includes different groups of nodes (or vertex) to represent itemsets, courses, and sections, each one with proper identifiers as discussed ahead. The graph also includes different kinds of non-directed and directed edges to represent the relationships among the different kinds of nodes.

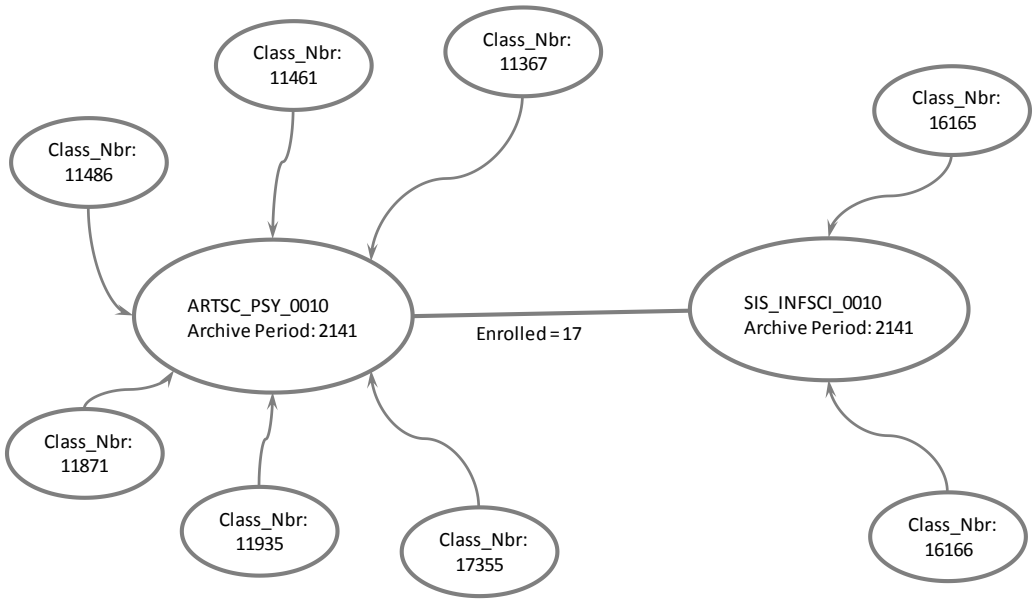
There is a class of nodes representing courses, and two of those nodes are linked with a non-directed edge when there are students enrolled in both courses (i.e. they are adjacent nodes). The weight of the edge corresponds to the number of students enrolled in the courses at either end of the edge. This group or sub-graph $G_c = (C, E)$ consists of the set of nodes $C = \{c_1, c_2, \ldots, c_n\}$ where c_i is a course, and the set of edges E, which are unordered pairs of elements of C. That is, $E \subseteq C \times C$. Each edge has a weight $w(c_i, c_j)$ that represents the number of students enrolled in both courses c_i and c_j. *C* includes all courses that appear in closed itemsets identified during the Association Rule Analysis. In fact, to generate part of the data for the course sub-graph, the Association Rule Analysis is run with support and confidence equal to zero and a maximum itemset size of two. Figure 9 below illustrates the relationship between nodes representing enrollments in adjacent courses using a case from the Pitt enrollment data set in archive period 2141 (September to December 2013).



Archive Period	Academic Group Description	Subject Code	Catalog Number	Course Title	Enrollment Total
2141	Dietrich Sch Arts and Sciences	PSY	0010	Introduction to Psychology	1,748
2141	School of Information Science	INFSCI	0010	Introduction to Information Systems and Sociey	99

Figure 9 Enrollment relationship between course nodes in graph

One or more sections of a course are offered every term, and each section is associated with only one course. To model this, the graph includes nodes to represent sections. The directed edge from a section to its course indicates a relationship where the section belongs to the course. Sub-graph $G_{CS} = (C, S, CS)$ is a bipartite graph consisting of the set of nodes $C = \{c_1, c_2, \ldots, c_n\}$ where c_i is a course; a set of nodes $S = \{s_1, s_2, \ldots, s_m\}$ where s_i is a section of a course offered in an academic term; and a set of edges CS, which are ordered pairs of elements of C and S indicating that section s_i belongs to course c_j. Figure 10 below illustrates with a specific example the relationship between nodes representing courses and sections.



Archive Period	Academic Group Description	Subject Code	Catalog Number	Course Title	Class Number	Enrollment Total
2141	Dietrich Sch Arts and Sciences	PSY	0010	Introduction to Psychology	11367	374
2141	Dietrich Sch Arts and Sciences	PSY	0010	Introduction to Psychology	11461	398
2141	Dietrich Sch Arts and Sciences	PSY	0010	Introduction to Psychology	11486	238
2141	Dietrich Sch Arts and Sciences	PSY	0010	Introduction to Psychology	11871	87
2141	Dietrich Sch Arts and Sciences	PSY	0010	Introduction to Psychology	11935	279
2141	Dietrich Sch Arts and Sciences	PSY	0010	Introduction to Psychology	17355	372
2141	School of Information Science	INFSCI	0010	Introduction to Information Systems and Sociey	16165	57
2141	School of Information Science	INFSCI	0010	Introduction to Information Systems and Sociey	16166	42

Figure 10 Relationship between nodes representing courses and sections (Section belongs to Course)

Each course can be present in one or more itemsets, and an itemset includes two or more courses. Thus, the graph includes nodes to represent course itemsets per term. An edge from a course node to an itemset node indicates that the course belongs to or is a member of the itemset. Sub-graph G_{IC} = (I, C, IC) is a bipartite graph consisting of the set of nodes $C = \{c_1, c_2, \ldots, c_n\}$ where c_i is a course; a set of itemsets $I = \{i_1, i_2, \ldots, i_l\}$ where i_i is a closed itemset of courses as previously discussed; and a set of edges IC which are ordered pairs of elements of C and I indicating that course c_i belongs to itemset i_j. Two courses c_i, c_k belong to itemset i_j if and only if they have students mutually enrolled. Figure 11 below illustrates the relationship between nodes representing courses and itemsets with a specific example.

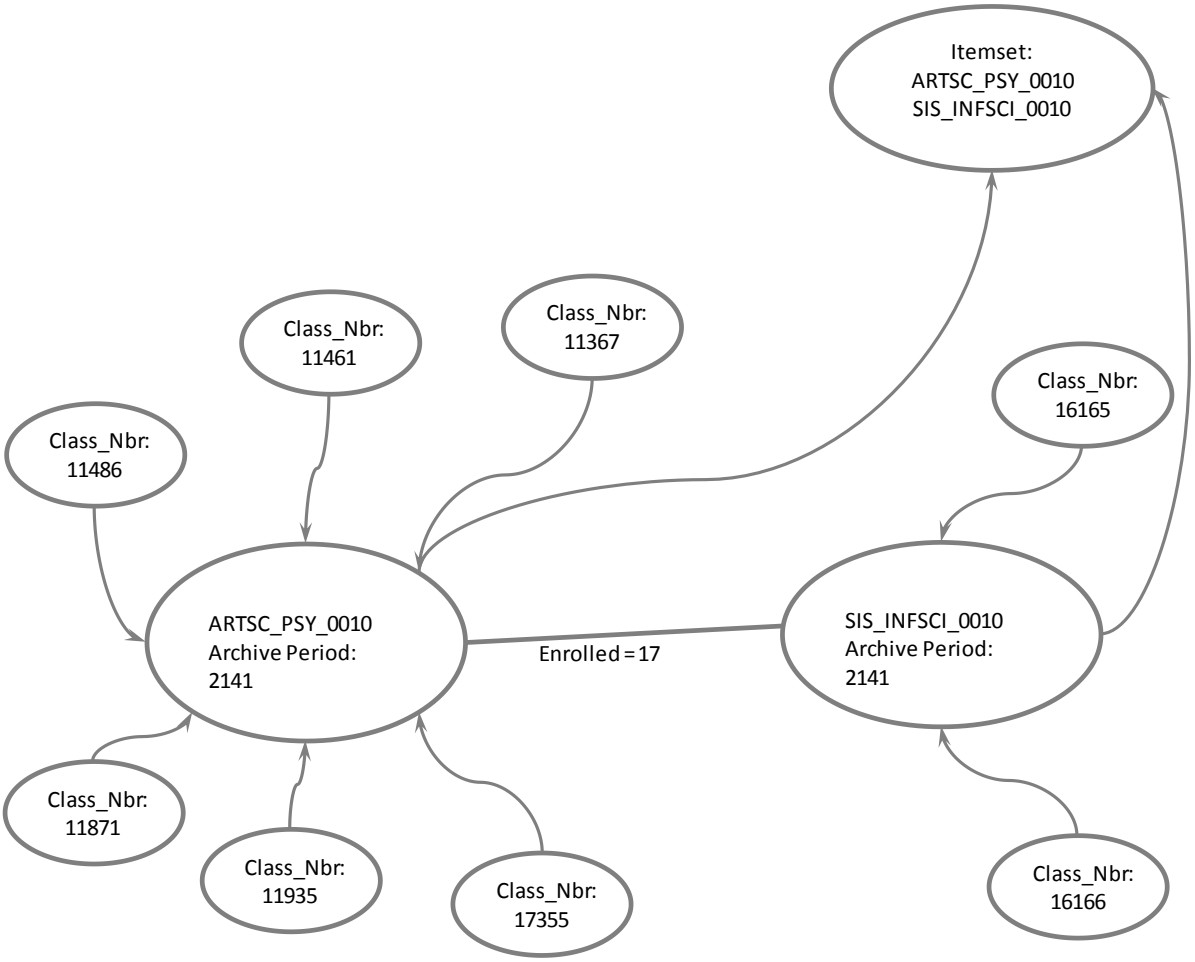


Figure 11 Relationship between courses and itemsets (Course belongs to itemset)

As the itemset includes courses that have students who are mutually enrolled in the courses in the itemset, the itemset is a clique. Then by definition, *a closed itemset is a clique as all the courses in the itemset are connected to each other.* More precisely, in the courses graph $G_c = (C, E)$, which consists of the set of nodes $C = \{c_1, c_2, \ldots, c_n\}$ where c_i is a course; and the set of edges E, which are unordered pairs of elements of C (i.e. $E \subseteq C \times C$); *a clique is a subset of nodes of G_c, such that every two nodes in the subset are adjacent.* Thus, when we identified all the closed itemsets using Association Rule Analysis, we also identified all the course cliques in *Gc.*

The following sections discuss a methodology to link selected course itemsets (cliques) that have courses in common to produce a clique graph, and then perform the identification of overlapping and hierarchical communities in that graph. As an initial illustration, Figure 12 below shows the links (or edges) between six out of 28 itemsets that contain the courses ARTSC_PSY_0010 and SIS_INFSCI_0010 in archive period 2141. The "MULTI-PERCOLATE" edge between course itemsets is discussed in detail in the next section. Figure 12 and subsequent were produced taking screenshots from a prototype implementation using the graph database system NEO4J discussed in section 5.5 (Miller, 2013; Webber, 2012).

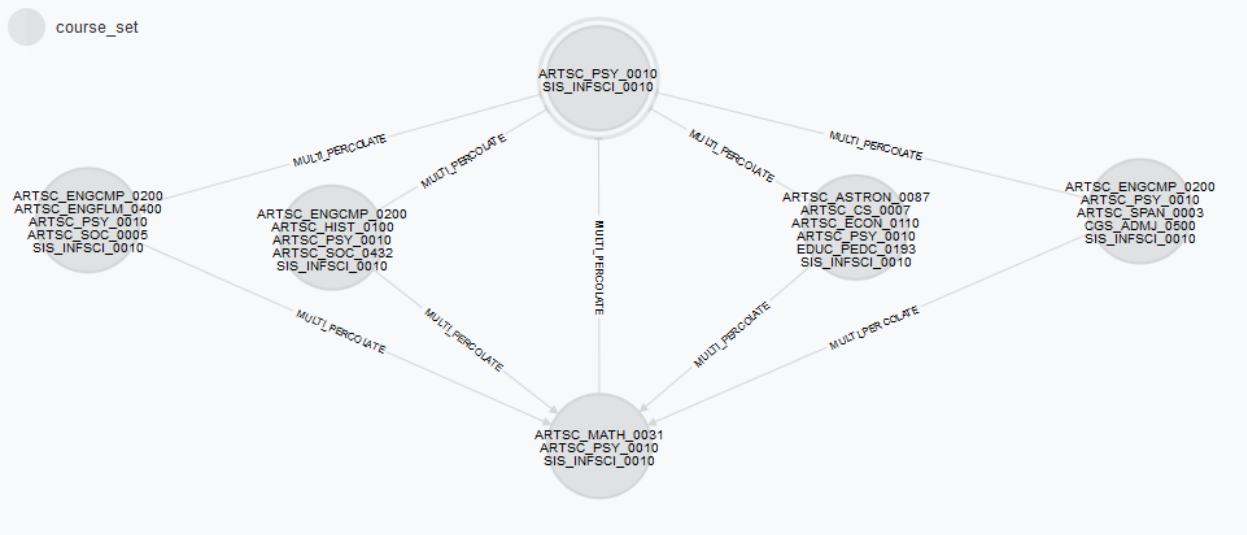


Figure 12 Selected itemsets where courses ARTSC_PSY_0010 and SIS_INFSCI_0010 are present in archive period 2141

5.2 OVERVIEW OF THE COMMUNITY IDENTIFICATION PROBLEM

Although there is not agreed upon definition of graph community or cluster, they are generally understood as group of nodes that have denser relations with each other than with the rest of the nodes in the graph. In other words, *"a community consists of a group of nodes that are relatively densely connected to each other but sparsely connected to other dense groups in the network"*

(Porter, Onnela, & Mucha, 2009). Figure 13 below provides an initial illustration of the concept by showing a sub-graph of courses for archive period 2141 that consists of two overlapping communities. Course ARTSC_HIST_0010 at the center overlaps the two communities at the left and right respectively (i.e. it has ENROLLED_TUPLE edges with four nodes in each community). Courses SIS_INFSCI_1014 and ARTSC_NROSCI_0081 overlap the two communities with two edges in both.

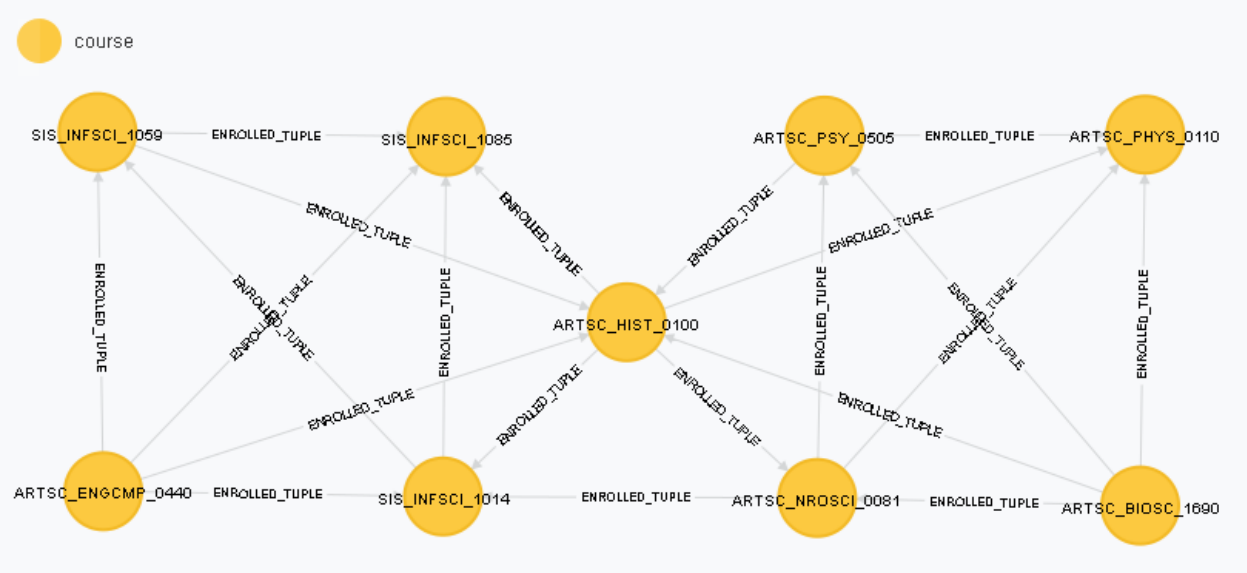


Figure 13 Sample sub-graph showing two overlapping communities of courses

In addition to overlapping communities, in the course graph we can expect to have nested communities. That is small communities form larger ones, which in turn might group together to form even larger communities that can have a hierarchical characteristic (Lancichinetti et al., 2009). Thus, any approach used to identify communities in the problem at hand needs to be able to handle those cases. Figure 14 below illustrates the idea with a sub-graph of course itemsets including the course ARTSC_HIST_0100.

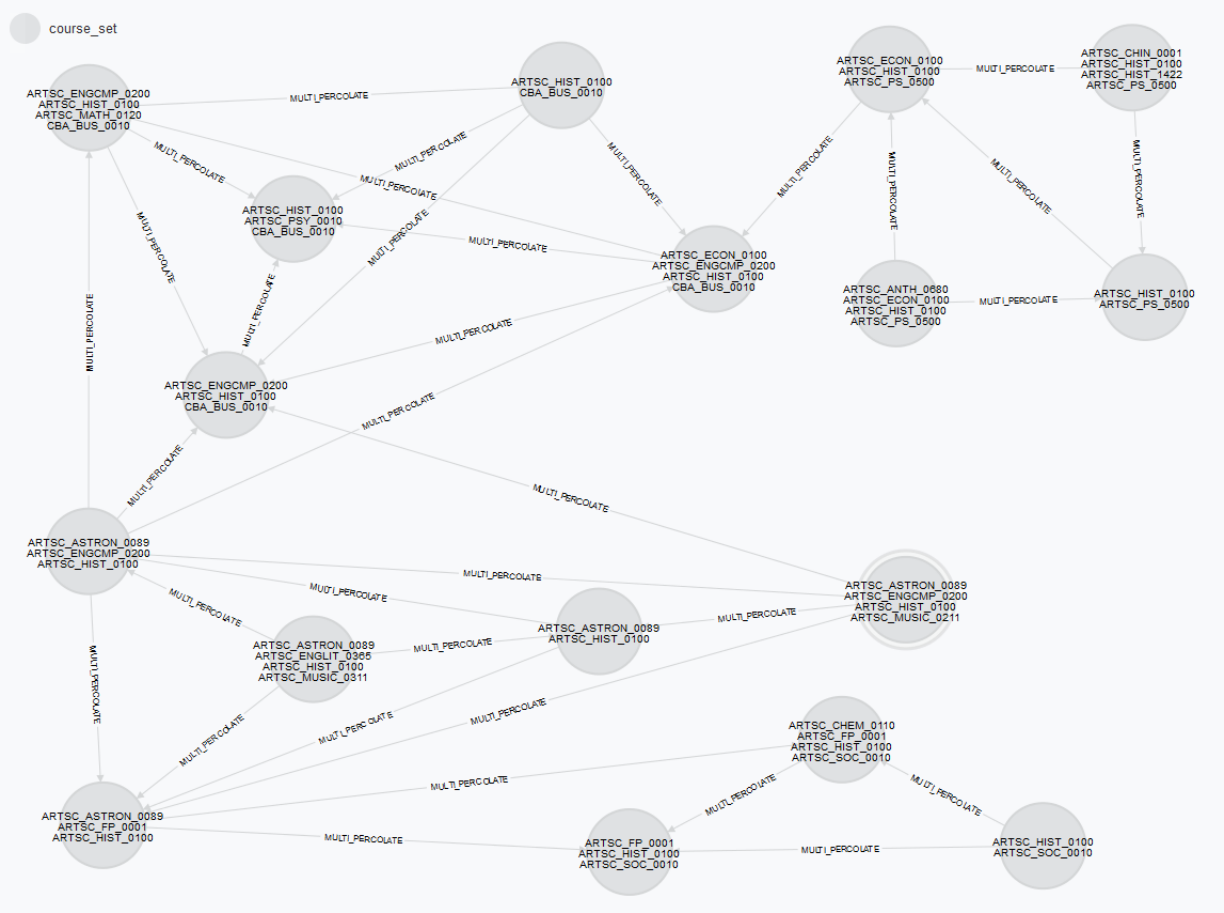


Figure 14 Sample sub-graph showing overlapping and hierarchical communities of course itemsets including course ARTSC_HIST_0010

The challenge is that community identification is a very hard problem that has not yet been satisfactorily solved despite the voluminous literature on the topic. Furthermore, the identification of overlapping hierarchies of communities is even harder and methodologies to that goal are still evolving. There are works that provide a good coverage on the state of the art in community detection in networks (T. Evans & Lambiotte, 2009; Fortunato, 2010; Porter et al., 2009; Xie, Kelley, & Szymanski, 2013). There are advances on link partitioning methods (Ahn, Bagrow, & Lehmann, 2010; T. Evans & Lambiotte, 2009; Tao, Wu, Shi, Cao, & Yu, 2014). There are developments on optimization methods based on statistical objectives (Havemann, Gläser, & Heinz, 2015; Lancichinetti et al., 2009; Lancichinetti, Radicchi, Ramasco, & Fortunato, 2011; Lee, Reid, McDaid, & Hurley, 2010). There are emerging methods that combine node partitioning, link partitioning and statistical objectives (He, Liu, Jin, & Zhang,

2015; Jin, Gabrys, & Dang, 2015). Other emerging methods assess the time evolution of community structure in networks considering each snapshot as an independent community detection problem or taking into account the whole evolution of the network (Granell, Darst, Arenas, Fortunato, & Gómez, 2015). There are also evolving approaches for ego-centered community identification in complex networks (Danisch, Guillaume, & Le Grand, 2014; Kanawati, 2015), greedy clique expansion for the identification of highly overlapping communities (Paul et al., 2015), and merging maximal cliques into a dendogram (Zhang & Wang, 2015)

In the case of the timetabling problem at hand, the need to identify overlapping and hierarchical communities precludes the use of community analysis algorithms that partition the set of vertices assigning each one to only one community (e.g. modularity maximization methods). A popular community identification method that identifies overlapping communities is the Clique Percolation Method (CPM). However, standard CPM cannot identify hierarchical communities and can reportedly take unacceptable long time to produce results even with moderately sized graphs. Thus, a Generalized Percolation Method (GCPM) is proposed to solve those challenges. As suggested by the literature on the topic, GCPM has potential application in numerous disciplines. In order to keep this document focused on the topic of timetabling in higher education the discussion on the use of GCPM in other disciplines is left for future work.

Before delving into the descriptions of CPM and GCPM, we provide a definition of clique graph by Harary (1969): *"A graph G is a clique graph if and only if it contains a family F of complete subgraphs, whose union is G, such that whenever every pair of such complete graphs in some subfamily F' have a nonempty intersection, the intersection of all the members of F' is not empty"* (Harary, 1969). Following from that definition, we are interested in identifying clique sub-graphs that include cliques of interest based on certain characteristics, as discussed ahead, as opposed to the overlap of all cliques in the complete graph.

Following is a discussion of both the Clique Percolation Method, and the proposed Generalized Clique Percolation Method (GCPM).

5.3 CLIQUE PERCOLATION METHOD CPM (A CURRENT COMMUNITY IDENTIFICATION METHODOLOGY)

The Clique Percolation Method (CPM) starts by constructing a clique graph (hypergraph) of *k*-cliques where *k* refers to the size of the cliques, and all cliques have the same size. Pairs of *k*-cliques that share *k*-1 nodes are said to percolate into each other. Then, to identify the communities in the graph it is necessary to identify cliques that percolate into each other. This can be done using a modularity based algorithm on the clique graph, or in a more straightforward way by thresholding it. The latter consists of dropping edges where the overlap between the cliques is less than *k*-1 (e.g. in a 3-clique graph, links between cliques that share less than two edges are dropped) or in case of weighted graphs by filtering out edges that have a weight below a pre-defined threshold. Additionally, the strength of the overlap can be assessed by the number of edges common to both k-cliques (Derényi et al., 2005). Evans notes that a consideration that is needed in this case is that nodes that appear in large number of cliques dominate the clique graph as a node that is present in *m* cliques contributes *m*(m-1)/2* edges to the clique graph. To control for that effect, he proposes the use of a *1/m* weight when computing the overlap strength of the two cliques (T. S. Evans, 2010)

While CPM partially identifies overlapping communities, it cannot easily identify hierarchical relationships in the communities. That limitation is originated in the restriction to set a fixed *k* size for the cliques while hierarchical relationships in communities involve cliques of different sizes (T. S. Evans, 2010; Porter et al., 2009). Even if multiple runs of CPM are performed using a fixed *k* for each run, the relationship between cliques of different sizes is still lost. Additionally, Although CPM is conceptually easy to understand, it turns out to be computationally hard, with the main challenges being finding the cliques, creating the clique graph and identifying the cliques that percolate into each other (Fu et al., 2014; Kumpula, Kivelä, Kaski, & Saramäki, 2008; Reid, McDaid, & Hurley, 2012):

- *Finding the cliques:* Even though finding the cliques in a graph is considered in theory an NP complete problem, in practice there are algorithms that solve the problem with the Bron-Kerbosch algorithm perhaps being the most popular (Bron & Kerbosch, 1973). Within the CPM problem the step of finding the cliques is the less demanding.

- *Creating the clique graph:* Densely connected graphs result in clique graphs that can have a number of edges several orders of magnitude larger than the original graph. This makes the computation expensive or even prohibitive as the number of connected components is quadratic in the number of maximal cliques in the graph. Reid and colleagues present a detailed discussion on the topic derived from the original specification from Everett and Borgatti ((Everett & Borgatti, 1998; Reid et al., 2012).

- *Identifying the cliques that percolate into each other:* This is the most challenging step in current clique percolation methods: "Because a *k*-community is defined as the set of nodes which all can be reached by a series of overlapping *k*-cliques, the crucial issue here is the efficient detection of overlap between *k*-cliques" (Kumpula et al., 2008). "Clique finding is a small part of the computational time in our algorithms. Instead computation is dominated by comparing cliques against each other" (Reid et al., 2012).

Figure 15 below illustrates the CPM concept with a 3-Clique and a 4-Clique Community. It is not possible for CPM to identify the relationships between the 3-Clique and the 4-Clique in the figure as by definition and methodology it only percolates cliques of equal *k* size with a *k*-1 overlap. Figure 16 shows the same set of cliques shown in Figure 15 with the relationships between cliques of different or equal size identified using the Generalized Clique Percolation Method GCPM discussed in the following section.

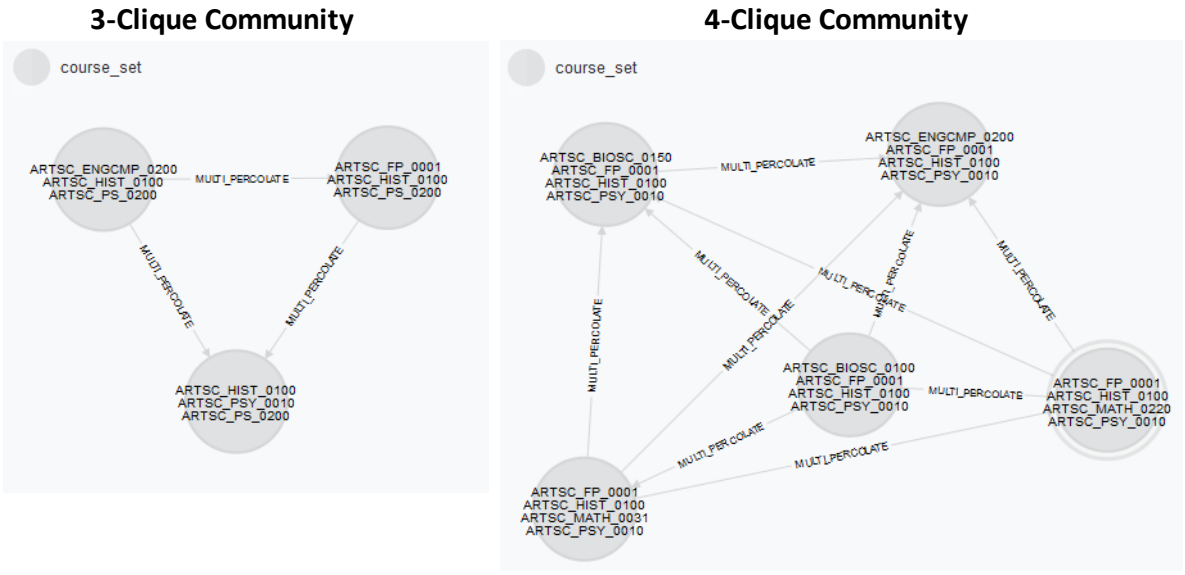


Figure 15 Sample of 3-clique and 4-clique communities identified with CPM. By definition and methodology, CPM cannot identify links between cliques of different sizes

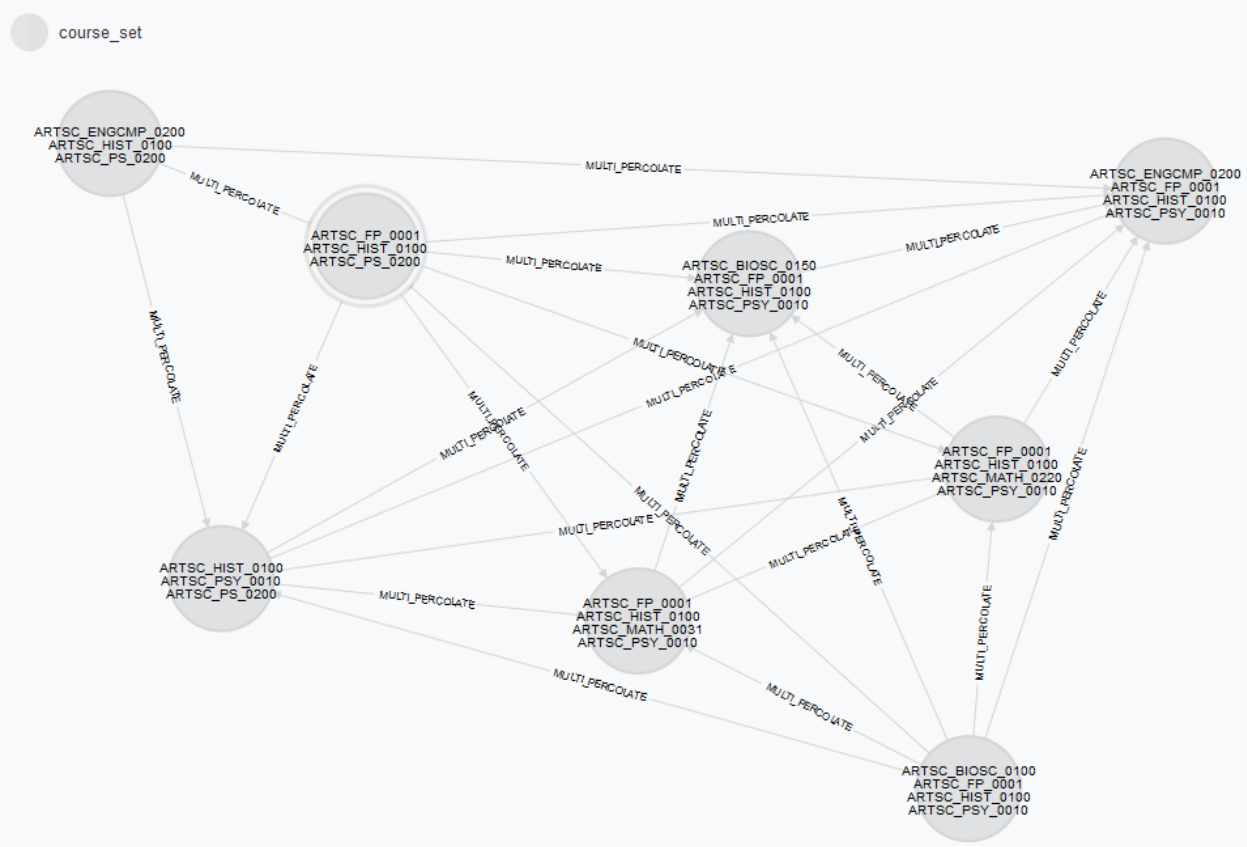


Figure 16 Cliques shown in Figure 15 with all edges between cliques of different or equal size identified using GCPM

The impact of the resulting loss of coverage when using CPM is illustrated in Figure 17, which is a copy of Figure 14 with communities formed by 3-Clique and 4-Clique enclosed in elliptical shapes. Even in the small sub-graph of cliques shown in Figure 17, there is a substantial loss of coverage as CPM identifies only the basic community structures formed by cliques of the same size. CPM misses all overlapping and hierarchical communities formed when the equal size clique restriction is relaxed.

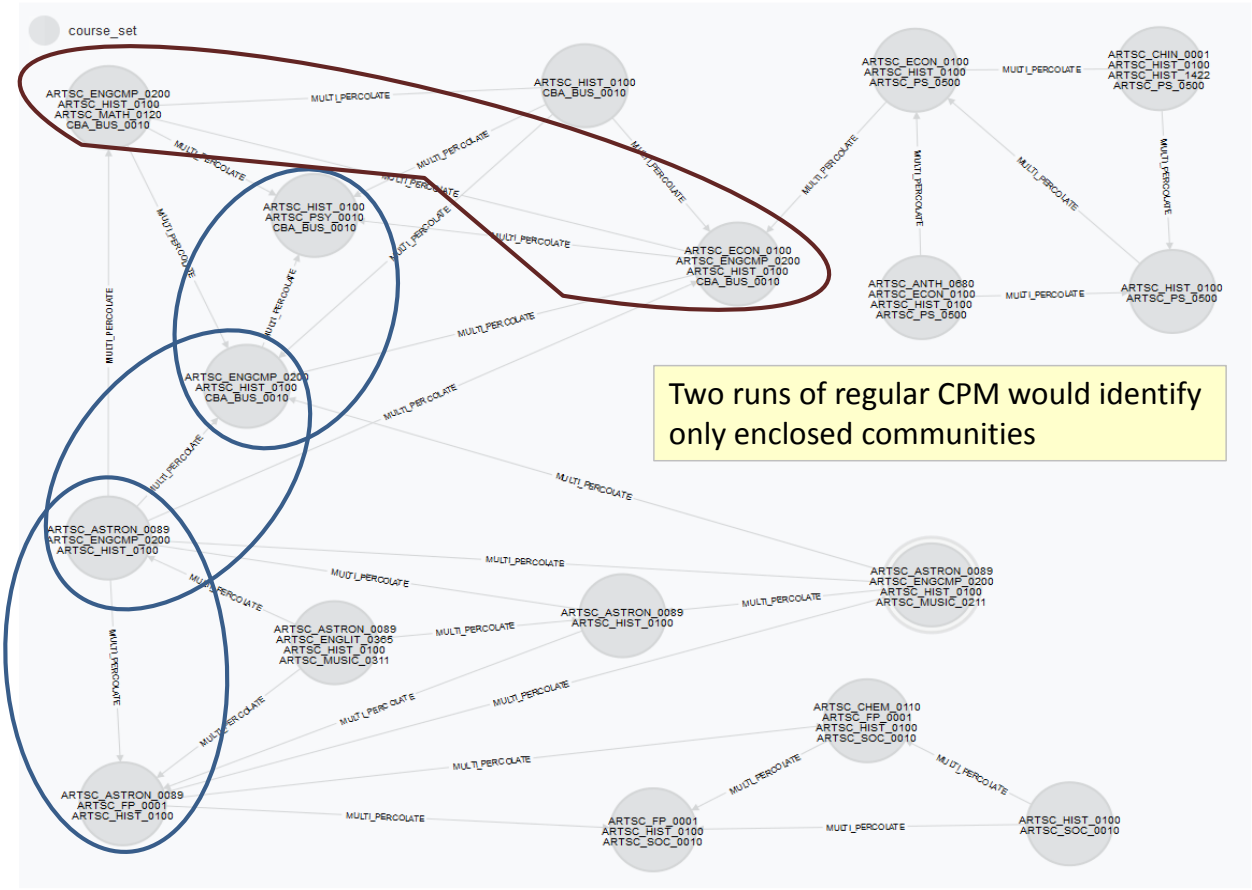


Figure 17 Sample sub-graph showing overlapping and hierarchical communities of course itemsets including course ARTSC_HIST_0010. 3-Clique and 4-Clique identified with CPM are enclosed in elliptical shapes. CPM does not identify all other overlapping and hierarchical structures between cliques of different sizes shown in the graph

5.4 GENERALIZED CLIQUE PERCOLATION METHOD GCPM (PROPOSED METHODOLOGY)

This section proposes a Generalized Clique Percolation Method (GCPM) to identify communities in graphs including overlapping and hierarchical communities. GCPM handles *k*-cliques of varied order as opposed to the standard approach in CPM of using *k*-cliques of fixed order.

We start with the course itemset nodes as defined in section 4.0. To recap, we have a set of itemsets $I = \{i_1, i_2, ..., i_l\}$ where i_i is a closed itemset of courses as identified with the Association Rule Analysis discussed in Section 4.1. All courses in an itemset i_i have students who are mutually enrolled in the courses in the itemset. Thus, by definition, *a closed itemset is a clique as all the courses in the itemset are connected to each other*. Therefore, when we identified all the closed itemsets and defined the relationship between courses and itemsets in the multi-mode graph (course belongs to itemset --BELONGS_TO_SET--), we also solved the first challenge of CPM, which is the identification of the cliques in the graph. Figure 18 illustrates the concept with an example of a closed itemset (clique) that has five courses.

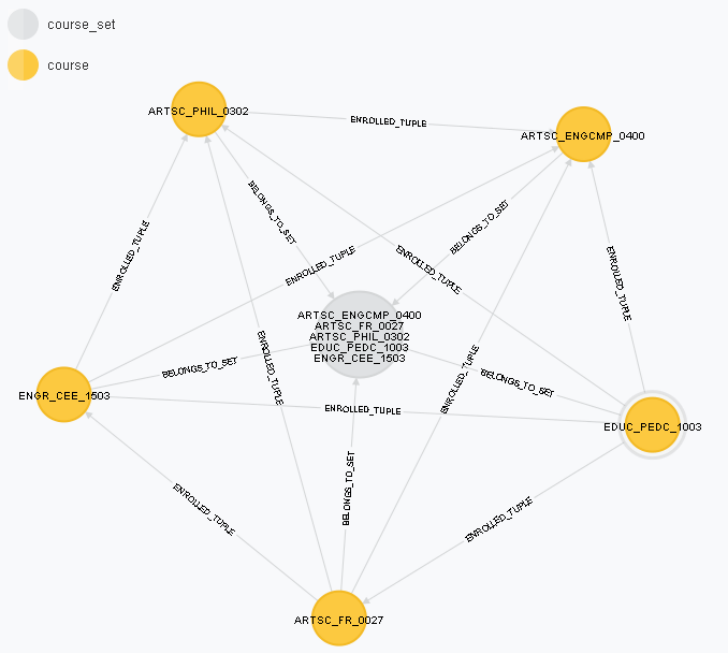


Figure 18 A sample closed itemset with the individual courses that belong to it. A closed itemset is a clique as all the courses in the itemset are connected to each other

Adding to the graph definitions presented in Section 5.1, we construct a hypergraph where the nodes are the cliques (closed itemsets), and edges are selectively created between pairs of cliques. This is done using the following methodology, which defines the clique overlap, the MULTI_PERCOLATE relationship between cliques, and the clique weighted overlap.

5.4.1 Clique Overlap

Clique Overlap: *The overlap between two or more cliques is the number of nodes that are present in those cliques* (i.e. Overlap is the number of courses that are in both course itemsets). An attribute of the clique nodes is a character string that identifies the course nodes that compose the clique (i.e. in the case at hand the ITEMSET attribute is a string identifying the courses in the clique). Thus, computing the overlap between two cliques is just a matter of comparing the two strings in the itemset attribute of each one of the cliques. That is done without the need to traverse the course graph as it would be required if using current approaches. In particular there is no need to compute a clique overlap matrix. For instance, Figure 19 below shows an example of three cliques of different sizes. A straightforward comparison of the character strings that identify the clique itemsets renders an overlap of two for the pairs of cliques that are connected with a "MULTI-PERCOLATE" edge, which is discussed next.

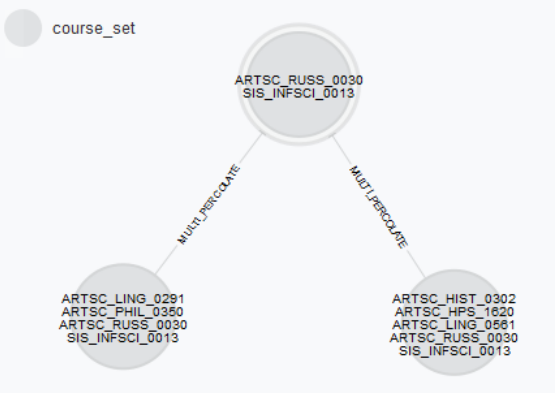


Figure 19 Sample of overlapping cliques of different sizes with overlap = 2

5.4.2 Multi-Percolate Edge

This sub-section discusses the procedure to determine if a MULTI-PERCOLATE edge needs to be created between two cliques. When comparing pairs of cliques to determine if they MULTI-PERCOLATE, three cases cover all the possibilities. *First,* the two cliques have size two (i.e. each one of them have two courses). *Second,* the two cliques have different sizes and the smallest clique has a size two. *Third,* the two cliques have a size larger than two.

Following is a specification of rules that generate MULTI-PERCOLATE edges linking cliques for these three cases that builds upon the literature on regular CPM. From the obtained results, the rules work well on course enrollment data. As part of future work, it would be necessary to test the rules using data from other domains. It is important to notice that the most salient aspect of GCPM is not the specific set of rules used here to generate the MULTI-PERCOLATE links, but that it enables the ability to link cliques of different sizes, using either the rules that follow or any other set of rules that might better fit different data sets.

CASE 1: The two cliques have size two

When two cliques have a size of two, and the overlap between them is one, then they multi-percolate into each other and an edge is created between them. That is, for every pair of cliques $\{i_i, i_j\} \subset I$, where the size of i_i is $k_i = 2$ and the size of i_j is $k_j = 2$; if $overlap(i_i, i_j) = 1$, then i_i and i_j multi-percolate into each other and an edge between them is created in the clique hypergraph indicating the relationship i_i *-[multi-percolate]-* i_j.

Figure 20 below illustrates this case with an example of two cliques of size two (2-clique) that multi-percolate into each other. Courses CBA_BUSACC_0030 and CBA_BUSMKT_1441 belong to the clique that contains both. As there are students enrolled in both courses, they are linked with the non-directional edge ENROLLED_TUPLE[2]. Courses CBA_BUSACC_0030 and

[2] The next section discusses the implementation of the graph in NEO4J. It requires the specification of a direction when creating a link. Afterwards relationships are equally well traversed in either direction without regard to the initial definition. However, the interface in version 2.1.2 of NEO4J used for this project does yet not enable control of the direction of the links (arrows) shown in the visualization.

ARTSC_COMMRC_0530 belong to the other clique in the example and both are also linked with an ENROLLED_TUPLE edge. The two cliques have size two and overlap in the course CBA_BUSACC_0030. Thus, they are linked with a MULTI_PERCOLATE edge.

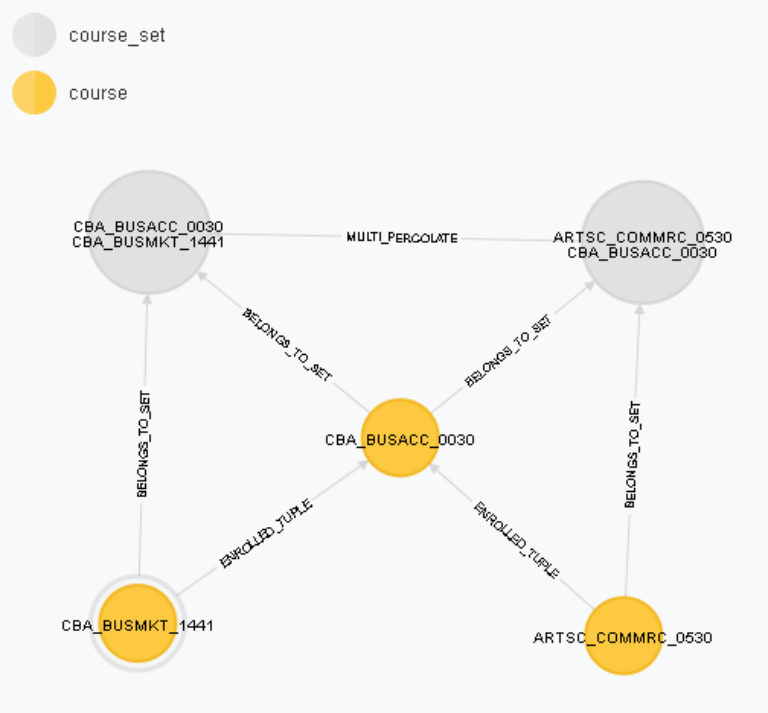


Figure 20 Example of clique multi-percolation when two cliques have a size k=2

CASE 2: The two cliques have different sizes and the smallest clique has a size two

When the smallest clique in a pair of cliques has a size two, and the overlap between the two cliques is two, then they multi-percolate into each other and an edge is created between them. That is, for every pair of cliques $\{i_i, i_j\} \subset I$, where the size of i_i is k_i and the size of i_j is k_j; if $min(k_i, k_j) = 2$ and $overlap(i_i, i_j) = 2$, then i_i and i_j multi-percolate into each other and an edge between them is created indicating the relationship i_i *-[multi-percolate]-* i_j.

Figure 21 below shows an example of clique percolation when the smallest clique in a pair has a size *k*=2. The three courses {ARTSC_PSY0010, ARTSC_MATH_0031, SIS_INFSCI0010} have students enrolled in common during ARCHIVE_PERIOD = 2141 as shown by the non-directional relationship "ENROLLED_TUPLE" between them. While ARTSC_PSY_0010 and SIS_INFSCI_0010 belong to one of the shown cliques, the three courses belong to the other. The two cliques are then linked through the MULTI-PERCOLATE edge with OVERLAP =2.

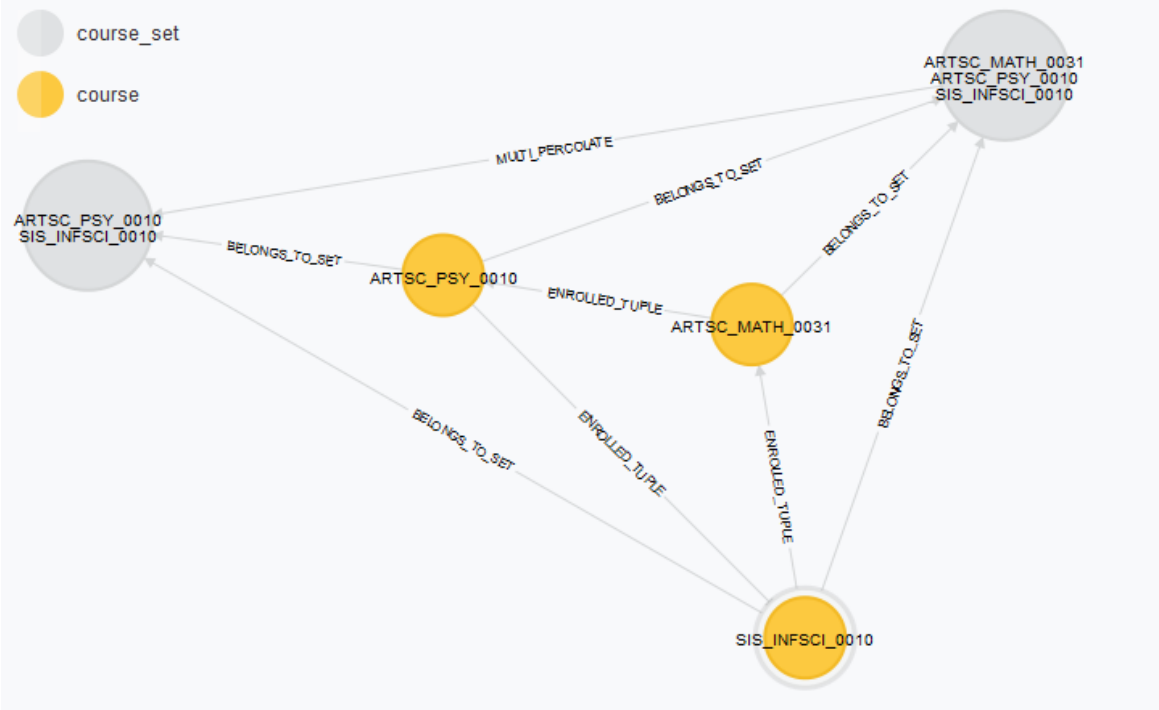


Figure 21 Example of clique multi-percolation when the smallest clique in a pair has a size *k*=2

CASE 3: The two cliques have a size larger than two.

When the smallest clique in a pair of cliques has a size larger than two and the overlap between the two cliques is at least the size of the smallest clique minus one, then they multi-percolate into each other and an edge is created between them. That is, for every pair of cliques

$\{i_i, i_j\} \subset I$, where the size of i_i is k_i and the size of i_j is k_j; if $min(k_i, k_j) > 2$ and $overlap(i_i, i_j) >= min(k_i, k_j) - 1$, then i_i and i_j multi-percolate into each other and an edge between them is created indicating the non-directional relationship i_i *-[multi-percolate]-* i_j. Figure 22 illustrates this case with two course itemsets where one has four courses, the other five, and they share three courses in common.

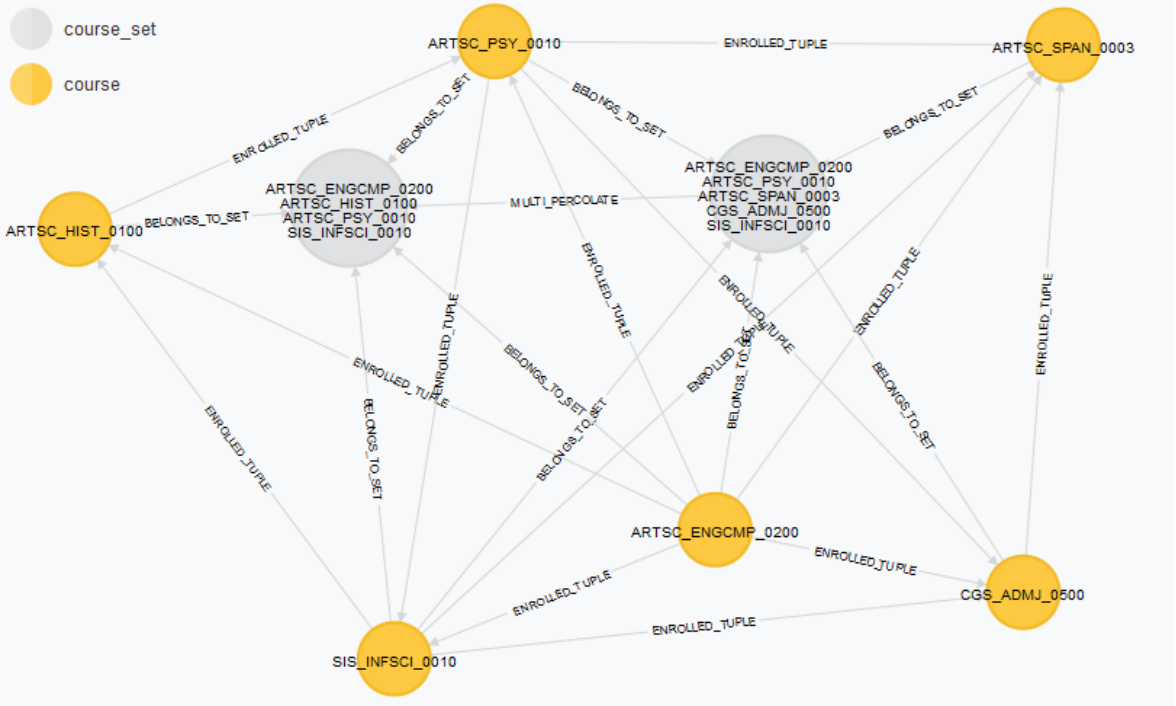


Figure 22 Example of clique multi-percolation when the smallest clique on a pair has a size $k > 2$

5.4.3 Clique Weighted Overlap

The *multi-percolate* edge that defines the relationship between a pair of cliques $\{i_i, i_j\}$ has two attributes that enable analyses of the graph using these metrics as filters. First, the overlap between the cliques $overlap(i_i, i_j)$ as described above. Second, following the discussion on the strategy to control for the effect of nodes that appear in large number of cliques dominating the

clique graph, a *weighted overlap* attribute is defined following Evan's specification (T. S. Evans, 2010).

Let $N = \{n_1, n_2,..n_k,..n_p\}$ be the set of nodes (courses) that are present in both cliques $\{i_i, i_j\}$. Let m_k be the number of cliques (course itemsets) on which node $n_k \in N$ is present. Then, $weighted\_overlap(i_i, i_j) = \sum 1/m_k, k = \{1,p\}$.

Notice that m_k is the number of cliques to which each course belongs. In the case at hand, the computation is performed after the graph is instantiated, and a property in the course nodes is set to record the value of m_k for each course.

5.4.4 Discussion of GCPM salient aspects

As an initial illustration of the results obtained with GCPM, Figure 23 below shows a sample of cliques of different sizes that percolate into each other with a weighted overlap > 0.1 and including courses offered by the School of Information Sciences (SIS) in archive period 2141. In the figure, each group of cliques that are interconnected forms a clique community with weighted overlap > 0.1. That is, every clique can be reached from all other cliques in the group through the multi-percolate links. The courses that belong to interconnected cliques form course communities at the specified clique weighted overlap.

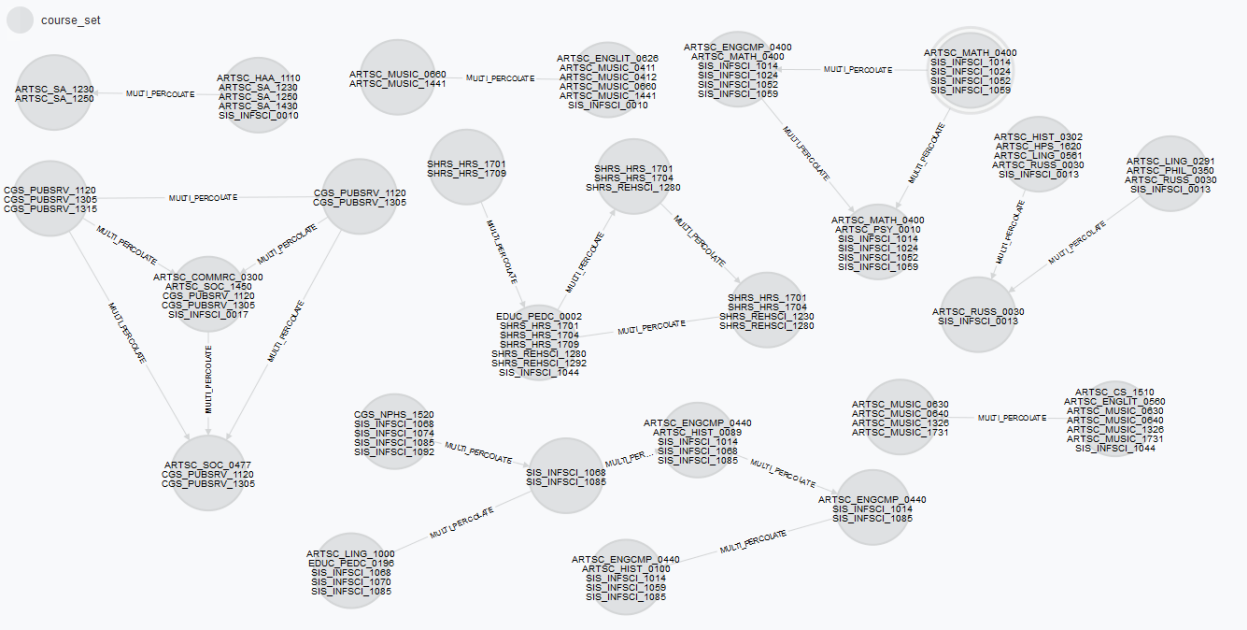


Figure 23 Sample of cliques of different sizes that multi-percolate into each other with a weighted overlap > 0.1 and including SIS courses in archive period 2141

The GCPM methodology results in the identification of cliques of equal or different sizes that multi-percolate into each other. The multi-percolate edges are created using local information regarding the overlap of cliques that have nodes in common according to the cases discussed above. The weighted overlap incorporates information on the global connections in the graph as it considers all the cliques that a node belongs to. Of central importance to the problem at hand, the clique node attributes include the number of students enrolled obtained with Association Rule Analysis, and the maximum number of seats available computed with MASAI. These attributes can now be naturally used as a filters when performing graph operations.

It is important to notice that the standard clique percolation method CPM is a special case of the GCPM method. One just need to query the multi-mode graph using the *overlap* attribute of the multi-percolate edges along with the attribute in the cliques (itemsets) that records their size to obtain fixed *k* clique graphs (e.g. query the graph for all the cliques with three courses (3-cliques) linked with a multi-percolate edge with overlap = 2).

The proposed approach is based on the multi-mode graph previously discussed, which enables seven key aspects:

- The cliques are available when the multi-mode graph is created (i.e. they are the closed course itemsets). Thus, while there is no need to perform a clique identification procedure, the identification of closed course itemsets takes just a few seconds using the apriori algorithm.

  Due to the characteristics of the timetabling problem at hand, we started with an Association Rule Analysis that led to the identification of cliques. However, this is just incidental as in the absence of the information on transactions we could have used any clique identification algorithm.
- The use of a multi-mode graph obviates the need to compute a clique overlap matrix, which is perhaps the most computationally expensive operation in standard CPM. In the multi-mode graph, a course node can belong to multiple cliques and both types of nodes have attributes that help turn the overlap computation into a graph traversal operation.
- The detection of percolating cliques is performed through a traversal of the clique graph comparing an attribute in the cliques that lists the nodes (courses) that form the clique.
- While traversing the clique graph, only cliques that multi-percolate into each other are linked. Thus, there is no need for an extra step thresholding the links by weight.
- All the information is maintained in the graph for future use. That, in contrast to existing approaches that discard information that is not deemed needed in order to reduce the complexity of the computations.
- In order to reduce the number of cliques to analyze, standard CPM is based on maximal cliques. In contrast, GCPM as discussed in this document is based on cliques that are closed itemsets in the Association Rule Analysis. Thus, GCPM uses all information available.
- Crucially, the identification of cliques of different sizes that percolate into each other is made possible. This enables the identification of overlapping and hierarchical communities.

The combination of a multi-mode graph and GCPM enable a natural way to perform longitudinal analyses on the graph either by considering individual snapshots (i.e. individual academic terms) or by considering a defined span of time as a whole (i.e. taking multiple individual academic terms as a single observation). Furthermore, we are able to take the global view of the graph to identify communities of interest in a top-down fashion or conversely take

the ego-centered approach by starting with a node of interest and identifying the communities to which it belongs.

Although, the focus of this work is on timetabling in higher education, from the discussion of the proposed methods and initial comparison of results with other works on the topic presented in Section 5.4., it is apparent that they provide a general framework to deal with a variety of problems that can be modelled as a graph community identification problem.

Before delving further into the discussion of results and in order to facilitate it, the following section presents details of the implementation of the graph database in NEO4J.

5.5 MULTI-MODE GRAPH DATABASE IMPLEMENTATION USING NEO4J

A prototype implementation of the multi-mode graph has been performed using the graph database system NEO4J (Miller, 2013; Webber, 2012) and its query language Cypher (NEO4J). Enrollment data on the six academic terms under study have been loaded. Objects in the graph database are labelled with an integer identifier and have multiple attributes that fully identify them. When applicable and with the goal of facilitating compatibility, the fields available in the relational schema discussed in 4.2 are used as attributes in the graph database. Appendix C includes the scripts that create the objects and relationships, and load the data into NEO4J. Following is the specification of nodes and edges in the implemented graph database.

5.5.1 Nodes

- *COURSE_SET:* These are the clique nodes or itemset nodes. The attributes are Node Identifier (generated by NEO4J), Itemset identifier (ID from relational database), Academic Term (ARCHIVE_PERIOD), set of courses in the itemset (ITEMSET), maximum number of seats possible at the beginning of enrollment period (MAX_INIT_CAPACITY), maximum number of seats available at the end of enrollment period (MAX_SEATS_LEFT), number of students enrolled in itemset (ENROLLED), number of courses in the itemset

(NBR_COURSES_ITEMSET), number of courses in the itemset with seats available (CRS_WITH_SEATS_ITEMSET).

The implemented graph database has the following count of COURSE_SET nodes per term 2091: 42,611; 2101: 43,897; 2111: 45,595; 2121: 46,591; 2131: 47,145; 2141: 47,479. These figures are obtained using the following Cypher query

```
        MATCH (n:`course_set`)
        RETURN n.ARCHIVE_PERIOD, count(n)
ORDER BY n.ARCHIVE_PERIOD
```

- *COURSE:* These nodes represent the courses. The attributes are: Node Identifier (generated by NEO4J), Academic Term (ARCHIVE_PERIOD), course descriptor (COURSE -e.g. CBA_BUSMKT-1441) Academic Unit or School offering the course (ACAD_GROUP_CD), Subject Descriptor (SUBJECT_DESCR), Subject Code (SUBJECT_CD), Catalog Number (CATALOG_NBR), number of students enrolled in course (ENROLLED_CRSE), number of sections offered (SECTIONS), enrollment limit in all sections offered in course (CAP_CRSE), course title (COURSE_DESCR), the number of itemsets (cliques) to which the course belongs (ITEMSETS), and the degree of the course (DEGREE).

  The implemented graph database has the following count of COURSE nodes per term 2091: 1,737; 2101: 1,763; 2111: 1,828; 2121: 1,922; 2131: 2,014; 2141: 2,078. These figures are obtained using the following Cypher query

```
        MATCH (n:`course`)
        RETURN n.ARCHIVE_PERIOD, count(n)
ORDER BY n.ARCHIVE_PERIOD
```

- *SECTION:* These nodes represent the sections of a course. The attributes are: Node Identifier (generated by NEO4J), Academic Term (ARCHIVE_PERIOD), Academic Unit or School offering the course (ACAD_GROUP_CD), Subject Descriptor (SUBJECT_DESCR), Subject Code (SUBJECT_CD), Catalog Number (CATALOG_NBR), Class Number (CLASS_NBR), section enrollment limit (ENRL_CAP), total enrollment in section (ENRL_TOT), start time (STARTT), stop time (STOPT), yes/no attributes indicating the

days of the week when sessions are scheduled (MON, TUES, WED, THURS, FRI), and class room identifier (FACILITY_ID).

The implemented graph database has the following count of "SECTION" nodes per term 2091: 5,842; 2101: 5,918; 2111: 6,150; 2121: 6,464; 2131: 6,718; 2141: 6,958. These figures are obtained using the following Cypher query

```
        MATCH (n:`section`)
        RETURN n.ARCHIVE_PERIOD, count(n)
ORDER BY n.ARCHIVE_PERIOD
```

5.5.2 Edges

- *ENROLLED_TUPLE:* Are undirected edges connecting adjacent course nodes. They have the attribute "enrolled", which refers to the number of students enrolled in the courses that the edge links. If *a* and *b* are two adjacent course nodes course nodes with 20 students enrolled in both courses, then using the syntax of the NE04J query language Cypher, the relationship between the two nodes is expressed as[3]

```
(a:COURSE) -[:`ENROLLED_TUPLE` {enrolled: 20}]- (b:COURSE)
```

- *BELONGS_TO_SET:* Are directed edges that link courses (nodes) with itemsets (cliques). They indicate that a course belongs to or is a member of an itemset. If *a* is a course node and *i* is a course_set node, then in Cypher the relationship is expressed as

```
(a: COURSE) -[:`BELONGS_TO_SET`]-> (i: COURSE_SET)
```

- *BELONGS_TO_COURSE:* Are directed edges that link section nodes with course nodes. They indicate that a section belongs to a course or in other words that is an offering of a

[3] A detail to note is that NEO4J always stores relationships/edges as directed, but when traversing/querying it treats the graph as undirected unless a direction is specified.

course. If a is a course node and s is a section node, then in Cypher the relationship is expressed as

```
(s: SECTION) -[:`BELONGS_TO_COURSE`]-> (a: COURSE)
```

5.6 GCPM IMPLEMENTATION IN NEO4J

GCPM is instantiated through a MULTI-PERCOLATE edge linking course sets (cliques) following the methodology explained in Section 5.4. This section discusses the process and Cypher queries that generate the MULTI_PERCOLATE edges.

*MULTI-PERCOLATE***:** Are non-directed edges that link course sets (cliques). They indicate that two course sets multi-percolate into each other as defined in the Section 5.4. If *a* is a course set node and *c* is a course set, then in Cypher the relationship is expressed as

```
(a: COURSE_SET) -[:`MULTI-PERCOLATE`]- (c:COURSE_SET)
```

The Cypher query below creates a MULTI-PERCOLATE edge between pairs of cliques that meet the criteria specified in Section 5.4. It does so in one traversal of the multi-mode graph creating the MULTI-PERCOLATE edges and setting the OVERLAP attribute. Following from the discussion in section 5.4.2, it is important to notice that it would be straightforward to implement a different set of rules to generate the MULTI-PERCOLATE edges if the need arises.

Line 1 in the query traverses the graph matching pairs of cliques (`course_set`) that have courses (`course`) in common. That is, when a course belongs to two course sets then that course is common to both course sets. In that sense, GCPM is a local or ego approach to the identification of communities as it starts from the cliques to which each course node belongs. This operation is possible because the design of the multi-mode graph permits the direct linking of course nodes in one mode with the cliques in the other mode, which leverages NEO4J's capabilities.

Line 2 optionally limit the process to the specified academic term (ARCHIVE_PERIOD). Line 3 avoids double matching between pairs of course sets. Without this line, course sets would be matched twice, as (a:`course_set`) and (b:`course_set`), and an edge would be created for each direction.

```
1  MATCH (a:`course_set`) <-[k:`BELONGS_TO_SET`]- (c:`course`) –
      [m:`BELONGS_TO_SET`]-> (b:`course_set`)
2  WHERE a.ARCHIVE_PERIOD = 2091
3   AND a.ITEMSET_ID < b.ITEMSET_ID
4   AND (
5         (
6               a.NBR_COURSES_ITEMSET = 2
7      AND b.NBR_COURSES_ITEMSET = 2
8      AND length(FILTER(x IN split(a.ITEMSET, " ")
            WHERE x IN split(b.ITEMSET, " "))) = 1
9      )
10       OR
11     (
12        (a.NBR_COURSES_ITEMSET + b.NBR_COURSES_ITEMSET
         - ABS(a.NBR_COURSES_ITEMSET - b.NBR_COURSES_ITEMSET))/2 =2
13      AND length(FILTER(x IN split(a.ITEMSET, " ")
         WHERE x IN split(b.ITEMSET, " "))) =2
14     )
15       OR
16       (
17        (a.NBR_COURSES_ITEMSET + b.NBR_COURSES_ITEMSET
         - ABS(a.NBR_COURSES_ITEMSET - b.NBR_COURSES_ITEMSET))/2 > 2
18      AND length(FILTER(x IN split(a.ITEMSET, " ")
         WHERE x IN split(b.ITEMSET, " "))) >=
         (a.NBR_COURSES_ITEMSET + b.NBR_COURSES_ITEMSET
         - ABS(a.NBR_COURSES_ITEMSET - b.NBR_COURSES_ITEMSET))/2 -1
19     )
20      )
21 WITH a, b,
22    length(FILTER(x IN split(a.ITEMSET, " ")
         WHERE x IN split(b.ITEMSET, " "))) as ab_overlap,
23    collect(c.COURSE) as crses
24 FOREACH (r in crses | create unique
      (a) -[l:`MULTI_PERCOLATE` {OVERLAP: ab_overlap}]-> (b));
```

Lines 4 to 20 match itemsets according to their overlap and size as follows:

- Lines 6 to 8 take care of the case when the two cliques have size 2 (*2*-cliques) and the course overlap is one. To that goal, we use the clique attribute NBR_COURSES_ITEMSET as shown in lines 6 and 7. This attribute holds the number of courses present in a clique. Line 8

filters pairs of cliques with an overlap of one course. It uses the clique attribute ITEMSET, which holds the string with the courses in the clique, and compares the two strings.

- Lines 12 and 13 deal with the case when the smaller of the two cliques being matched has a size of two. In that case, if the overlap between the two cliques is also two, then an edge is created. To that goal, line 12 uses the number of courses in each clique and determines if the minimum is equal to two. Having the number of courses in each one of the two matched cliques, the minimum of the two figures (positive integers) is obtained using

$$min(a, b) = (a + b – abs(a- b)) / 2$$

  Line 13 determines if the overlap between the two cliques being considered is equal to two, using the cliques attribute ITEMSET in the same way as explained in the previous case.
- Lines 17 and 18 handle the case when the smallest clique has more than two courses. In this case, we want to create an edge when the overlap between every matched pair of cliques is larger or equal than the number of courses in the smallest clique minus one. Line 17 computes the minimum size of the two cliques as explained in the previous case, and matches the cases when the minimum is larger than two. Line 18 computes the overlap between the two cliques using the ITEMSET attribute, and guarantees that the number of common courses present in the matched pair of cliques is larger or equal than the number of courses in the smallest clique minus one.

Lines 21, 22 and 23 return the collection of matched courses along with the pair of cliques and the overlap between the cliques. Line 24, goes through the collection creating a unique MULTI_PERCOLATE edge for every couple of matched cliques, and setting the OVERLAP attribute for the edge.

Figures 24 and 25 below further illustrate the described GCPM methodology using as example the course SCI-FI: EAST AND WEST (ARTSC_SLAV_0660) offered by the department of Slavic Languages and Literature (SLAV) at the Dietrich School of Arts and Sciences (ARTSC) with catalog number 0660. In the academic term 2091 (August to December 2008), there were 115 students enrolled in three sections of this course, and it was present in 211 cliques. Figure 24 shows the course node for ARTSC_SLAV_0660 and eight of the cliques to

which it belongs (BELONGS_TO_SET) before GCPM identifies the cliques that multi-percolate into each other (i.e. the cliques that need to be linked with MULTI_PERCOLATE edges). Figure 25 shows the same nodes after the cliques have been linked with MULTI-PERCOLATE nodes using the GCPM Cypher implementation described above.

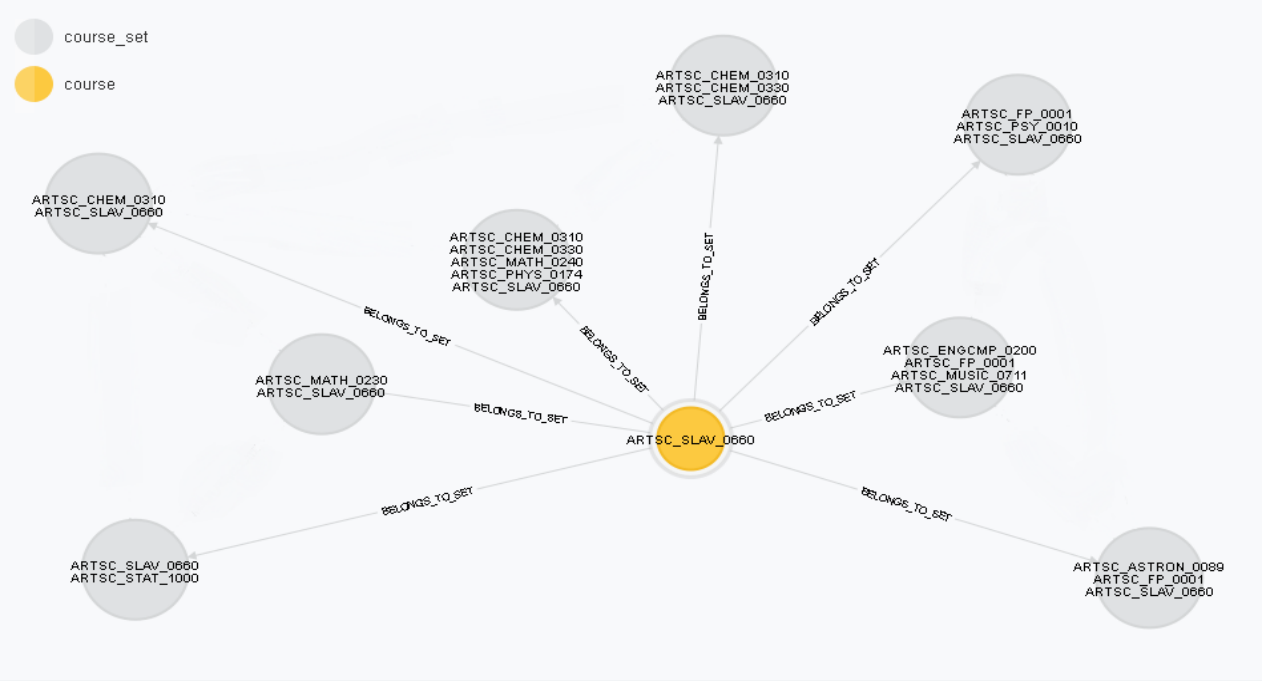


Figure 24 Before GCPM: Course ARTSC_SLAV_0660 with sub-set of eight cliques of multiple sizes to which it belongs

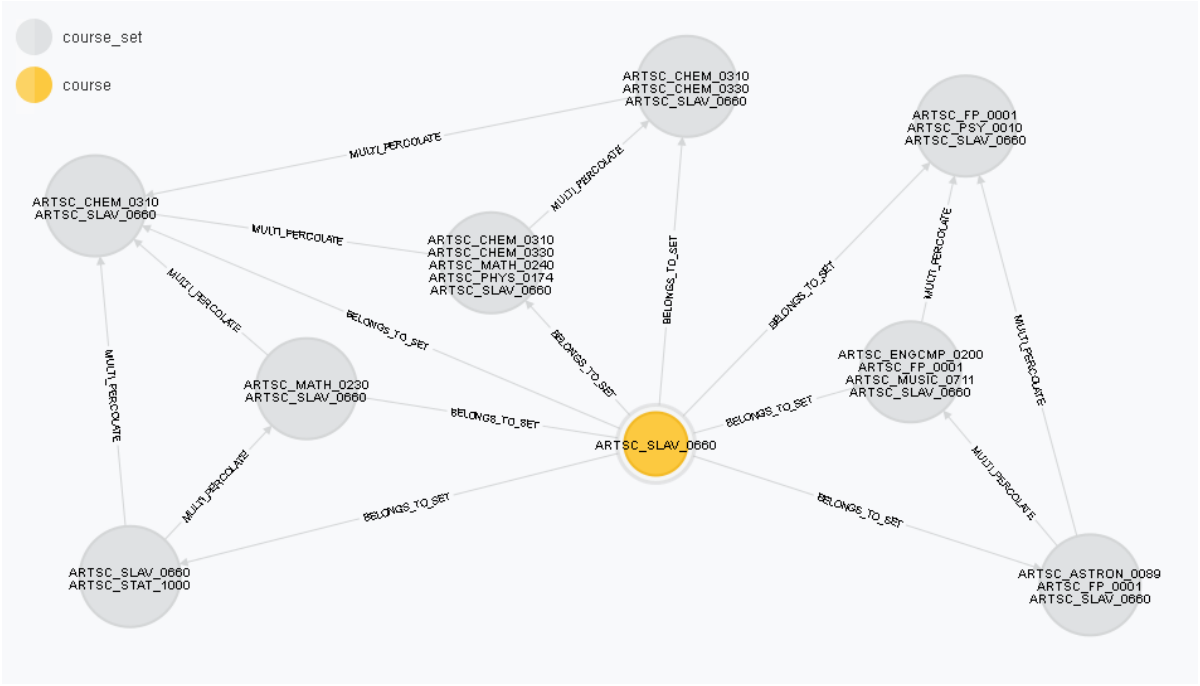


Figure 25 After GCPM: Course ARTSC_SLAV_0660 with sub-set of eight cliques of multiple sizes to which it belongs, and MULTI-PERCOLATE edges linking cliques

Setting the attribute WEIGHTED_OVERLAP in a MULTI-PERCOLATE edge requires the knowledge of the number of cliques (ITEMSETS) to which every course common to the cliques linked by the MULTI-PERCOLATE edge belongs. The Cypher query below sets the attribute ITEMSETS for every course. The query matches every course with the course sets to which it belongs and then counts the number of edges "BELONGS_TO_SET". Then, the attribute ITEMSETS is set to the total count of cliques to which the course belongs.

```
MATCH (a:`course`)-[k:`BELONGS_TO_SET`]->(b:`course_set`)
WITH a as crse, count(k)as cnt
set crse.ITEMSETS = cnt;
```

The Cypher query below sets the value of the attribute WEIGHTED_OVERLAP on every MULTI-PERCOLATE edge. For every pair of cliques linked with a MULTI-PERCOLATE edge it matches the courses that are present in both cliques (d:`course`). Then, the *weighted overlap* is computed as the sum of the inverse of the value of the ITEMSETS attribute in those courses. The WEIGHTED_OVERLAP attribute brings in a global approach to the graph

analysis as it makes available information beyond the immediate vicinity of the percolating cliques that is used to generate the MULTI_PERCOLATE edges.

```
MATCH (d:`course`) -[m:`BELONGS_TO_SET`]->(a:`course_set` )
    -[c:`MULTI_PERCOLATE`]-
    (b:`course_set`) <-[k:`BELONGS_TO_SET`]-(d:`course`)
WHERE a.ARCHIVE_PERIOD = 2091
 AND id(a) < id(b)
 WITH c, sum(round(toFloat(100000/d.ITEMSETS))/100000) as WO
 SET c.WEIGHTED_OVERLAP = WO;
```

5.7 RESULTS OF MULTI-MODE GRAPH AND GCPM PROTOTYPE IMPLEMENTATION

The described prototype implementation of the multi-mode graph database using Pitt's enrollment data from six fall academic terms render the number of nodes and edges/ relationships shown in Tables 25 and 26 below.

Table 25 Count of nodes in prototype multi-mode graph database

Nodes			
Archive Period	**Sections**	**Courses**	**Closed Course Itemsets (Cliques)**
2091	2,921	1,488	42,611
2101	2,959	1,465	43,897
2111	3,075	1,484	45,595
2121	3,232	1,525	46,591
2131	3,359	1,546	47,145
2141	3,478	1,538	47,479
Total	**19,024**	**9,046**	**273,318**

Table 26 Count of edges / relationships in prototype implementation of multi-mode graph database

Edges / Relationships				
Archive Period	**Belongs to Course**	**Enrolled Tuple**	**Belongs to Set**	**Multi-Percolate**
2091	2,921	69,212	146,006	3,874,947
2101	2,959	70,614	150,408	3,993,547
2111	3,077	72,062	156,411	4,256,677
2121	3,236	72,942	160,342	4,363,354
2131	3,369	74,064	162,261	4,303,479
2141	3,490	73,615	164,359	4,429,996
Total	**19,052**	**432,509**	**939,787**	**25,222,000**

These results are in line with literature on applications that use clique graphs for the identification of communities in the sense that clique graphs are generally orders of magnitude bigger than the original network, especially in densely connected graphs (T. S. Evans, 2010). This can be seen in Table 26 by comparing the number of multi-percolate edges versus the number of other edges in the graph.

As the standard Clique Percolation Method CPM is a special case of the proposed Generalized Clique Percolation Method GCPM, then it is straightforward to compare the results obtained with both methods. The Cypher query below produces the total count of cliques of the same size per academic term shown in Table 27 below in column "Total". A second run of the query with line three uncommented renders the number of *2*-clique or other sizes. The column "Other k-cliques" is equal to the Total minus the *2*-clique. The column percentage of GCPM Multi-percolate edges is obtained using the figures from Table 26.

```
MATCH (a)-[c:`MULTI_PERCOLATE`]->(b)
WHERE a.NBR_COURSES_ITEMSET = b.NBR_COURSES_ITEMSET
//AND a.NBR_COURSES_ITEMSET = 2 //3 //4 //5 //6 //7 //8 //9
RETURN a.ARCHIVE_PERIOD, count(c) order by a.ARCHIVE_PERIOD
```

In the data set used for this work, after eight runs (one for each size of cliques of *2*-clique to *9*-clique) CPM would identify 45.4% of the relationships between cliques that GCPM

identifies in one run. Regardless of the number of runs of CPM, it will not able to identify the relationships between cliques of different sizes. In alignment with results reported in the literature, there is a rapid loss of coverage when using CPM as the size of the cliques is increased. This is apparent from the low percentage of cliques of size larger than two (13.1%). In contrast, GCPM identifies 13,761,160 relationships between cliques of different sizes (25,222,000 – 11,460,840).

Table 27 Number of Percolate edges identified with regular CPM versus Multi-Percolate edges identified with GCPM

Archive Period	CPM Percolate			% of GCPM Multi-Percolate
	2-clique	Other K-cliques	Total	
2091	1,565,070	229,375	1,794,445	46.3%
2101	1,608,886	234,581	1,843,467	46.2%
2111	1,664,179	253,568	1,917,747	45.1%
2121	1,699,083	259,854	1,958,937	44.9%
2131	1,734,810	255,686	1,990,496	46.3%
2141	1,687,865	267,883	1,955,748	44.1%
Total	**9,959,893**	**1,500,947**	**11,460,840**	**45.4%**
% of CPM Total	**86.9%**	**13.1%**	**100.0%**	

Another important result of the GCPM method described in Section 5.2 is that the computation time of the clique graphs per academic term was between two and half and three hours on a regular machine with an Intel i5 -200 CPU@ 3.3 GHZ with 16 Gigabytes of RAM memory, running the 64-bit version of Windows 7 Enterprise. This results is obtained while generating clique graphs with no restrictions on clique sizes. In order to illustrate the size of the problem, a naïve approach to determine which pairs of the 47,479 cliques in academic term 2141 multi-percolate into each other would need 1.13 billion comparisons to be made ($m*(m-1)/2$). In the proposed approach, the multi-mode graph enables the implementation of GCPM using the Cypher queries discussed in Section 5.4, which substantially reduce the number of operations required to identify pairs of cliques that multi-percolate into each other. For instance, below we

take the first three lines of the Cypher queries that instantiate the MULTI-PERCOLATE edges between cliques, and add a fourth line to return the count of edges of type BELONGS_TO_SET. The result is 56.6 million matches for academic term 2141 and processing takes ten minutes.

```
1  MATCH (a:`course_set`) <-[k:`BELONGS_TO_SET`]- (c:`course`) –
      [m:`BELONGS_TO_SET`]-> (b:`course_set`)
2  WHERE a.ARCHIVE_PERIOD = 2141
3   AND a.ITEMSET_ID < b.ITEMSET_ID
4 RETURN COUNT(K)
```

In contrast, literature on the topic of clique percolation refers that the computation time to generate the clique graph has been the main obstacle when using CPM. Reid et.al. (2012) refer to benchmarks for computation time that they performed on several networks using several clique percolation algorithms, which can only perform clique percolation on cliques of fixed size: "Similarly, looking at maximal cliques of size greater than 5 with intersection of 4 nodes, we generated over 1,700,000 edges over a period of several days of computation. We note that the total number of edges in the clique graph, in this small network with only 769 nodes, may be even much larger than this" (Reid et al., 2012).

In order to benchmark the results obtained with GCPM, we would need to use it on the same networks that Reid and colleagues used for their work. That benchmarking work is beyond the scope of this dissertation and is thus left for future research.

5.7.1 Course Degree, and Course Enrollment Weighted Degree Centrality (EWDC)

In a graph or network, *Degree is the number of links incident on a node or, in other words, the number of nodes that a focal node is connected to*. Degree measures the level of involvement of the node in the network (Henning, Brandes, & Pfeffer, 2012). There are numerous metrics that can be used to study networks, among the most popular are Closeness Centrality and Betweeness Centrality. Their characteristics would not bring added value as global metrics if used for the case under analysis.

Closeness Centrality is the inverse sum of the shortest distance to all other nodes from a focal node. Thus, it cannot be applied to networks with disconnected components or communities as two nodes that belong to different communities do not have a finite distance between them. Betweeness Centrality measures the degree to which a node lies in the shortest path between any two other nodes. In the case at hand, a large proportion of nodes do not lie on a shortest path between any two other nodes, and thus would receive a score of zero or close to zero (Freeman, 1977, 1979). Closeness Centrality and Betweeness Centrality are useful locally to identify nodes of importance in communities as all nodes in a community have paths between them (Henning et al., 2012; Opsahl, Agneessens, & Skvoretz, 2010).

In the case at hand, we have a multi-mode graph where the core level is a network of courses linked to each other through edges that are weighted by the number of students mutually enrolled in the courses at each end of the edge. Thus it is of interest to know not only which course nodes have the highest Degree, but also to consider the importance of the number of edges incident on a node relative to the weight of those edges. In order to capture that information, we propose a metric called Enrollment Weighted Degree Centrality (EWDC) as discussed ahead.

We start by setting the attribute DEGREE for every course node in the graph using the Cypher query below. The query matches every pair of courses linked with an ENROLLED_TUPLE edge. For every course, it counts the number of ENROLLED_TUPLE edges and then sets the attribute DEGREE to the total count.

```
        MATCH (a:course) -[e:ENROLLED_TUPLE]- (b:course)
        WHERE a.ARCHIVE_PERIOD = 2091
         WITH a, count(e) as d, collect(id(a)) as crses
FOREACH (r in crses | set a.DEGREE = d)
```

The course node Degree attribute enables the analysis shown in Table 28 below. The first line in the table shows a high correlation index between the course enrollment and Degree. The second line shows and almost perfect correlation between course enrollment and total weight[4].

[4] The Cypher query below returns figures needed for the analysis in Table 28: Archive period, course, number of students enrolled, degree of the course, and sum of all common enrollments with other course nodes with a common edge (i.e. the total edges weight for the course node).

This is expected as the larger the number of students enrolled in a course, the larger the number of distinct combinations of courses that those students enroll in, and consequently the larger the number of edges linking the course under consideration with other courses. Thus, in the case at hand the regular Degree metric does not add substantial information that could not be obtained from the straight figures on course enrollment and edge total weight in the course to course sub-graph.

The referred characteristic of the problem at hand calls for a metric that weights the Degree of a course node based on the number of enrolled students. A natural metric is the number of edges incident on a course node per enrolled students in the course. *That is, the Enrollment Weighted Degree Centrality (EWDC) of a course node is the Degree of a course node divided by the number of students enrolled in the course*[5].

Table 28 Correlations between course enrollments and various graph metrics

	2091	2101	2111	2121	2131	2141
Course Enrollment vs. Degree	0.731	0.725	0.717	0.716	0.676	0.689
Course Enrollment vs. Total Weight	0.978	0.976	0.973	0.975	0.914	0.941
EWDC vs. Course Enrollment	(0.382)	(0.401)	(0.407)	(0.404)	(0.402)	(0.394)
EWDC vs. Degree	(0.191)	(0.202)	(0.211)	(0.220)	(0.232)	(0.254)
EWDC vs. Opsahl	(0.326)	(0.342)	(0.353)	(0.357)	(0.369)	(0.380)
Opsahl vs. Enrolled (α = 0.5)	0.929	0.923	0.917	0.917	0.860	0.885

The third and fourth lines in Table 28 show the EWDC correlation indexes with course enrollment and Degree. The correlations are small and negative. In particular, the correlations

```
MATCH  (a:course) -[e1:ENROLLED_TUPLE]- (b:course)
RETURN  a.ARCHIVE_PERIOD as ARCHIVE_PERIOD,
        a.COURSE as COURSE,
        a.ENROLLED_CRSE as ENROLLED_CRSE,
        a.DEGREE as DEGREE,
        sum(e1.enrolled) as TOTAL_EDGES_WEIGHT
ORDER BY  a.ARCHIVE_PERIOD,  a.DEGREE desc
```

[5] An alternate metric that would be more in alignment with the network analysis literature is the degree/ total weight of enrolled tuple edges. In this case EWDC is preferred because it provides equivalent results and might be easier to understand for users of a timetabling system who might not be familiar with network analysis. In any case, both metrics could be implemented on a system.

between EWDC and Degree are the smallest indicating that EWDC provides different information than Degree[6].

Opsahl and colleagues propose a degree metric generalization that combines the number of edges and their weight. They propose a "degree centrality measure, which is the product of the number of nodes that a focal node is connected to, and the average weight to these nodes adjusted by [a] tuning parameter [alpha (α)]." Although their approach appears to successfully consider the number of edges and their weights in the examples provided in their paper, a limitation that they acknowledge is that the there is no clear way to determine the best suited alpha parameter for an analysis (Opsahl et al., 2010).

Lines five and six in Table 28 show the correlation indexes between EWDC, and the Degree Centrality measure proposed by Opsahl and colleagues. Line six shows the correlation between Degree Centrality using an alpha parameter of 0.5 ($\alpha = 0.5$). An alpha of zero ($\alpha = 0$) renders a Degree Centrality that is equal to the standard Degree metric. An Alpha of one ($\alpha = 1$) renders a Degree Centrality that is equal to the Total Weight of the edge. The high correlations shown in Table 27 indicate that in the case at hand, the Degree Centrality metric as proposed by Opsahl and colleagues would not add substantial information beyond what can be derived from the straight enrollment figures.

5.7.2 The Course sub-graph is scale-free on Degree and random on EWDC

The chart in Figure 26 below shows that the course to course sub-graph has scale-free characteristics as indicated by the high fit ($R^2 = 0.7341$) with a power law distribution, even without removing outlier nodes. That is, there is a good fit with a linear relationship between the $\log_{10}$ of the number of course nodes versus $\log_{10}$ of the number of edges (Barabási, 2009; Barabási & Bonabeau, 2003). This means that there are course nodes that are hubs in the network. In other words, there is a relatively low number of courses that have a disproportionally

[6] Appendix D contains statistics on course nodes enrollment, degree and enrollment weighted degree centrality (EWDC)

large number of links with other courses. More precisely, 88% of the course nodes have less than 200 links with other course nodes, 10% have between 200 and 399, and 3% have more than 400.

The chart in Figure 27 below shows the distribution of course nodes versus EWDC on a $\log_{10}$. The chart illustrates that the course sub-graph exhibits characteristics of a random network on EWDC. That is, the EWDC metric controls for the effect of courses with large enrollment and provides a different perspective on the relationship between courses. It enables the identification of courses with a large number of links per students independently of the total enrollment in the course (i.e. highly connected nodes that are not hubs).

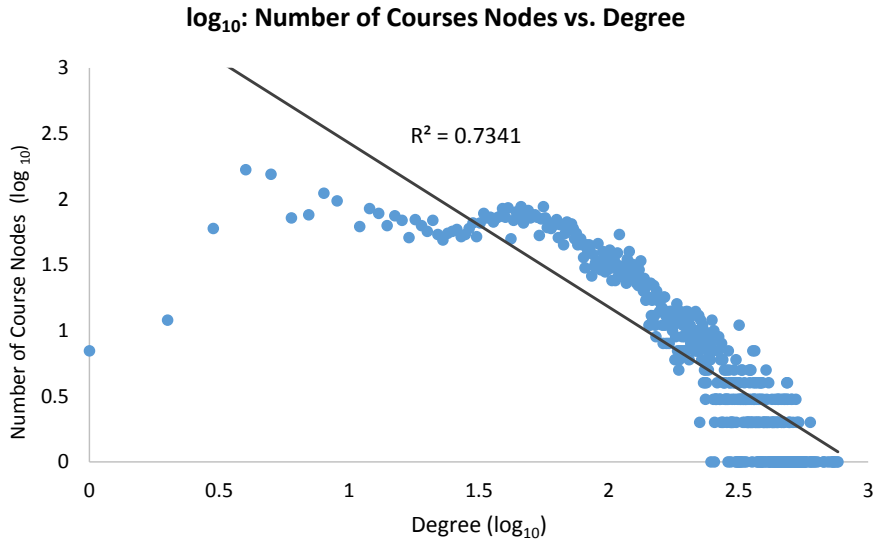


Figure 26 Power law distribution of course node edges on Degree (i.e. scale-free)

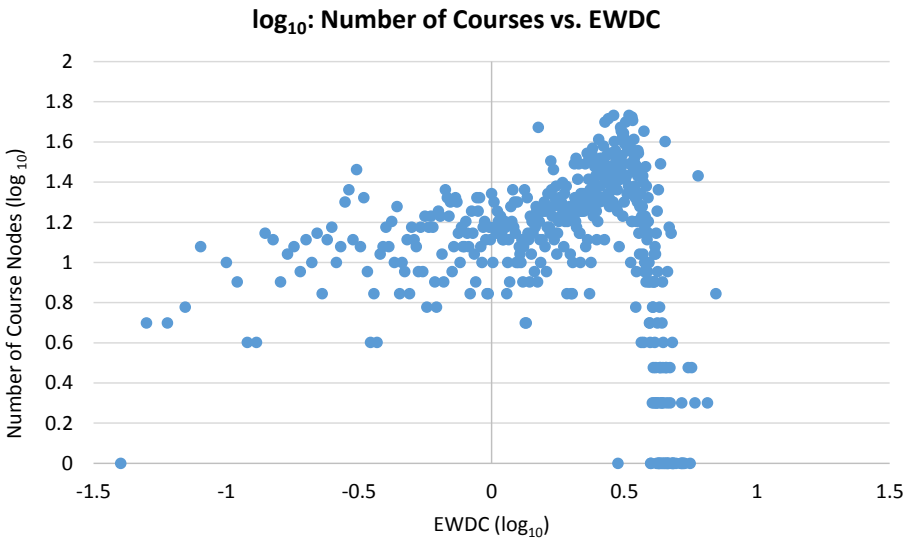


Figure 27 Distribution of course node edges on EWDC (i.e. random)

The use of both metrics, Degree and EWDC, provide complementary information. The scale-free characteristics of the course sub-graph on Degree suggests that targeted changes on the offerings of some of the courses that are hubs would have a large impact on the structure of the graph and consequently on the quality of the schedules. The random characteristics of the course sub-graph on EWD enables the identification of courses that although not necessarily enroll a large number of students, have a high impact on the quality of the overall schedule.

5.7.3 Examples illustrating Degree and EWDC on the course sub-graph

In order to illustrate the discussed concepts, Tables 29, 30 and 31 below show the top 20 courses with the highest average Enrollment, Degree, and Weighted Degree Centrality over the six academic terms under study. The analysis takes the global approach by presenting the results for the complete graph, which includes all courses in the data set. An identical approach can be used to identify courses of interest at the local level of individual schools and/or departments.

In alignment with the discussion above, while Table 30 includes courses with high enrollment, Table 31 includes courses that although not necessarily highly enrolled, have a high number of incident edges per enrolled student. Table 32 lists courses at the other end of the EWDC range; that is the 20 courses with the lowest average EWDC over the six academic terms under study.

Table 29 Top 20 Courses by average enrollment in academic terms 2091 to 2141

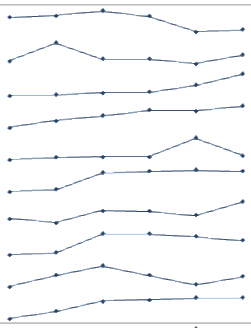
Line #	Course	Course Title	Average Enrollment	Enrollment: Terms 2091 to 2141	Average Degree	Average EWDC
1	ARTSC_PSY_0010	INTRODUCTION TO PSYCHOLOGY	1,860		740.7	0.399
2	ARTSC_FP_0001	INTRO TO THE ARTS & SCIENCES	1,449		417.8	0.289
3	ARTSC_PHYS_0174	BASC PHYS SCI & ENGR 1 (INTGD)	1,401		297.2	0.233
4	ARTSC_CHEM_0110	GENERAL CHEMISTRY 1	1,348		444.3	0.331
5	ARTSC_BIOSC_0150	FOUNDATIONS OF BIOLOGY 1	1,304		488.8	0.404
6	ARTSC_MATH_0220	ANALYTC GEOMETRY & CALCULUS 1	1,134		479.7	0.425
7	ARTSC_ENGCMP_0200	SEMINAR IN COMPOSITION	1,132		481.2	0.428
8	ARTSC_PHYS_0110	INTRODUCTION TO PHYSICS 1	1,089		513.5	0.474
9	ARTSC_BIOSC_0050	FOUNDATIONS OF BIOLOGY LAB 1	1,082		463.3	0.429
10	ARTSC_ECON_0100	INTRO MICROECONOMIC THEORY	1,037		626.2	0.604
11	ARTSC_CHEM_0310	ORGANIC CHEMISTRY 1	868		529.7	0.613
12	ARTSC_CHEM_0960	GENERAL CHEM FOR ENGINEERS 1	686		106.7	0.156
13	ARTSC_CHEM_0330	ORGANIC CHEMISTRY LABORATORY 1	644		467.7	0.726
14	ARTSC_ANTH_0780	INTRO TO CULTURAL ANTHROPOLOGY	603		554.8	0.925
15	ARTSC_ECON_0110	INTRO MACROECONOMIC THEORY	540		526.7	0.979
16	ARTSC_COMMRC_0520	PUBLIC SPEAKING	515		612.3	1.194
17	ARTSC_MATH_0240	ANALYTC GEOMETRY & CALCULUS 3	503		349.0	0.699
18	ARTSC_STAT_0200	BASIC APPLIED STATISTICS	492		529.0	1.077
19	ARTSC_MATH_0120	BUSINESS CALCULUS	491		365.5	0.745
20	ARTSC_PSY_0310	DEVELOPMENTAL PSYCHOLOGY	471		535.8	1.139

Table 30 Top 20 Courses with the highest average Degree in academic terms 2091 to 2141

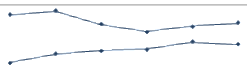
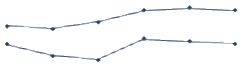
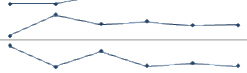
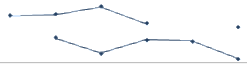
Line #	Course	Course Title	Average Degree	Degree: Terms 2091 to 2141	Average Enrollment	Average EDWC
1	ARTSC_PSY_0010	INTRODUCTION TO PSYCHOLOGY	740.7		1,860	0.399
2	ARTSC_ECON_0100	INTRO MICROECONOMIC THEORY	626.2		1,037	0.604
3	ARTSC_COMMRC_0520	PUBLIC SPEAKING	612.3		515	1.194
4	ARTSC_ANTH_0780	INTRO TO CULTURAL ANTHROPOLOGY	554.8		603	0.925
5	ARTSC_PSY_0310	DEVELOPMENTAL PSYCHOLOGY	535.8		471	1.139
6	ARTSC_CHEM_0310	ORGANIC CHEMISTRY 1	529.7		868	0.613
7	ARTSC_STAT_0200	BASIC APPLIED STATISTICS	529.0		492	1.077
8	ARTSC_ECON_0110	INTRO MACROECONOMIC THEORY	526.7		540	0.979
9	ARTSC_PHYS_0110	INTRODUCTION TO PHYSICS 1	513.5		1,089	0.474
10	ARTSC_ENGFLM_0400	INTRODUCTION TO FILM	495.8		293	1.695
11	ARTSC_PSY_0105	INTRODUCTION TO SOCIAL PSYCH	495.7		333	1.497
12	ARTSC_BIOSC_0150	FOUNDATIONS OF BIOLOGY 1	488.8		1,304	0.404
13	ARTSC_ENGCMP_0200	SEMINAR IN COMPOSITION	481.2		1,132	0.428
14	ARTSC_MATH_0220	ANALYTC GEOMETRY & CALCULUS 1	479.7		1,134	0.425
15	ARTSC_CHEM_0330	ORGANIC CHEMISTRY LABORATORY 1	467.7		644	0.726
16	ARTSC_MUSIC_0711	HISTORY OF JAZZ	459.0		368	1.248
17	ARTSC_STAT_1000	APPLIED STATISTICAL METHODS	454.7		395	1.154
18	ARTSC_CHEM_0110	GENERAL CHEMISTRY 1	444.3		1,348	0.331
19	ARTSC_BIOSC_0050	FOUNDATIONS OF BIOLOGY LAB 1	470.8		1,093	0.431
20	ARTSC_PSY_0160	PSYCHOLOGY OF PERSONALITY	448.2		297	1.517

Table 31 Top 20 Courses with the highest average Enrollment Weighted Degree Centrality (EWDC) in academic terms 2091 to 2141

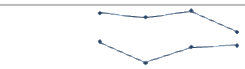
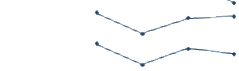
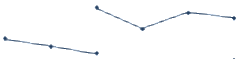
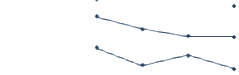
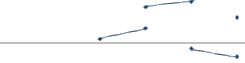
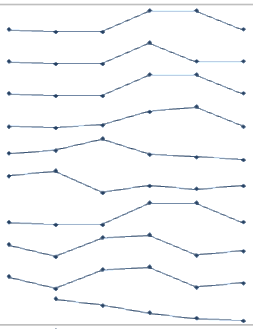
Line #	Course	Course Title	Average EWDC	EWDC : Terms 2091 to 2141	Average Enrollment	Average Degree
1	EDUC_PEDC_0262	YOGA 1	4.312		29	125.0
2	EDUC_PEDC_0225	BUDO	4.301		21	91.5
3	EDUC_PEDC_0226	FITNESS BOXING 1	4.267		21	87.5
4	ARTSC_MUSIC_0614	WOMEN'S CHORALE	4.246		26	110.5
5	EDUC_PEDC_0264	POWER YOGA	4.240		23	96.5
6	EDUC_PEDC_0207	PILATES	4.071		40	163.0
7	EDUC_PEDC_0380	CARDIO PILATES	4.457		20	86.7
8	EDUC_PEDC_0194	SPORTS CONDITIONING	4.369		15	67.0
9	EDUC_PEDC_0266	PILATES FUSION	4.366		20	85.7
10	EDUC_PEDC_0265	YOGA AND PILATES	4.266		20	84.0
11	EDUC_PEDC_0232	TOUCH FOOTBALL 1	4.225		27	112.7
12	ARTSC_MUSIC_0640	JAZZ ENSEMBLE	4.181		18	76.0
13	ARTSC_MILS_1031	BASC LEADER PLN & COMBAT OPRTN	4.144		19	78.7
14	EDUC_PEDC_0209	ON THE BALL	4.085		38	156.3
15	EDUC_PEDC_0154	VARSITY SPORTS 4	4.084		20	80.3
16	EDUC_HPA_1300	NUTRITION IN EXERCISE & SPORT	4.064		21	86.7
17	ARTSC_ARTSC_1999	SENIOR LEADERSHIP SEMINAR	3.955		18	71.0
18	ARTSC_AFRCNA_0639	HISTORY OF JAZZ	3.944		20	77.3
19	EDUC_PEDC_0197	BOOTCAMP FITNESS	4.533		15	68.0
20	ARTSC_PEDC_0194	SPORTS CONDITIONING	4.355		16	69.5

Table 32 Top 20 Courses with the lowest average Enrollment Weighted Degree Centrality (EWDC) in academic terms 2091 to 2141

Line #	Course	Course Title	Average EWDC	EWDC : Terms 2091 to 2141	Average Enrollment	Average Degree
1	DEMED_DENHYG_1116	DENTAL HYGIENE PRACTICUM	0.055		101	5.5
2	DEMED_DENHYG_1117	CHEM, BIOCHEMISTRY & NUTRITION	0.079		66	5.2
3	DEMED_DENHYG_1113	INTRODUCTION TO DENTISTRY	0.082		67	5.5
4	DEMED_DENHYG_1110	BIOLOGICAL SCIENCES 1	0.087		62	5.3
5	SHRS_HIM_1415	INTRO HEALTH INFOR & HLTH CARE	0.128		64	8.2
6	ARTSC_CHEM_0960	GENERAL CHEM FOR ENGINEERS 1	0.156		686	106.7
7	DEMED_DENHYG_1112	INTRODUCTION TO DENTAL HYGIENE	0.164		34	5.5
8	SHRS_EM_1114	MEDICATION ADMINISTRATION	0.164		40	6.3
9	SHRS_EM_1116	PHYSICAL EXAM LAB	0.164		40	6.3
10	NURS_NUR_1282	NUR MGT ACUT/CHRNC HLTH PROBS	0.065		55	3.4
11	NURS_NUR_1281	FOUNDATIONS OF NURSING PRACT 1	0.074		46	3.4
12	SHRS_CDN_1620	MACRONUTRIENT METABOLISM	0.161		65	10.4
13	SHRS_HRS_1027	PATHOPHYSIOLOGY	0.161		64	10.2
14	SHRS_ATHLTR_1831	THERAPEUTIC MODALITIES AND LAB	0.170		31	4.2
15	SHRS_EM_1111	FOUNDATIONS OF EMERGENCY CARE	0.181		36	6.4
16	SHRS_EM_1115	INTRO TO PHYSICAL ASSESSMENT	0.181		36	6.4
17	SHRS_EM_1112	PATHOPHYSIOLOGY	0.182		36	6.4
18	SHRS_CDN_1609	CLINICAL BIOCHEMISTRY	0.186		58	10.8
19	SHRS_ATHLTR_1824	ATHLETIC TRAINING PRACTICUM 1	0.203		21	4.2
20	SHRS_ATHLTR_1821	INJURY EVAL AND TREATMENT 1	0.210		21	4.4

5.7.3.1 Example for Highest EWDC

From the list of courses in Table 31, it appears that high EWDC are associated with courses of general interest. Although these courses do not necessarily have high enrollments they have a high degree of connectivity relative to their enrollments as students from different programs enroll in them. As an example, the Cypher query below returns 40 nodes and 39 relationships including all the cliques and communities where the course Yoga 1 “EDUC_PEDC_0262” is present in 2141. The resulting seven communities are shown in Figure 28 below. The variety in the composition of the clique communities confirms the assessment expressed in the previous paragraph (i.e. with only 29 students enrolled in the course, it is linked to 126 other courses across numerous schools and departments).

```
MATCH (a)-[c:`MULTI_PERCOLATE`]-(b)
WHERE a.ARCHIVE_PERIOD = 2141
 AND a.ITEMSET =~ '.*EDUC_PEDC_0262.*'
 AND b.ITEMSET =~ '.*EDUC_PEDC_0262.*'
 AND c.OVERLAP > 1
RETURN a,b
```

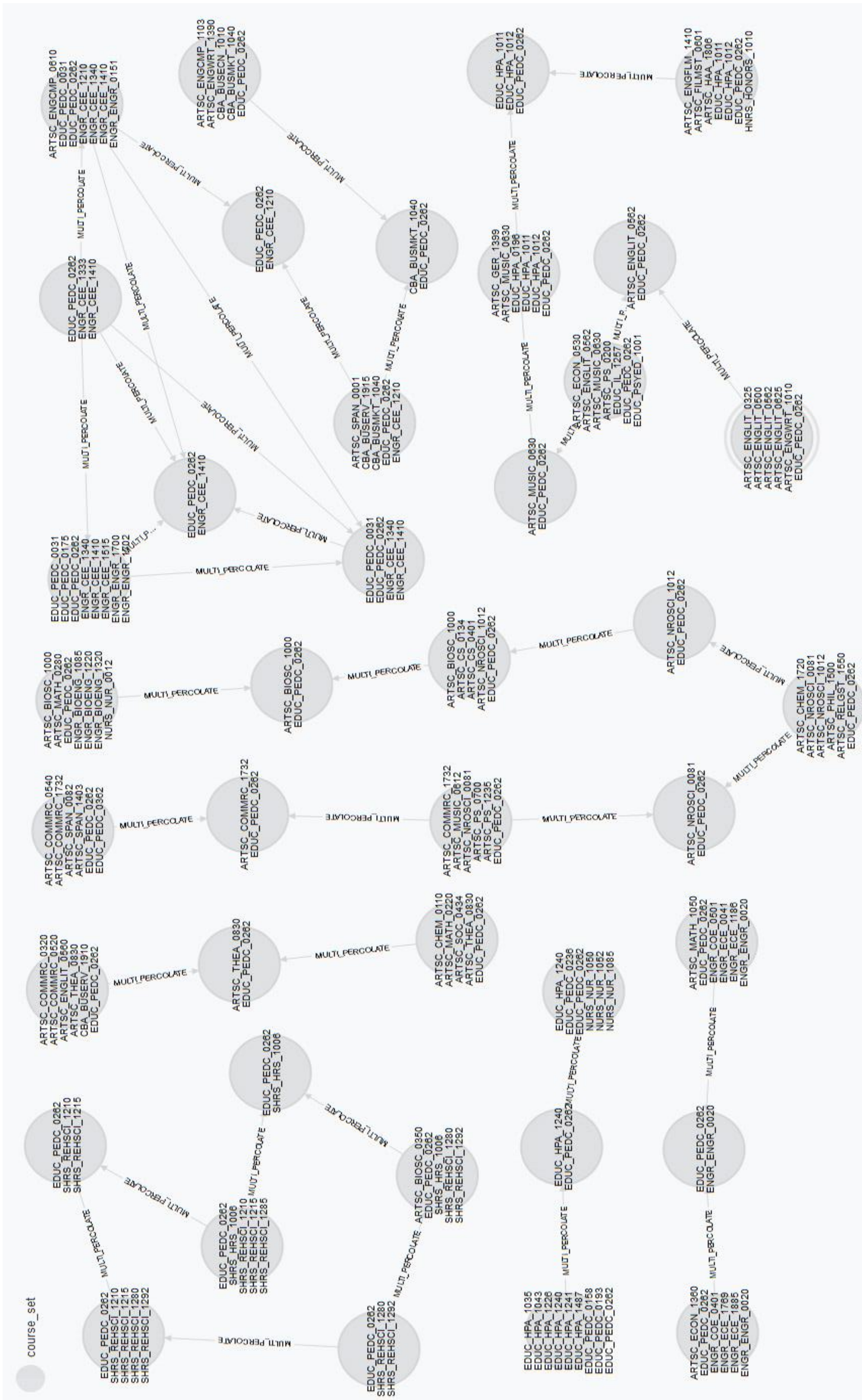


Figure 28 Example of highest EWDC: Communities including course Co-Ed Physical Education "EDUC_PEDC_0262" in all cliques in term 2141

5.7.3.2 Example for Lowest EWDC

From the list of courses in table 32, it appears that low EWDC scores correspond to courses in programs where students follow a very specific program or curriculum. Thus, it is expected that those courses would not have a high degree per student. As an illustration, the cypher query below renders the graph shown in Figure 29 for the course Dental Hygiene Practicum "DEMED_DENHYG_1116", which has the lowest average EWDC. It is linked to only five other courses, all of them in the Dental Hygiene program. The six courses form a clique that is not linked to any other and thus form a standalone community.

```
MATCH (a:`section`)-[:`BELONGS_TO_COURSE`]->(b:`course`)-[:`BELONGS_TO_SET`]->(c)
WHERE c.ARCHIVE_PERIOD = 2141
  AND c.ITEMSET =~ '.*DEMED_DENHYG_1116.*'
RETURN a, b, c
```

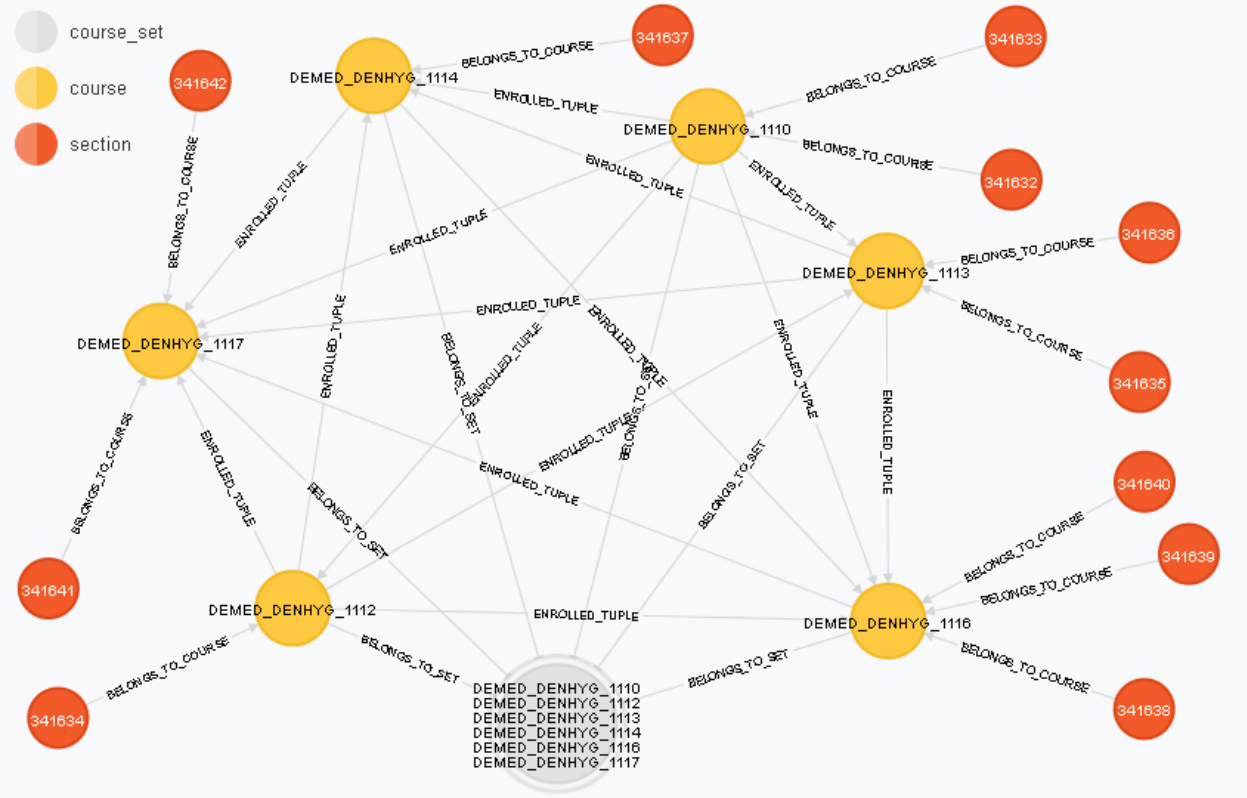


Figure 29 Example of lowest EWDC: Complete graph including course Dental Hygiene Practicum "DEMED_DENHYG_1116" for archive period 2141

5.7.4 Clique Overlap and Weighted Overlap

This section presents examples on the clique graph edge metrics *overlap* and *weighted overlap* discussed in Section 5.4. Appendix E presents statistics on those metrics.

5.7.4.1 Maximum Weighted Overlap Example

The Cypher query below renders the five communities of cliques with the highest weighted overlap in academic term 2141, and the results are shown in Figure 30. A high weighted overlap identifies communities that could be described as compact as they include groups of courses that students take together within a program and that not necessarily have the highest number of enrolled students. The highest enrollment in the cliques in the five communities is 36.

```
 MATCH (a) -[c:`MULTI_PERCOLATE`]-(b)
 WHERE a.ARCHIVE_PERIOD = 2141
RETURN a,b order by c.WEIGHTED_OVERLAP desc limit 20
```

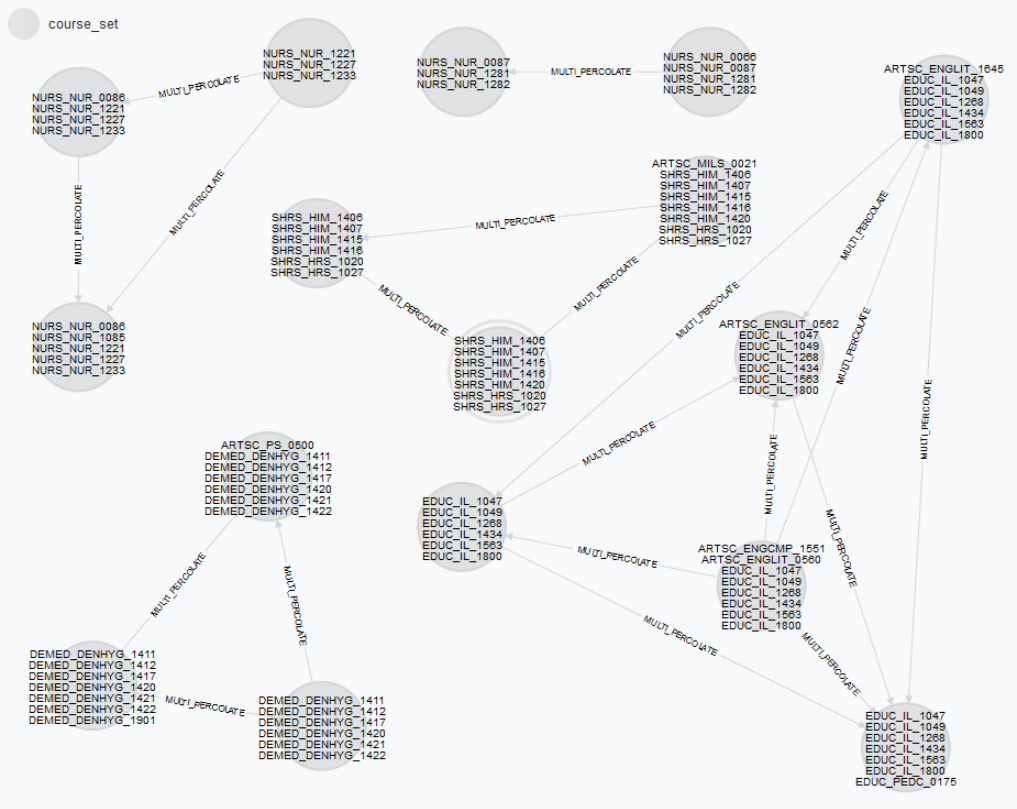


Figure 30 Top five communities of course cliques linked with the highest weighted overlap – period 2141

5.7.4.2 Minimum Weighted Overlap Example

The query below returns the count of cliques linked with a multi-percolate edge with minimum weighted overlap (0.00028) in archive period 2141. The count of those cliques is 87,153.

```
MATCH (a) -[c:`MULTI_PERCOLATE`]-> (b)
 WHERE a.ARCHIVE_PERIOD = 2141
 WITH min(c.WEIGHTED_OVERLAP) as MINWO
 //-------------
 MATCH (a) -[c:`MULTI_PERCOLATE`]-> (b)
 WHERE a.ARCHIVE_PERIOD = 2141
  AND c.WEIGHTED_OVERLAP = MINWO
RETURN count(a)
```

In order to illustrate the results, the query below renders a subset of 20 the cliques with the highest enrollment, and that are linked through a MULTI-PERCOLATE edge with the minimum weighted overlap of 0.00028 in archive period 2141.

```
  MATCH (a) -[c:`MULTI_PERCOLATE`]- (b)
  WHERE a.ARCHIVE_PERIOD = 2141
   AND c.WEIGHTED_OVERLAP= 0.00028
 RETURN a,b
ORDER BY a.ENROLLED desc, b.ENROLLED desc
LIMIT 20
```

Figure 31 below shows the resulting community of cliques at the level specified in the query. All returned cliques have a size two and include the course Introduction to Psychology "ARTSC_PSY_0010". This course has the highest enrollment and the highest degree. All the cliques include other courses that have high enrollment and high degree. Thus, as those courses are present in many cliques the weighted overlap metric is low. A low weighted overlap score helps to identify communities that include highly interconnected cliques with courses from different departments and schools. That is, it helps find cliques with global connections in the network as opposed to the local connections favored by the overlap metric.

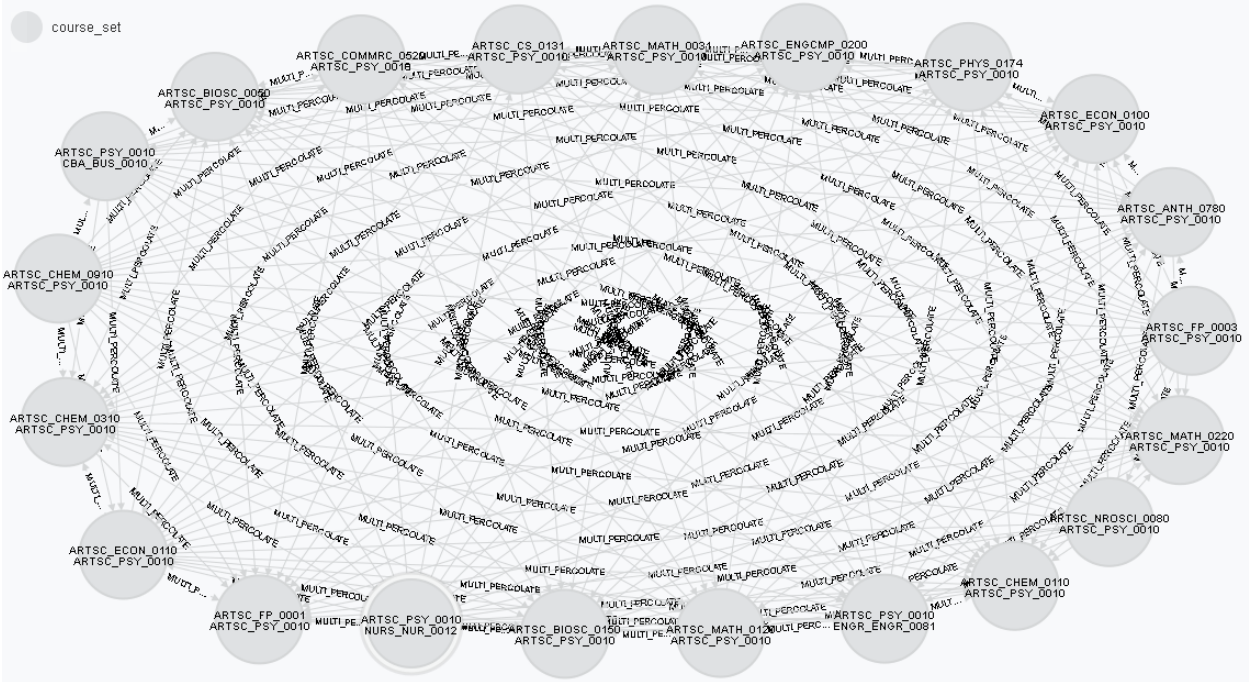


Figure 31 Sub-Community of Top 20 course cliques by enrollment, linked with Minimum Weighted Overlap in academic term 2141

5.8 STAGE II ARCHITECTURAL DIAGRAM

Figure 32 below shows the Stage II Architectural Diagram including the components discussed in previous sections. Data is extracted from the relational database, transformed, and loaded into the graph database. The latter, in conjunction with the discussed methodology enables the identification of courses, course itemsets (cliques) and communities of interest.

It could be correctly argued that the architecture can operate using only the graph database obviating the need of the relational database. The presented architecture uses the relational database mainly because it provides flexibility in terms of making the architecture and a potential implementation easily compatible with existing systems in higher education and with currently available timetabling systems (i.e. they operate on relational databases).

The Stage II Components add a Data Preparation process, the Graph Database used to implement the multi-mode graph and the Communities Identification Process (i.e. GCPM).

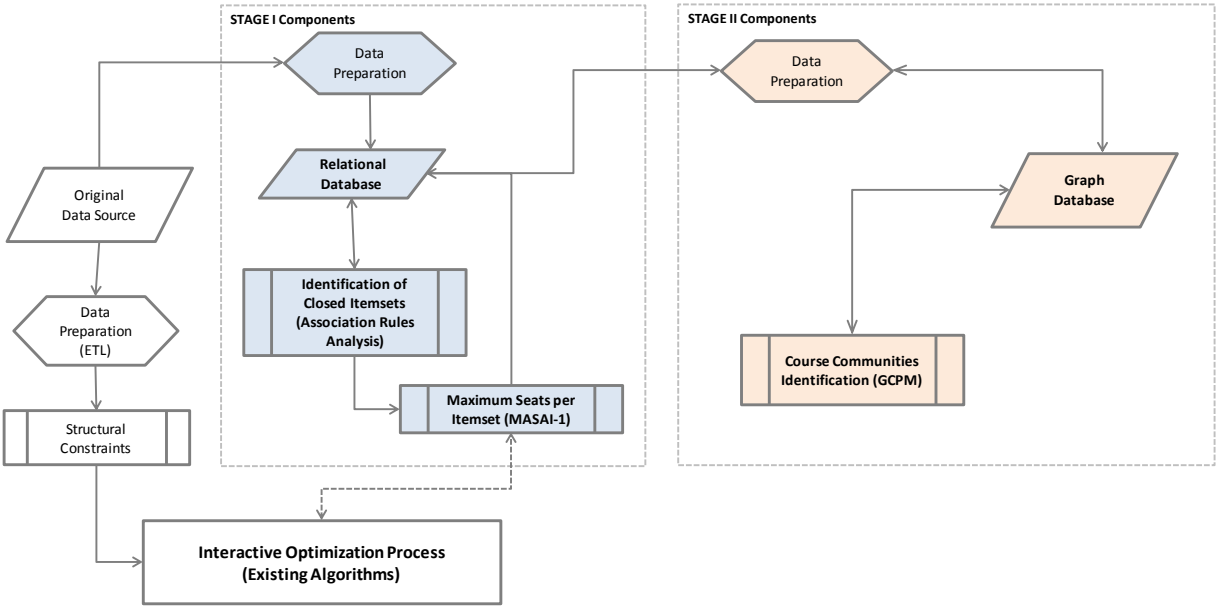


Figure 32: Stage II Architectural Diagram

6.0 SAMPLE CASE ANALYSES COMBINING RESULTS FROM ASSOCIATION RULE ANALYSIS, MASAI AND GCPM

This section presents sample case analyses that integrate the results of the architectural elements and methodology discussed so far. The examples deal with the identification of course itemsets where it would not be possible for students to enroll in all the courses in an itemset, despite seats being available in all the courses in the itemset. After those course itemsets are identified the findings enable the discovery of scheduling practices that limit enrollment options for the cases under consideration.

The Cypher query below returns the total count of itemsets of interest in this case in academic term 2141. The query traverses the graph as follows: Line 1 matches courses with the course itemsets to which they belong. Line 2 limits the results to academic term 2141. Line 3 uses results from MASAI to limit the results to only course itemsets with no seats left available at the end of the enrollment period. Line 4 uses results from the Association Rule Analysis and MASAI to limit the results to only course itemsets with seats available in all courses in the set. That is, the number of courses in the itemset is equal to the number of courses with seats left available. Line 5 returns the count of course itemsets. The total count is 744 course itemsets (cliques).

```
1   MATCH  (a:`course`) -[b:`BELONGS_TO_SET`]-> (c:`course_set`)
2   WHERE  a.ARCHIVE_PERIOD = 2141
3    AND  c.MAX_SEATS_LEFT = 0
4    AND  c.CRS_WITH_SEATS_ITEMSET = c.NBR_COURSES_ITEMSET
5  RETURN  count(distinct c)
```

In order to limit the result set to a tractable size for an example, the Cypher query below renders the clique communities that in Archive Period = 2141 had more than five students enrolled in all cliques; did not have any seats left available in any of the cliques by the end of the enrollment period; and that had seats available in all the individual courses in the cliques. The itemset enrollment threshold of more than five students filters out about 90% of the course

itemsets with no seats left at the end of the enrollment period (Refer to figures in Table 14). As the constraints are imposed in both cliques that multi-percolate, the results are limited to a small group of cliques.

```
 1  MATCH (a:`course_set`) -[b:`MULTI_PERCOLATE`]- (c:`course_set`)
 2  WHERE a.ARCHIVE_PERIOD = 2141
 3   AND a.MAX_SEATS_LEFT = 0
 4   AND c.MAX_SEATS_LEFT = 0
 5   AND c.ENROLLED > 5
 6   AND a.ENROLLED > 5
 7   AND c.CRS_WITH_SEATS_ITEMSET = c.NBR_COURSES_ITEMSET
 8   AND a.CRS_WITH_SEATS_ITEMSET = a.NBR_COURSES_ITEMSET
 9  RETURN a,b,c
10 ORDER BY a.ENROLLED desc, c.ENROLLED desc
```

Line 1 in the query above gets all the course cliques ("course_set") that are linked through a multi-percolate edge. Lines 2 limits the results to archive period 2141. Lines 3 and 4 limit the results to course itemsets (cliques) with no seats left available and the end of the enrollment period as computed using MASAI. Lines 5 and 6 use results from the Association Rule Analysis to limit the results to itemsets with more than five students enrolled. Lines 7 and 8 use combined results from association rules and MASAI to limit the results to itemsets that have seats available in all courses in the set. Line 9 sets the return objects. Line 10 orders the result set by descending itemset enrollment.

The resulting set includes fifty nodes in five communities of cliques (course itemsets) with more than five students enrolled per clique. Every clique has seats available in all the individual courses that compose it. However, it would not possible for more students to enroll in all the courses in the clique. Following is an analysis of two of those communities, which provide insights on counterproductive scheduling practices (i.e. Case 1 and Case 2).

6.1 CASE 1: COURSES WITH MULTIPLE SECTIONS OFFERED AT THE SAME SCHEDULE

Figure 33 below shows one of the communities of cliques identified in the resulting set. It includes seven cliques where the most central (i.e. it has the highest Betweeness Centrality in the set of nodes that form the community) is the 2-clique that has the courses offered by the Dietrich School of Arts and Sciences, Analytic Geometry and Calculus 2 ("ARTSC_MATH_0230") and Basic Physics for Science and Engineering 2 ("ARTSC_PHYS_0175"). The two courses are present in all the three cliques with more than three courses. They are also individually present in all the four 2-clique.

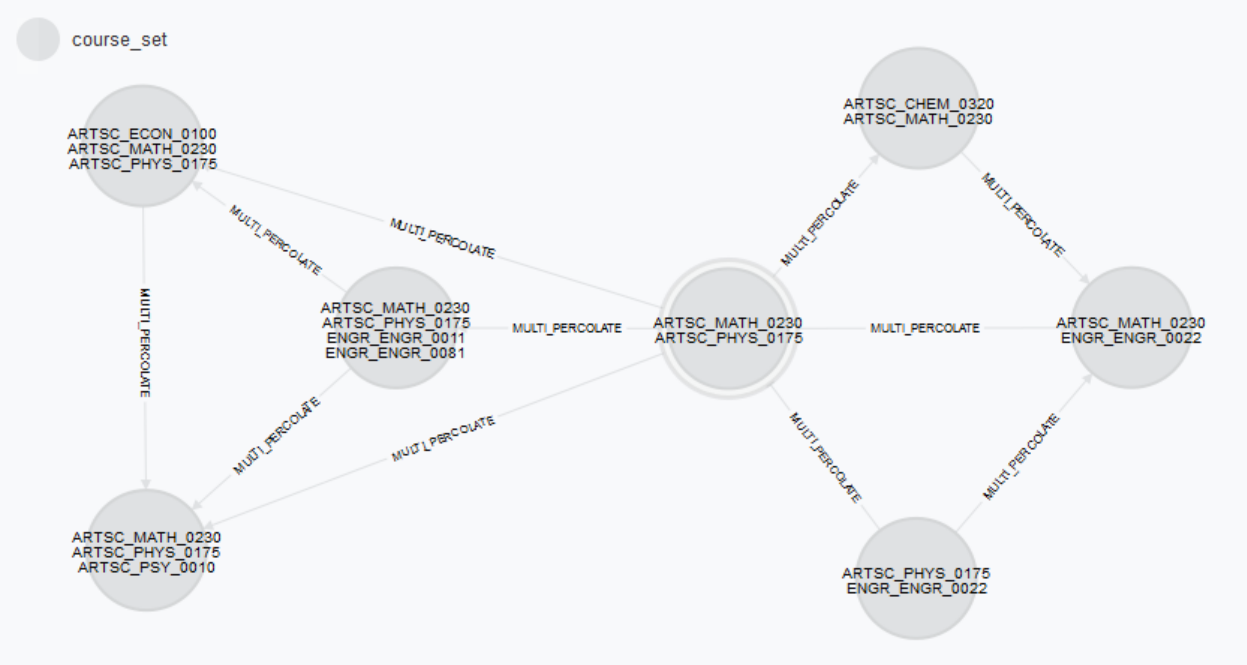


Figure 33 Community of cliques with no seats are left available while there is capacity in all courses in clique (ARTSC_MATH_0230 ARTSC_PHYS_ 0175)

Table 33 below shows detail information on the schedules of all the sections of the courses shown in cliques in Figure 33. The information in the table makes it immediately apparent that the schedule bottleneck in this community is generated by multiple sections of the courses ARTSC_MATH_0230 and ARTSC_PHYS_ 0175 being offered at the same times. In the case of ARTSC_MATH_0230, the three sections offered Monday, Wednesday and Friday from 9:00 a.m. to 9:50 a.m. have a total of 19 seats available, while sections offered at other non-conflicting times are full. All sections of ARTSC_PHYS_ 0175 have seats available.

The practice of offering multiple sections of the same course at the same schedules reduces the enrollment options for students. In particular, if the sections are offered at prime time. In the case at hand, it would not possible to take any combination of courses that include ARTSC_MATH_0230 and ARTSC_PHYS_ 0175 even though there are still seats available in both courses. For the same reason, it would not possible to take ARTSC_CHEM_0320 and ARTSC_MATH_0230, or ARTSC_PHYS_ 0175 and ENGR_ENGR_022, or ARTSC_MATH_0230 and ENGR_ENGR_022.

Table 33 Section schedules for courses included in clique community shown in Figure 33 - academic term 2141

Course	Course Title	Days	Start Time	End Time	Enrollment Total	Enrollment Capacity	Seats Left	
ARTSC_CHEM_0320	ORGANIC CHEMISTRY 2	M W F	9:00	9:50	202	240	38	38
ARTSC_ECON_0100	INTRO MICROECONOMIC THEORY				20	20		18
		W	18:00	20:30	30	30		
		T H	11:00	11:50	255	260	5	
		T H	13:00	13:50	255	260	5	
		M W	12:00	12:50	257	260	3	
		M W	13:00	13:50	258	260	2	
		M W	15:00	16:15	37	40	3	
ENGR_ENGR_0011	INTRO TO ENGINEERING ANALYSIS	T H	10:00	11:50	84	84		39
		T H	14:00	15:50	83	84	1	
		T H	14:00	15:50	83	84	1	
		T H	16:00	17:50	84	84		
		T H	16:00	17:50	83	84	1	
		T H	18:00	19:50	48	84	36	
ENGR_ENGR_0022	MATERLS STRUCTURE & PROPERTIES	M W F	9:00	9:50	71	84	13	13
		M W F	10:00	10:50	138	105	(33)	
ENGR_ENGR_0081	FRESHMAN ENGINEERING SEMINAR 1	T	12:00	12:50	261	285	24	30
		T	13:00	13:50	279	285	6	
ARTSC_MATH_0230	ANALYTC GEOMETRY & CALCULUS 2	M W F	9:00	9:50	69	75	6	19
		M W F	9:00	9:50	72	75	3	
		M W F	9:00	9:50	40	50	10	
		M W F	11:00	11:50	75	75		
		M W F	13:00	13:50	75	75		
		M W F	14:00	14:50	72	72		
		M W	18:00	19:10	75	75		
ARTSC_PHYS_0175	BASC PHYS SCI & ENGR 2 (INTGD)	W	8:00	9:50	164	172	8	32
		W	8:00	9:50	164	172	8	
		F	9:00	9:50	164	172	8	
		M F	9:00	9:50	164	172	8	
ARTSC_PSY_0010	INTRODUCTION TO PSYCHOLOGY	T H	14:30	15:45	398	400	2	342
		T H	16:00	17:15	372	400	28	
		M	18:00	20:30	87	90	3	
		M W	15:00	16:15	279	400	121	
		M W F	9:00	9:50	238	400	162	
		M W F	11:00	11:50	374	400	26	

Although the schedules in the course ENGR_ENGR_0011 are not in direct conflict with the section schedules of the other two courses, the offering of multiple sections of the same course at the same schedules limit the enrollment options for students. If not in this community, perhaps in other communities where ENGR_ENGR_0011 is present.

There are six sections of Introduction to Psychology (ARTSC_PSY_0010) with a total of 2,090 seats. There are also 342 seats left available between the six sections offered, which is

almost equivalent to the capacity of one of them. Given that term after term, this course has the highest enrollments in the whole data set and is the most connected in the course graph, the figures indicate that perhaps there is room for improvement in the schedules of ARTSC_PSY_0010 sections. Although further analysis would be required, and idea to consider would be to limit the offering of courses with large sections at prime time with the goal of minimizing the schedule conflicts with smaller sections.

6.2 CASE 2: SUB-UTILIZATION OF AVAILABLE WEEKLY TIME SLOTS

Figure 34 below shows the largest of the five communities obtained with the second Cypher query discussed at the beginning of Section 6. Every clique in this community has five or more students enrolled and has seats available in sections of all the individual courses that compose it. However, it would not possible for more students to enroll in all the courses in any of the cliques due to capacity limits or schedule conflicts.

The community includes cliques with courses in eight out of the 12 academic subjects offered by the College of Business Administration (CBA) and courses offered by seven Dietrich School departments (ARTSC). In this case, there are two overlapping communities. At the top of the graph, there is a small community of four cliques, with the clique conformed by courses Consumer Behavior "CBA _BUSMKT_1441" and Quantitative Methods "CBA_BUSQOM_0050" being the most central. Then, there is a large cluster with the most central clique including the two courses Quantitative Methods "CBA_BUSQOM_0050" and Financial Accounting "CBA_BUSACC_0030". These two courses are present in all other cliques in the cluster, which has a hierarchical structure[7].

[7] Given that in a community paths exist between all nodes, then Betweeness Centrality can be used to identify nodes of interest. To that end, in NEO4J the collection of courses present in the cliques in a community could be dynamically linked with a community edge, and then Betweeness Centrality for courses in the community can be computed using the Cypher query proposed by Van Bruggen (Van Bruggen, 2014).

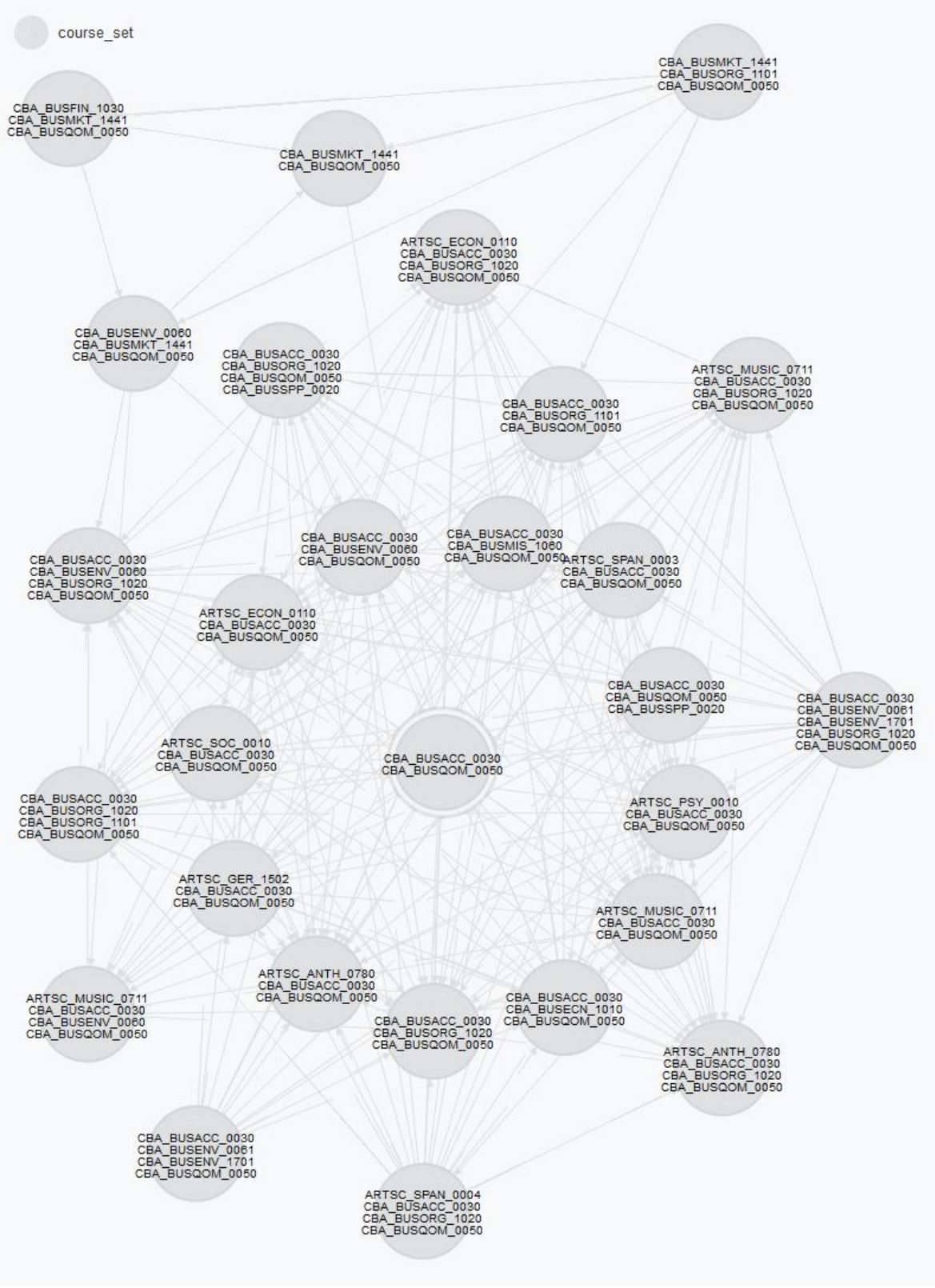


Figure 34 Community of cliques with no seats are left available while there is capacity in all courses in clique (CBA_BUSQOM_0050 CBA_BUSACC_0030 CBA _BUSMKT_1441)

Table 34 below shows detail scheduling information for the three courses in the two most central cliques as discussed above. A couple of observations are that while there are seats available in sections of the three courses between 6:30 p.m. and 9:00 p.m. sections at other times are oversubscribed, and that here are no sections offered with meeting on Fridays. There appears to be a capacity limit being reached in the courses Consumer Behavior and Financial Accounting that is combined with a sub-optimal schedule for Quantitative Methods. If additional sections were to be offered, perhaps it would be opportune to consider sections of these courses on Fridays at non-conflicting times.

Table 34 Section schedules for courses CBA_BUSACC_0030, CBA_BUSMKT_1441 and CBA_BUSQOM_0050 in academic term 2141

Course	Course Title	Days	Start Time	Stop Time	Enrollment Total	Enrollment Cap	Seats Left
CBA_BUSACC_0030	FINANCIAL ACCOUNTING	T H	14:00	15:15	134	130	(4)
CBA_BUSACC_0030	FINANCIAL ACCOUNTING	T H	17:00	18:15	131	130	(1)
CBA_BUSACC_0030	FINANCIAL ACCOUNTING	M	18:30	21:00	44	45	1
CBA_BUSACC_0030	FINANCIAL ACCOUNTING	T H	12:30	13:45	131	130	(1)
CBA_BUSMKT_1441	CONSUMER BEHAVIOR	M	18:30	21:00	41	42	1
CBA_BUSMKT_1441	CONSUMER BEHAVIOR	M W	14:00	15:15	42	42	0
CBA_BUSMKT_1441	CONSUMER BEHAVIOR	M W	15:30	16:45	42	42	0
CBA_BUSQOM_0050	QUANTITATIVE METHODS	M	18:30	21:00	38	55	17
CBA_BUSQOM_0050	QUANTITATIVE METHODS	M W	11:00	12:15	135	135	0
CBA_BUSQOM_0050	QUANTITATIVE METHODS	M W	12:30	13:45	137	135	(2)
CBA_BUSQOM_0050	QUANTITATIVE METHODS	M W	17:00	18:15	136	135	(1)

It is of interest to notice that three courses that appear to be central to the offerings of the College of Business Administration, and that in the current schedule are limiting the enrollment options for students, do not have sections with class meetings on Fridays. Following up on that observation, Table 35 below shows the percentage of sections with meeting on Fridays for all academic units at Pitt's Pittsburgh campus for the six terms under study. The figures suggest that it might be worth exploring in further detail this aspect of the scheduling practices at the

institutional level. Obviously, not offering classes on Fridays mean that 20% of the possible scheduling timeslots are not being used.

Table 35 Percentage of undergraduate sections with class meetings on Friday - academic terms 2091 to 2141

School		All sections Terms 2091 to 2141	Sections with Friday Meeting	% Friday Sections / Total
Dietrich School of Arts and Sciences	ARTSC	11,930	2,669	22.37%
Swanson School of Engineering	ENGR	1,078	172	15.96%
College of Business Administration	CBA	1,007	8	0.79%
School of Education	EDUC	904	11	1.22%
School of Health and Rehabiliation Sciences	SHRS	638	94	14.73%
College of General Studies	CGS	424	12	2.83%
School of Nursing	NURS	255	13	5.10%
School of Social Work	SOCWK	129	3	2.33%
School of Dental Medicine	DEMED	125	23	18.40%
School of Information Sciences	SIS	113	3	2.65%
	Total	**16,603**	**3,008**	**18.12%**

6.3 CASE 3: A PERSPECTIVE FROM AN INDIVIDUAL DEPARTMENT

From the discussion in previous sections, it is clear that enrollments in courses offered by an academic unit (e.g. school or department) are affected by offerings in other units. Now, taking the perspective of a single department, the Cypher query below illustrates an approach to find the cliques of interest that correspond to the cases discussed in sections 6.1 and 6.2

In the Cypher query below, line 1 matches all cliques (course sets). Line 2 filters the results to only academic term 2141. Line 3 includes only cliques with no seats left at the end of the enrollment period. Line 4 includes only cliques where the number of courses in the clique is equal to the number of courses with seats available. Line 5 includes only cliques with courses offered by the academic unit of interest; in this case the department of Economics at the Dietrich

School of Arts and Sciences. Line 6 returns the itemset description and the number of enrolled students in the set. Line 7 orders the result set.

```
1   MATCH (a:`course_set`)
2   WHERE a.ARCHIVE_PERIOD = 2141
3    AND a.MAX_SEATS_LEFT = 0
4    AND a.CRS_WITH_SEATS_ITEMSET = a.NBR_COURSES_ITEMSET
5    AND a.ITEMSET =~ '.*ARTSC_ECON.*'
6  RETURN a.ITEMSET, a.ENROLLED
7 ORDER BY a.ENROLLED desc
```

The query returns 61 cliques. Table 36 below lists the top 10 cliques by enrollment. It appears that students who enroll in the course Introduction to Macro Economic Theory (ARTSC_ECON_0110) are finding limitations to enroll in the other listed courses offered by the College of Business Administration and a few Arts and Sciences departments. Salient examples are the cases previously discussed.

Table 36 Top 10 Cliques of Interest Including Economics courses in term 2141

Course Itemset (Clique)	Enrollment
ARTSC_ECON_0110 CBA_BUSACC_0030 CBA_BUSQOM_0050	16
ARTSC_ECON_0110 CBA_BUSACC_0030 CBA_BUSORG_1020 CBA_BUSQOM_0050	7
ARTSC_ECON_0100 ARTSC_MATH_0230 ARTSC_PHYS_0175	6
ARTSC_ECON_0100 CBA_BUSACC_0030 CBA_BUSORG_1020 CBA_BUSQOM_0050	4
ARTSC_ECON_0110 CBA_BUSACC_0030 CBA_BUSQOM_0050 CBA_BUSSPP_0020	3
ARTSC_ECON_0110 CBA_BUSACC_0030 CBA_BUSENV_0060 CBA_BUSQOM_0050	3
ARTSC_CHEM_0100 ARTSC_ECON_0900	3
ARTSC_ECON_0400 ARTSC_SOC_0010	2
ARTSC_ECON_0400 CBA_BUSACC_0030 CBA_BUSQOM_0050	2
ARTSC_ECON_0110 ARTSC_SPAN_0003 CBA_BUSACC_0030 CBA_BUSQOM_0050	2

One can envision that interactive information and visualizations along the lines of the examples discussed in this section and through the document would be presented to users. This would be done within the framework of a timetabling system that implements a socially translucent environment as discussed in the following section.

7.0 FRAMEWORK FOR A COLLABORATIVE TIMETABLING ENVIRONMENT

Previous sections discussed Stage I and II of an architecture for collaborative timetabling systems. Those sections also presented results based on real enrollment data from six fall academic terms at a large university that demonstrate the viability and value of the proposed approach. This section discusses Stage III, which completes the proposed architecture. The presentation starts with a discussion on Group Decision Support Systems (GDSS) and their application to timetabling. Then, it discusses the Social Translucence approach to the design of collaborative systems and how it would help to cover aspects that are not considered in the GDSS literature. The section closes with an overview of Stage III of the proposed architecture and how it integrates with components from other stages. While the framework for a translucent environment is discussed, the details of an implementation and assessment of results are left for future work.

7.1 DECISION GROUPS TASK STRUCTURES

Steiner (1972) classifies group tasks in additive, disjunctive and conjunctive tasks. *In additive tasks*, each member of the group has similar information and responsibilities, and contributes a part to the group's decision. In that case, group performance is determined by the aggregation of individual effort. *In disjunctive tasks*, each member also has similar information and responsibilities. However, in this case the group selects and optimal solution from the array of solutions presented by individual group members. *In conjunctive tasks,* each group member has unique information. That is, no one has all required information for an optimal decision and such a decision can only be achieved when all the group members maximize their efforts. An optimal decision cannot be achieved if at least one member of the group fails to contribute to the decision task (Steiner, 1972).

Research shows that there is a systematic relationship between patterns of communication and decision quality in a GDSS environment that significantly improve decision quality in disjunctive and conjunctive tasks (Lam, 1997). Higher education timetabling can be

classified as a conjunctive task as administrators at an academic unit do not have complete information or authority to produce an optimal schedule of classes for the whole institution. Thus, in principle the achievement of this goal would be facilitated through a collaborative effort. However, a critical aspect that GDSS research does not consider and that is of crucial importance in the case at hand is the decentralized nature of higher education institutions, where a "territorial" organizational culture is common and rigid views exist on when and where teaching should take place (McCollum, 1998). Thus, it is not likely that a collaborative timetabling effort would naturally develop without an environment that overcomes the characteristics of higher education institutions that might prevent it.

7.2 SOCIAL TRANSLUCENCE APPROACH TO THE DESIGN OF A COLLABORATIVE TIMETABLING SYSTEM

Social translucence is an approach to systems design that support social processes. Socially translucent systems enable people to extend their social experiences and expertise in support of their interactions with others in such a system. The initial proponents of these ideas suggested that socially translucent systems have three main characteristics: visibility, awareness, and accountability, which are further discussed ahead. (Erickson, Halverson, Kellogg, Laff, & Wolf, 2002; Erickson & Kellogg, 2000; Erickson et al., 1999).

Proposals for systems that implement social translucence have done so through visualizations, many of them built on top of existing systems (Begole, Tang, Smith, & Yankelovich, 2002; Díaz & Puente, 2010; Erickson et al., 2006; Fogarty, Lai, & Christensen, 2004; Szostek, Karapanos, Eggen, & Holenderski, 2008; Wattenberg, Viégas, & Hollenbach, 2007). It is recently that work has started to appear that proposes that social translucence should be part of the architectural design of systems as opposed to components added after the systems have been built (Gilbert, 2012; McDonald, Gokhman, & Zachry, 2012). This dissertation takes the latter approach by proposing an architecture and methodology that builds on the natural structure, requirements and goals of the task at hand support the three main characteristics of a socially translucent system. *First*, it enables visibility through the identification of groups of users that would benefit from collaborative work in an environment that provides shared

information on schedules and the specific scheduling cases that are relevant for the group. *Second,* it enables collective awareness by providing users with specific and shared information on the schedule cases that need attention and the scheduling practices that have negative impact on the students and the institution. *Third,* it facilitates accountability by enabling groups of users and/or the institution's leadership to determine plans of action based on the findings about aspects of the schedules that might need to be modified. By advancing work on the referred direction, this work contributes to emerging research on the use of socially translucent designs to support organizational changes and institutional transparency. In the case of this dissertation, the goal is to achieve improvements in scheduling practices in contrast to emerging research on social translucence that focuses on achieving personal behavior changes (Barreto, Szóstek, & Karapanos, 2013; Stuart, Dabbish, Kiesler, Kinnaird, & Kang, 2012).

One can envision an environment where users would have authorization to determine the course offerings and section schedules for one or more academic units for a coming term. Additionally, users would have view access to historical and current course offerings and section schedules at the whole institution as well as daily information on enrollments in sections. If the system is coupled with an interactive timetabling optimization algorithm, users would need to define the list of courses to be offered along with constraints, and then work with the system during the optimization process to create the class schedule. If a timetabling optimization algorithm is not implemented, then users would develop the schedule for the academic units under their responsibility using their usual method. In both cases, users would have access to analyses of historical patterns that have generated schedule bottlenecks derived from their class offerings and the offerings at other academic units. For instance, a user responsible for the schedules in the department of Mathematics would be able to identify all the department's courses and sections with scheduling issues in previous terms and if enrollments are open, receive daily reports on courses and combinations of courses of interests that are reaching capacity limits or that are under enrolled.

Referring to the example discussed in Section 6.1, users responsible for schedules in courses on the subjects of Chemistry, Economics, Engineering, Mathematics, Physics, and Psychology, would be able to see the information in all their courses, and information on the course itemsets, courses and schedules discussed in the section. Users would also know that other users in the referred academic units are seeing the same information as the system would

present that case to all of them and provide the tools for discussion. That information would provide mutual *visibility* of the situation, in the sense that the three users would be informed of the problems that their scheduling is causing for other academic units and students, as well as the problems that the scheduling of other academic units are causing for their own unit offerings and students. The shared information would make them *aware* of the specific problems associated with scheduling multiple sections of the same course at the same days and times, and scheduling very large sections at prime time. Depending on the institution's administration and governance, the *accountability* for the users' actions or inaction to improve the situation would be determined by them or by higher level administrators who would have complete access to the information and receive reports on identified conflicts.

As reported by Erickson and Kellogg on the results of their work, hopefully a subtler reason to improve the class schedules through collaboration in a translucent timetabling environment would be derived from the individual feelings of accountability that make norms, rules and customs effective mechanisms for social control on the presence of visibility and awareness (Erickson & Kellogg, 2000). Whether that is the case in a collaborative timetabling system would need to be assessed through a test implementation.

7.3 STAGE III ARCHITECTURAL DIAGRAM

Figure 35 ahead presents the complete proposed architecture to support collaborative timetabling in higher education. The stage III components that are now added include a translucent environment that would have an end user interface.

A complete collaborative timetabling system would be designed and implemented to be compatible with existing optimization algorithms or systems and with Stage I and II of components. As different institutions have different organizational structures and practices, one can envision that the design of the translucent environment would need to enable different configurations in response to those organizational characteristics. Figure 35 illustrates the main elements that would be required, namely:

- An interface with MASAI that would enable users to run canned or ad-hoc reports on the maximum capacity available per course itemsets as discussed in Section 4.3.
- An interface with the graph database that would enable users to run canned or ad-hoc analyses along the lines discussed in Section 6.
- A process to define initial constraints either individually or in group.
- A process to modify constraints during the timetabling construction.
- Users groups could be formed ad-hoc depending on results of analyses and/or predefined based on preliminary analysis of results.

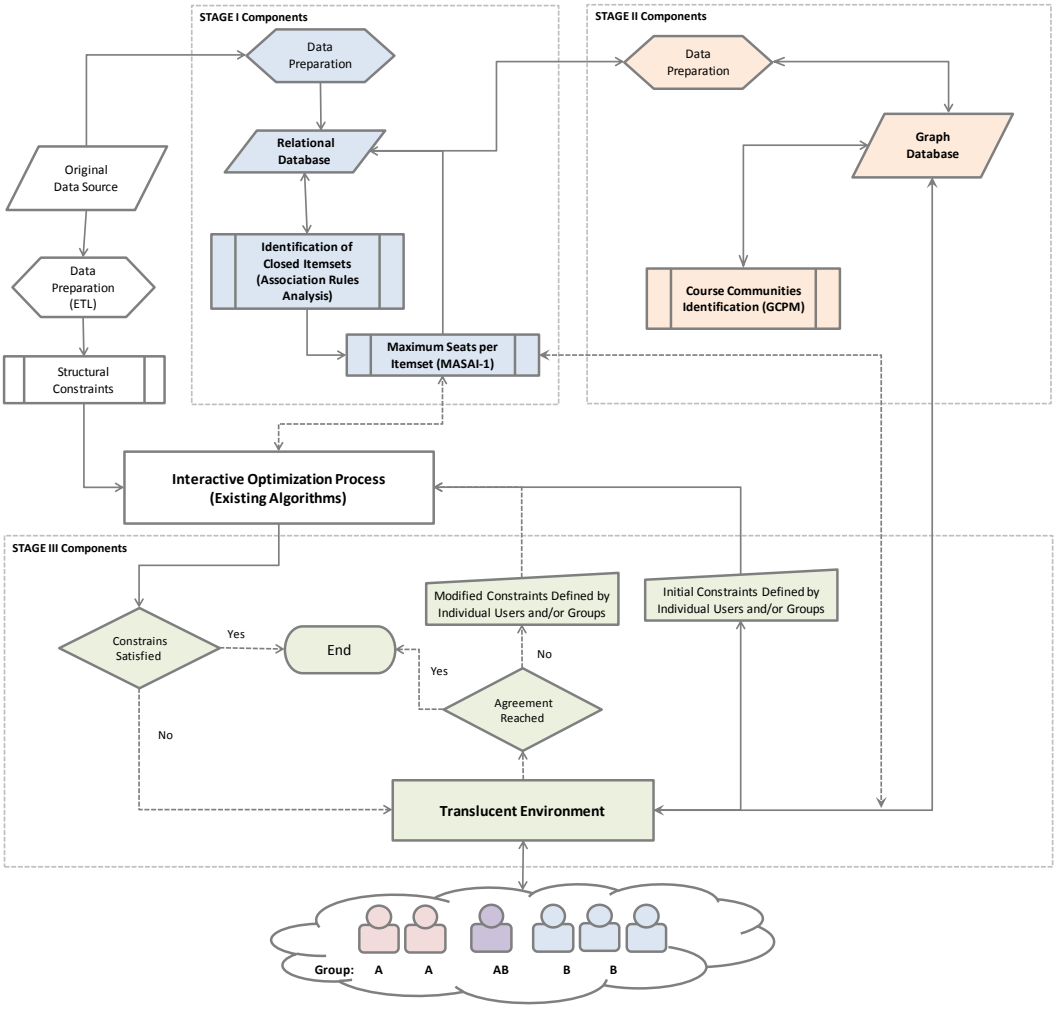


Figure 35 Stage III Complete Proposed Architecture

8.0 CONCLUSIONS AND FUTURE RESEARCH

This dissertation has presented an architecture and methodology for collaborative timetabling in higher education. It expands the scope of the domain by addressing core reported gaps that exist between research on the topic and the needs of higher education institutions. In order to illustrate the presented ideas with specific examples, actual undergraduate enrollment data from six recent fall terms at the University of Pittsburgh's Pittsburgh campus have been used. This section discusses the main contributions and plans for future work.

8.1 CONTRIBUTIONS

First, literature on timetabling in higher education has traditionally focused on the development of optimization algorithms that provide optimal solutions given the requirements and constraints presented by users. In contrast, the proposed architecture and methodology is intended to help higher education administrators to identify non-trivial enrollment patterns and practices that have a negative impact in the schedule of classes. With that knowledge, they would be able to develop better class schedules, and when applicable formulate better informed requirements and constraints to be passed to the optimization algorithms embedded in the aforementioned systems that the timetabling community has developed.

Second, in order to identify all the combinations of courses that students take in a term, the problem has been modeled as an Association Rule Analysis, also known as market basket analysis, where those combinations of courses are treated as transactions. This analysis renders an initial set of combinations of courses of interest called course itemsets. A novel approach has been proposed to identify course itemsets of interest that departs from the traditional Association Rule Analysis by using information that is not available in the transactional data. A backtracking algorithm called MASAI has been proposed to determine the maximum number of seats available per itemset. It considers all the individual sections of courses in an itemset, including information on days and times at which those sections are offered along with maximum capacities and actual enrollments in each section of a course. MASAI uses that information to

determine the maximum number of seats possible in the itemset. Course itemsets where no more enrollments are possible are deemed of primary interest. The complexity of MASAI, and a prototype implementation were discussed along with results using several examples using University of Pittsburgh enrollment data.

Third, a multi-mode graph was modelled and implemented using the NEO4J graph database. The graph incorporates information obtained with the Association Rule Analysis and MASAI. Furthermore, it enables deeper analyses that consider the relationships between course itemsets and facilitates the creation of a clique graph that is instrumental to community identification.

Fourth, A Generalized Clique Percolation Method (GCPM) was proposed to enable the identification of overlapping and hierarchical communities in graphs/networks. A theoretical discussion, results, and practical implementation using Pitt's enrolment data were discussed.

GCPM was used to analyze sample cases enabling the discovery of non-trivial enrollment patterns, and the identification of scheduling bottlenecks that limit the enrollment options for students. The presented sample analyses use minimal domain knowledge and would be likely enriched if a system implemented using the presented methodologies is used by expert higher education administrators.

Fifth, the elements that would form the core of a socially translucent environment that is based on the presented architectural components were discussed. As part of a timetabling system, this environment would provide scheduling authorities with access to shared information on enrollment patterns, and how decisions on scheduling of courses in their departments impact the overall institution's schedule and the enrollment options for students.

8.2 FUTURE WORK

The Generalized Percolation Method (GCPM) has potential applications in multiple disciplines. Thus, there is work to be advanced in an independent track using GCPM on standard data sets that the network analysis community use as benchmarks. That work would be intended to validate the results obtained with the enrollment data using GCPM on data from other domains.

The following step would be the development of an end user interface that enables a translucent environment in support of proof of concept and pilot tests of a collaborative timetabling system involving higher education faculty and administrators. The involvement of domain experts on timetabling in higher education would likely bring additional questions and proposals that build upon the ideas discussed in this document. That information is crucial to the goal of implementing collaborative timetabling systems that can be successfully configured to respond to the needs of people in the field.

This document included numerous results from a prototype implementation of the new architectural components using actual enrollment data. There are however aspects of the new architecture that still need to be empirically tested including, acceptance of the presented approach, and assessment of the impact of the discussed changes on the quality of class schedules.

One can envision that those two aspects can be validated at two levels. First, using sample cases to engage with groups of selected academic units with the goal of performing a few coordinated targeted schedule changes. That effort would enable a first pass at acquiring information on how a representative sample of faculty and administrators at several schools and departments respond to the new collaborative approach. The small scale of that effort would not result in significant risk or financial investments. The implementation of those few targeted changes would enable the assessment of the impact in enrollments and student satisfaction, via analysis of actual enrollments and surveys, in a manageable group of courses. Test cases could include the following:

- Results from the analyzed data set suggest that the offering of multiple sections of the same course at the same time reduce the enrollment options for students. In the case of Pitt, the

cases discussed in section 4.4.1, and section 6.1 would be good candidates to assess the impact of action on that observation by offering all sections of these courses at different times.

- Given that the course network is scale-free and the observed results in the discussed examples, it appears that offering large sections (hubs) at prime-time reduce the enrollment options for students. Moving a section from a large course currently offered at prime time, to a non-prime time would enable to assess the impact of actions on that observation. In the case of Pitt, a good candidate is the section of course Introduction to Psychology (ARTSC_PSY_0010) discussed in section 6.1, and currently offered Monday, Wednesday and Friday from 9:00 to 9:50 a.m.
- The case discussed in section 6.2 suggests that not offering sections with Friday sessions reduces the use of the available Monday-Friday timeslots by 20% to the detriment of the schedule quality. In the case of Pitt, the College of Business Administration is the unit with the lowest percentage of sections with session meetings during Fridays. Offering sections of the courses Quantitative Methods (CBA_BUSQOM_0050) and Financial Accounting (CBA_BUSACC_0030) would serve as good test cases for the referred observation.

Second, the examples and demonstrations provided through the document show that the proposed architectural components can provide valuable insights on areas of improvement when developing course schedules that current timetabling systems are not able to provide. The addition of results and observations from the tests described above would serve for the development of a timetabling system based on the new collaborative architecture. The idea is then to develop a system that can interface with existing optimization algorithms and/or an optimization based systems (e.g. Purdue's UNITIME (UNITIME), Bullet TimeTabler Education (BTTE) (Fernandes et al., 2015)).

All the analyses presented in this document are based on information derived from the set of courses that students enroll in during each term treating them as transactions. An aspect that still needs to be explored regards the sequences of individual courses that students enroll in across terms. For that endeavor, it would be important to use data that was not available for this project including cohorts, grades, majors, degrees, etc. These additional fields would enable the stratification of the data and the exploration of questions like: Given the course enrollments during a term, what are the expected enrollments in specific courses in subsequent terms? Based

on historical enrollments and outcomes, can we make recommendations to current students on which courses to enroll? Is there a significant association between certain combinations and sequences of courses and success or failure in terms of graduating from a major on time?

There are natural extensions to the multi-mode graph that would enable the addition of information and development of analyses that are useful for timetabling and planning in higher education. For instance, nodes could be added to represent individuals (i.e. students, instructors, etc.) academic plans, and classrooms. Then, edges could be added to represent and link individuals with sections they teach or with sections they enroll in, and classrooms with sections assigned to them. Edges could be added between courses to indicate co-requisites and pre-requisites. At the course level, it would be possible to add edges between cross-listed courses. Edges could be added to link students with their academic plans[8].

[8] Although not discussed in this document, nodes representing students and edges linking the student nodes to the classes they enroll have already been created in the multi-mode graph using the scripts listed in Appendix C.

APPENDIX A STUDY ENVIRONMENT – THE UNIVERSITY OF PITTSBURGH, PITTSBURGH CAMPUS

Pitt's Pittsburgh campus is used as study environment to illustrate the ideas proposed in this work with a specific case. This Appendix describes the university and provides details on enrollment characteristics as well as scheduling policies and practices at the Pittsburgh campus.

The University's Institutional Review Board "designated as exempt under section 45 CFR 46.101(b)(4) Existing data, documents, or records" the use of information on scheduling practices, schedules, class enrollment, and facilities for the purpose of this dissertation. Data on undergraduate enrollments during the six fall terms between 2008 (archive period 2091) and 2013 (archive period 2141) compose the analysis set.

Founded in 1787, Pitt is among the oldest universities in the United States. It is a member of the Association of American Universities (AAU), which includes 62 preeminent doctorate-granting research institutions in North America. Currently, Pitt has five campuses in Pittsburgh, Greensburg, Johnstown, Bradford and Titusville, with the Pittsburgh campus being the largest. Pitt has approximately 25,000 undergraduate students and 10,000 graduate students. There are 4,450 Full-Time faculty and 813 part-time faculty. The university also employs approximately 7,000 staff[9].

The Pittsburgh campus includes 17 academic units with approximately 29,000 students. The Dietrich School of Arts and Sciences (A&S), which is the largest academic unit, has approximately 11,000 undergraduate students enrolled in the fall term. A&S has 47 departments and programs across the Humanities, Natural Sciences and Social Sciences[10]. The College of General Studies is also part of the Dietrich School of Arts and Sciences[11]. Many freshmen students come to Pitt through A&S and later on transfer to other schools. Furthermore, A&S provides most of the general education requirements for undergraduate students. Because of that, in any given term, between 60% and 80% of students from other schools enroll in A&S sections.

[9] http://www.pitt.edu/about
[10] http://www.asundergrad.pitt.edu/
[11] http://www.cgs.pitt.edu/

Table 37 below shows information on undergraduate enrollment at Pitt's Pittsburgh campus for five of the six fall terms used as case example for this dissertation[12].

Table 37 Fall terms undergraduate enrollment at Pitt's Pittsburgh Campus (2009 - 2013)

School		2091	2101	2111	2121	2131	2131 %
ARTSC	Dietrich School of Arts and Sciences	10,481	10,818	10,926	11,049	10,921	59.26%
CGS	College of General Studies	1,196	1,183	1,239	1,151	1,053	5.71%
	Sub-Total Arts and Sciences	*11,677*	*12,001*	*12,165*	*12,200*	*11,974*	*64.97%*
ENGR	Swanson School of Engineering	2,014	2,104	2,191	2,323	2,468	13.39%
CBA	College of Business Administration	1,960	2,021	2,075	2,027	2,032	11.03%
NURS	School of Nursing	555	628	673	637	627	3.40%
SHRS	School of Health and Rehabilitation Sciences	441	476	498	508	559	3.03%
EDUC	School of Education	261	272	238	215	221	1.20%
PHARM	School of Pharmacy	214	218	215	215	220	1.19%
SIS	School of Information Sciences	129	135	142	148	157	0.85%
SOCWK	School of Social Work	99	96	99	86	94	0.51%
DEMED	School of Dental Medicine	77	80	75	68	77	0.42%
	Total	**17,427**	**18,031**	**18,371**	**18,427**	**18,429**	**100.00%**

Note on naming convention for academic terms at Pitt:

Fall Terms correspond to September-December of previous calendar year, i.e. 2131 corresponds to Sept-Dec 2012

In the fall term approximately 3,500 undergraduate sections are offered at the Pittsburgh campus when counting only Lectures, Seminars, Practicums and Workshops. A&S offers approximately 74% of those undergraduate sections.

Regarding the distribution of the number of subjects that students register for, Table 38 below shows that in fall terms approximately 80% of undergraduate students register for three or more subjects. Furthermore, approximately 73.5% of undergraduate students register for three to five distinct subjects in the fall terms.

[12] http://www.ir.pitt.edu/factbook/

Table 38 Pitt Pittsburgh: Distribution of number of distinct subjects that undergraduate students enroll in

Including only Lectures, Seminars, Practicums and Workshops

Number of Subjects	Fall Terms				
	2091	2101	2111	2121	2131
1	10.32%	10.07%	9.32%	9.28%	9.18%
2	9.96%	10.30%	10.60%	10.88%	10.95%
3	16.20%	16.53%	17.37%	16.98%	17.01%
4	28.30%	28.75%	28.83%	28.24%	28.32%
5	28.90%	28.43%	27.72%	27.75%	28.03%
6	6.04%	5.66%	5.86%	6.42%	6.16%
7	0.27%	0.25%	0.29%	0.43%	0.35%
8		0.01%	0.01%	0.02%	0.01%
Total	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
3 or more Subjects	79.72%	79.63%	80.07%	79.83%	79.87%

Naming convention for academic terms at Pitt:

Fall Terms correspond to September-December of previous calendar year,
i.e. 2131 corresponds to Sept-Dec 2012

Table 39 below shows the percent distribution of undergraduate seats taken by students across schools throughout the fall term September to December 2012. Figures show that the highest levels of cross registration occur between A&S and other schools with the top three being the School of Information Sciences, School of Engineering and the College Business Administration (The College of General Studies is part of A&S).

Table 39 Pitt Pittsburgh: Distribution of undergraduate seats taken by undergraduate students across schools (Term 2131: Sept – Dec 2012)

Including only Lectures, Seminars, Practicums and Workshops

School Offering the Classes		Percentage of Seats Taken by Student's School										
		ARTSC	ENGR	CBA	CGS	EDUC	SHRS	NURS	SIS	SOCWK	DEMED	COED
ARTSC	Dietrich School Arts and Sciences	90.12%	38.11%	30.78%	57.74%	7.11%	12.25%	20.58%	35.94%	18.81%	8.84%	79.93%
ENGR	Swanson School of Engineering	0.20%	60.16%	0.07%	0.28%		0.03%		0.16%			7.27%
CBA	College of Business Administration	1.03%	0.15%	65.25%	0.77%	0.08%			1.25%			10.03%
CGS	College of General Studies	4.05%	0.43%	1.66%	33.99%	0.97%	1.12%	3.49%	7.66%	3.71%	3.26%	1.73%
EDUC	School of Education	3.11%	0.86%	2.14%	3.38%	91.43%	2.45%	3.99%	1.72%	0.74%	0.47%	0.69%
SHRS	School of Health and Rehab. Sciences	0.75%	0.05%		1.41%	0.16%	83.84%			0.25%		0.35%
NURS	School of Nursing	0.16%	0.04%		1.11%	0.16%	0.19%	71.54%			0.23%	
SIS	School of Information Sciences	0.26%	0.01%	0.05%	0.83%		0.06%		53.28%			
SOCWK	School of Social Work	0.16%			0.25%	0.08%	0.03%	0.04%		76.49%		
DMED	School of Dental Medicine	0.02%			0.25%						87.21%	
COED	Cooperative Education	0.13%	0.20%	0.06%			0.03%	0.35%				
	Total	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**

Although not shown in the tables, there is a high level of enrollment across A&S departments that is mostly driven by the interdisciplinary nature of the school's general education requirements and that approximately 25% of A&S students are registered for two or more majors.[13]

A.1. COURSE SCHEDULING POLICY AT THE UNIVERSITY OF PITTSBURGH

The University has policies that establish the course meeting times for the Pittsburgh campus[14]. Other campuses specify their own policies. The spirit of the policies is to provide guidelines to facilitate the equitable distribution of course meetings, improve classroom utilization and achieve consistency in scheduling. The main guidelines of the policy are:

[13] Pitt A&S General Education Requirements: http://www.asundergrad.pitt.edu/requirements/gened.html
[14] University of Pittsburgh – Course Scheduling Policies and Procedures: http://www.registrar.pitt.edu/course_policies-procedures.html

- Departments are encouraged to schedule classes equitably between 8:00 a.m. and 6:00 p.m.
- All day classes are to start on the hour at 8:00 a.m. and must stop by 5:50 p.m. The exceptions are classes offered on Tuesday and Thursday.
- All classes of 75 minute duration meet on Tuesdays and Thursdays starting at 8:00 a.m., 9:30 a.m., 11:00 a.m., 2: 30 p.m., and 4:00 p.m.
- All 50 minute classes meeting on Tuesdays and Thursdays must start on the hour beginning at 8:00 a.m. and must stop before 5:50 p.m.
- Saturday classes are to be offered at times that are convenient for students and faculty.
- All standard three-credit undergraduate evening classes meeting once a week are to begin at 6:00 p.m.
- Other start times for evening courses that meet more than once a week are 6:30 p.m., 6:45 p.m., and 7:10 p.m.

A.2. COURSE SCHEDULING PRACTICE AT THE UNIVERSITY OF PITTSBURGH

Information presented in this section was obtained from conversations with course scheduling authorities at A&S, the University Registrar's Office and from the author's work experience at Pitt. Course scheduling at the University of Pittsburgh main campus starts between six and seven months in advance of every term. It is done at the level of departments within each school with the exception being the small schools. Scheduling authorities (administrators and/or faculty) at departmental level have the most accurate description of plans for subsequent terms. They work independently of each other starting with the latest schedule for the equivalent term and then adjusting for changes in coming and going faculty, changes in course offerings if any, and changes in facilities availability. Most schedules have minimum changes over time. In fact, departments only have to submit to the Registrar's Office forms including changes for coming terms relative the latest equivalent term using an on-line form.

Departments submit their schedules request to Registrar's Office, which is the central scheduling authority. Every term, departmental scheduling authorities submit requests for change to the schedule that was used in the last equivalent term. The Registrar's Office takes care of

registering the requests in the university system (currently PeopleSoft) and when required assign rooms. The software SCHEDULE25 (CollegeNET) is used to assign rooms to the requested schedule of classes in the cases when the request does not specify a room. The pool of rooms is divided in three groups: Rooms that are assigned to departments, which they can use to schedule their classes, rooms that departments control and can use but that the Registrar's Office can take back if necessary, and rooms that the Registrar's Office manages. SCHEDULE25 has algorithms that can assign rooms matching the requests in the most optimal way.

There are at least two drawbacks to the described process. *First*, from conversations with department administrators, most of their requests already specify rooms, times, capacities and instructors; and are mostly a copy of the previous equivalent term, i.e.: last fall schedule is used to schedule for the next fall. As this approach does not proactively consider the growing total enrollments and the enrollments across sections offered by different departments and schools, it is necessary to continuously make adjustment to the schedules. These adjustments keep going for several days after classes have started. The most common approach in the case of classes filling up is to make requests for increases in the maximum capacity of sections or for additional sections. In the case of low enrolled sections, there are requests for integration of multiple sections or cancellations. As a result, the Registrar's Office is not able to effectively use SCHEDULE25 in reason to the frequent and numerous requests for change. *Second*, schedules that departments request have built-in inefficiencies derived from the way they are constructed, which does not consider that students register for classes across departments and schools. As reported in the literature, one of the main reasons that permit the continued operation under these circumstances is the low utilization of class rooms. For instance, in the analyzed fall terms, the average utilization of class room capacity at the Pitt Pittsburgh campus was 39.7%[15].

The described way of operating ads an unnecessary strain to the university's resources and does not guarantee that the final schedule of classes offered satisfies the needs and preferences of students in the best possible manner. Moreover, it unnecessarily adds to the work load of large numbers of people across campus. The Registrar's Office has made requests for the improvement of course schedules planning and the reduction of the number of requested changes

[15] Figure was computed using the listed capacity of all rooms used for classes at the Pittsburgh campus for 50 hours per week versus the used capacity during the same time.

as this affects their operations and prevents them from using the university facilities in an optimal way. As it is reported in the literature for other institutions, at Pitt the timetabling process involves significant effort to support what in practice has become a room booking exercise that does not consider the overall institutional goals and where the objective is to find a schedule that works until the next cycle starts as opposed to the best possible schedule.

APPENDIX B. GENERAL EDUCATION COURSES THAT CANNOT BE TAKEN TOGETHER

Tables 40 and 41 below show sets of two general education courses that cannot be taken together in a term in any of the six academic terms under study. Table 42 shows the courses in those sets listed individually. The latter table also shows the number of times that each course is present in the sets of two courses. Cross-listings are highlighted in Table 42.

Table 40 Sets of two General Education courses that cannot be taken together due to schedule conflicts for all the six terms under study (Part I)

Course Set	Course	Course Title	Course	Course Title
1	ARTSC_AFRCNA_0639	HISTORY OF JAZZ	ARTSC_ANTH_0620	BIOCULTURAL ANTHROPOLOGY
2	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_ARTSC_0020	LATIN AMERICA AND CARIBBEAN
3	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_ASTRON_0087	BASICS OF SPACE FLIGHT
4	ARTSC_ARTSC_0020	LATIN AMERICA AND CARIBBEAN	ARTSC_ASTRON_0087	BASICS OF SPACE FLIGHT
5	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_ASTRON_0113	INTRODUCTION TO ASTRONOMY
6	ARTSC_ASTRON_0088	STONEHENGE TO HUBBLE	ARTSC_ASTRON_0113	INTRODUCTION TO ASTRONOMY
7	ARTSC_ASTRON_0086	OBSERVATIONAL ASTRONOMY	ARTSC_BIOSC_0150	FOUNDATIONS OF BIOLOGY 1
8	ARTSC_ASTRON_0086	OBSERVATIONAL ASTRONOMY	ARTSC_CHEM_0100	PREPARATION GENERAL CHEMISTRY
9	ARTSC_ASTRON_0086	OBSERVATIONAL ASTRONOMY	ARTSC_CHEM_0110	GENERAL CHEMISTRY 1
10	ARTSC_CHEM_0710	UHC GENERAL CHEMISTRY 1	ARTSC_CHEM_0760	UHC GENERAL CHEM FOR ENGINRS 1
11	ARTSC_ANTH_0582	INTRODUCTION TO ARCHEOLOGY	ARTSC_CHEM_0910	CHEMCL PRINCPL HEALTH PROFESSN
12	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_CHIN_1090	GREAT MINDS OF CHINA
13	ARTSC_ARTSC_0020	LATIN AMERICA AND CARIBBEAN	ARTSC_CHIN_1090	GREAT MINDS OF CHINA
14	ARTSC_ASTRON_0087	BASICS OF SPACE FLIGHT	ARTSC_CHIN_1090	GREAT MINDS OF CHINA
15	ARTSC_ASTRON_0086	OBSERVATIONAL ASTRONOMY	ARTSC_ENGLIT_0354	WORDS AND IMAGES
16	ARTSC_CHIN_1088	NEW CHINESE CINEMA	ARTSC_ENGLIT_0628	WORKING CLASS LITERATURE
17	ARTSC_AFRCNA_0352	AFRICAN AMERICAN DANCE	ARTSC_GER_0004	INTERMEDIATE GERMAN 2
18	ARTSC_AFRCNA_0352	AFRICAN AMERICAN DANCE	ARTSC_HAA_0040	INTRO TO WESTERN ARCHITECTURE
19	ARTSC_AFRCNA_1555	AFRO CARIBBEAN DANCE	ARTSC_HAA_0040	INTRO TO WESTERN ARCHITECTURE
20	ARTSC_CHEM_0100	PREPARATION GENERAL CHEMISTRY	ARTSC_HAA_0040	INTRO TO WESTERN ARCHITECTURE
21	ARTSC_AFRCNA_0385	CARIBBEAN HISTORY	ARTSC_HIST_0521	CARIBBEAN HISTORY
22	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_HIST_0678	US AND THE HOLOCAUST
23	ARTSC_ANTH_0536	MESOAMERICA BEFORE CORTEZ	ARTSC_HIST_1677	JEWS IN THE UNITED STATES
24	ARTSC_ENGLIT_0597	BIBLE AS LITERATURE	ARTSC_HIST_1775	ORIGINS OF CHRISTIANITY
25	ARTSC_ASTRON_0086	OBSERVATIONAL ASTRONOMY	ARTSC_HPS_0611	PRINCPL OF SCIENTIFIC REASNING
26	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_JS_0283	US AND THE HOLOCAUST
27	ARTSC_ANTH_0536	MESOAMERICA BEFORE CORTEZ	ARTSC_JS_1260	JEWS IN THE UNITED STATES
28	ARTSC_HIST_1677	JEWS IN THE UNITED STATES	ARTSC_JS_1260	JEWS IN THE UNITED STATES
29	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_LING_1235	LANGUAGE, GENDER AND SOCIETY
30	ARTSC_ARTSC_0020	LATIN AMERICA AND CARIBBEAN	ARTSC_LING_1235	LANGUAGE, GENDER AND SOCIETY
31	ARTSC_ASTRON_0087	BASICS OF SPACE FLIGHT	ARTSC_LING_1235	LANGUAGE, GENDER AND SOCIETY
32	ARTSC_CHIN_1090	GREAT MINDS OF CHINA	ARTSC_LING_1235	LANGUAGE, GENDER AND SOCIETY
33	ARTSC_ENGLIT_0597	BIBLE AS LITERATURE	ARTSC_MATH_0200	PREP FOR SCIENTIFIC CALCULUS
34	ARTSC_ASTRON_0086	OBSERVATIONAL ASTRONOMY	ARTSC_MATH_0230	ANALYTC GEOMETRY & CALCULUS 2
35	ARTSC_AFRCNA_1555	AFRO CARIBBEAN DANCE	ARTSC_MATH_0235	HONORS 1 VARIABLE CALCULUS
36	ARTSC_ENGLIT_0597	BIBLE AS LITERATURE	ARTSC_MATH_0240	ANALYTC GEOMETRY & CALCULUS 3
37	ARTSC_ASTRON_0086	OBSERVATIONAL ASTRONOMY	ARTSC_MATH_0400	DISCRET MATHEMATCL STRUCTURES
38	ARTSC_ASTRON_0086	OBSERVATIONAL ASTRONOMY	ARTSC_MUSIC_0100	FUNDAMENTALS OF WESTERN MUSIC
39	ARTSC_ANTH_0582	INTRODUCTION TO ARCHEOLOGY	ARTSC_MUSIC_0123	BASC MUSICIANSHIP: CLASS VOICE
40	ARTSC_CHEM_0910	CHEMCL PRINCPL HEALTH PROFESSN	ARTSC_MUSIC_0123	BASC MUSICIANSHIP: CLASS VOICE

Table 41 Sets of two General Education courses that cannot be taken together due to schedule conflicts for all the six terms under study (Part II)

Course Set	Course	Course Title	Course	Course Title
41	ARTSC_MATH_0025	APPLIED COLLEGE ALGEBRA	ARTSC_MUSIC_0123	BASC MUSICIANSHIP: CLASS VOICE
42	ARTSC_AFRCNA_1555	AFRO CARIBBEAN DANCE	ARTSC_MUSIC_0896	MUSIC AND FILM
43	ARTSC_MATH_0200	PREP FOR SCIENTIFIC CALCULUS	ARTSC_MUSIC_0896	MUSIC AND FILM
44	ARTSC_MATH_0235	HONORS 1 VARIABLE CALCULUS	ARTSC_MUSIC_0896	MUSIC AND FILM
45	ARTSC_MUSIC_0122	BASC MUSICIANSHIP: CLSS GUITAR	ARTSC_MUSIC_0896	MUSIC AND FILM
46	ARTSC_ASTRON_0086	OBSERVATIONAL ASTRONOMY	ARTSC_PHYS_0110	INTRODUCTION TO PHYSICS 1
47	ARTSC_CHEM_0910	CHEMCL PRINCPL HEALTH PROFESSN	ARTSC_PHYS_0111	INTRODUCTION TO PHYSICS 2
48	ARTSC_MATH_0025	APPLIED COLLEGE ALGEBRA	ARTSC_PHYS_0111	INTRODUCTION TO PHYSICS 2
49	ARTSC_MUSIC_0123	BASC MUSICIANSHIP: CLASS VOICE	ARTSC_PHYS_0111	INTRODUCTION TO PHYSICS 2
50	ARTSC_MUSIC_0122	BASC MUSICIANSHIP: CLSS GUITAR	ARTSC_PHYS_0175	BASC PHYS SCI & ENGR 2 (INTGD)
51	ARTSC_AFRCNA_0352	AFRICAN AMERICAN DANCE	ARTSC_PHYS_0475	INTRO PHYS SCIENCE & ENGRG 1
52	ARTSC_AFRCNA_1555	AFRO CARIBBEAN DANCE	ARTSC_PHYS_0475	INTRO PHYS SCIENCE & ENGRG 1
53	ARTSC_CHEM_0100	PREPARATION GENERAL CHEMISTRY	ARTSC_PHYS_0475	INTRO PHYS SCIENCE & ENGRG 1
54	ARTSC_HAA_0040	INTRO TO WESTERN ARCHITECTURE	ARTSC_PHYS_0475	INTRO PHYS SCIENCE & ENGRG 1
55	ARTSC_ASTRON_0086	OBSERVATIONAL ASTRONOMY	ARTSC_PSY_0405	LEARNING AND MOTIVATION
56	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_PSY_0510	SENSATION AND PERCEPTION
57	ARTSC_ASTRON_0088	STONEHENGE TO HUBBLE	ARTSC_PSY_0510	SENSATION AND PERCEPTION
58	ARTSC_ASTRON_0113	INTRODUCTION TO ASTRONOMY	ARTSC_PSY_0510	SENSATION AND PERCEPTION
59	ARTSC_HIST_0125	RELIGIONS OF THE WEST	ARTSC_RELGST_0105	RELIGIONS OF THE WEST
60	ARTSC_ENGLIT_0597	BIBLE AS LITERATURE	ARTSC_RELGST_0115	BIBLE AS LITERATURE
61	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_RELGST_0283	US AND THE HOLOCAUST
62	ARTSC_ENGLIT_0597	BIBLE AS LITERATURE	ARTSC_RELGST_1120	ORIGINS OF CHRISTIANITY
63	ARTSC_RELGST_0115	BIBLE AS LITERATURE	ARTSC_RELGST_1120	ORIGINS OF CHRISTIANITY
64	ARTSC_ANTH_0536	MESOAMERICA BEFORE CORTEZ	ARTSC_RELGST_1260	JEWS IN THE UNITED STATES
65	ARTSC_HIST_1677	JEWS IN THE UNITED STATES	ARTSC_RELGST_1260	JEWS IN THE UNITED STATES
66	ARTSC_JS_1260	JEWS IN THE UNITED STATES	ARTSC_RELGST_1260	JEWS IN THE UNITED STATES
67	ARTSC_HIST_1757	RELIGION IN INDIA 1	ARTSC_RELGST_1500	RELIGION IN INDIA 1
68	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_RELGST_1760	RELIGION AND RATIONALITY
69	ARTSC_ASTRON_0088	STONEHENGE TO HUBBLE	ARTSC_RELGST_1760	RELIGION AND RATIONALITY
70	ARTSC_ASTRON_0113	INTRODUCTION TO ASTRONOMY	ARTSC_RELGST_1760	RELIGION AND RATIONALITY
71	ARTSC_PSY_0510	SENSATION AND PERCEPTION	ARTSC_RELGST_1760	RELIGION AND RATIONALITY
72	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_RUSS_0800	MASTERPIECES 19THC RUSSIAN LIT
73	ARTSC_CHIN_1088	NEW CHINESE CINEMA	ARTSC_RUSS_0850	EARLY RUSSIAN CULTURE
74	ARTSC_AFRCNA_0352	AFRICAN AMERICAN DANCE	ARTSC_SPAN_0082	LATIN AMERICA TODAY
75	ARTSC_AFRCNA_1555	AFRO CARIBBEAN DANCE	ARTSC_SPAN_0082	LATIN AMERICA TODAY
76	ARTSC_CHEM_0100	PREPARATION GENERAL CHEMISTRY	ARTSC_SPAN_0082	LATIN AMERICA TODAY
77	ARTSC_HAA_0040	INTRO TO WESTERN ARCHITECTURE	ARTSC_SPAN_0082	LATIN AMERICA TODAY
78	ARTSC_PHYS_0475	INTRO PHYS SCIENCE & ENGRG 1	ARTSC_SPAN_0082	LATIN AMERICA TODAY
79	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	ARTSC_THEA_0840	INTRODUCTION TO THEATRE DESIGN
80	ARTSC_ANTH_0536	MESOAMERICA BEFORE CORTEZ	ARTSC_URBNST_0080	INTRODUCTION TO URBAN STUDIES
81	ARTSC_HIST_1677	JEWS IN THE UNITED STATES	ARTSC_URBNST_0080	INTRODUCTION TO URBAN STUDIES
82	ARTSC_JS_1260	JEWS IN THE UNITED STATES	ARTSC_URBNST_0080	INTRODUCTION TO URBAN STUDIES
83	ARTSC_RELGST_1260	JEWS IN THE UNITED STATES	ARTSC_URBNST_0080	INTRODUCTION TO URBAN STUDIES

Table 42 General Education Courses that appear in sets of two courses that cannot be enrolled together for any of the six academic terms under analysis

Crosslisted	Course	Course Title	Times Present Indiviually	Total Times Present	% / Total	Cumulative %
	ARTSC_ANTH_1601	STRUCTURE AND FUNCTION	12	12	7.23%	7.23%
Y	ARTSC_HIST_1677	JEWS IN THE UNITED STATES	4			
y	ARTSC_JS_1260	JEWS IN THE UNITED STATES	4	12	7.23%	14.46%
y	ARTSC_RELGST_1260	JEWS IN THE UNITED STATES	4			
	ARTSC_ASTRON_0086	OBSERVATIONAL ASTRONOMY	10	10	6.02%	20.48%
Y	ARTSC_ENGLIT_0597	BIBLE AS LITERATURE	5	7	4.22%	24.70%
Y	ARTSC_RELGST_0115	BIBLE AS LITERATURE	2			
	ARTSC_AFRCNA_1555	AFRO CARIBBEAN DANCE	5	5	3.01%	27.71%
	ARTSC_PHYS_0475	INTRO PHYS SCIENCE & ENGRG 1	5	5	3.01%	30.72%
	ARTSC_HAA_0040	INTRO TO WESTERN ARCHITECTURE	5	5	3.01%	33.73%
	ARTSC_SPAN_0082	LATIN AMERICA TODAY	5	5	3.01%	36.75%
	ARTSC_AFRCNA_0352	AFRICAN AMERICAN DANCE	4	4	2.41%	39.16%
	ARTSC_MUSIC_0123	BASC MUSICIANSHIP: CLASS VOICE	4	4	2.41%	41.57%
	ARTSC_ASTRON_0087	BASICS OF SPACE FLIGHT	4	4	2.41%	43.98%
	ARTSC_CHIN_1090	GREAT MINDS OF CHINA	4	4	2.41%	46.39%
	ARTSC_ASTRON_0113	INTRODUCTION TO ASTRONOMY	4	4	2.41%	48.80%
	ARTSC_URBNST_0080	INTRODUCTION TO URBAN STUDIES	4	4	2.41%	51.20%
	ARTSC_LING_1235	LANGUAGE, GENDER AND SOCIETY	4	4	2.41%	53.61%
	ARTSC_ARTSC_0020	LATIN AMERICA AND CARIBBEAN	4	4	2.41%	56.02%
	ARTSC_ANTH_0536	MESOAMERICA BEFORE CORTEZ	4	4	2.41%	58.43%
	ARTSC_MUSIC_0896	MUSIC AND FILM	4	4	2.41%	60.84%
	ARTSC_CHEM_0100	PREPARATION GENERAL CHEMISTRY	4	4	2.41%	63.25%
	ARTSC_RELGST_1760	RELIGION AND RATIONALITY	4	4	2.41%	65.66%
	ARTSC_PSY_0510	SENSATION AND PERCEPTION	4	4	2.41%	68.07%
Y	ARTSC_RELGST_1120	ORIGINS OF CHRISTIANITY	2	3	1.81%	69.88%
Y	ARTSC_HIST_1775	ORIGINS OF CHRISTIANITY	1			
Y	ARTSC_HIST_0678	US AND THE HOLOCAUST	1			
Y	ARTSC_JS_0283	US AND THE HOLOCAUST	1	3	1.81%	71.69%
Y	ARTSC_RELGST_0283	US AND THE HOLOCAUST	1			
	ARTSC_CHEM_0910	CHEMCL PRINCPL HEALTH PROFESSN	3	3	1.81%	73.49%
	ARTSC_PHYS_0111	INTRODUCTION TO PHYSICS 2	3	3	1.81%	75.30%
	ARTSC_ASTRON_0088	STONEHENGE TO HUBBLE	3	3	1.81%	77.11%
Y	ARTSC_AFRCNA_0385	CARIBBEAN HISTORY	1	2	1.20%	78.31%
Y	ARTSC_HIST_0521	CARIBBEAN HISTORY	1			
Y	ARTSC_HIST_1757	RELIGION IN INDIA 1	1	2	1.20%	79.52%
Y	ARTSC_RELGST_1500	RELIGION IN INDIA 1	1			
Y	ARTSC_HIST_0125	RELIGIONS OF THE WEST	1	2	1.20%	80.72%
Y	ARTSC_RELGST_0105	RELIGIONS OF THE WEST	1			
	ARTSC_MATH_0025	APPLIED COLLEGE ALGEBRA	2	2	1.20%	81.93%
	ARTSC_MUSIC_0122	BASC MUSICIANSHIP: CLSS GUITAR	2	2	1.20%	83.13%
	ARTSC_MATH_0235	HONORS 1 VARIABLE CALCULUS	2	2	1.20%	84.34%
	ARTSC_ANTH_0582	INTRODUCTION TO ARCHEOLOGY	2	2	1.20%	85.54%
	ARTSC_CHIN_1088	NEW CHINESE CINEMA	2	2	1.20%	86.75%
	ARTSC_MATH_0200	PREP FOR SCIENTIFIC CALCULUS	2	2	1.20%	87.95%
	ARTSC_MATH_0230	ANALYTC GEOMETRY & CALCULUS 2	1	1	0.60%	88.55%
	ARTSC_MATH_0240	ANALYTC GEOMETRY & CALCULUS 3	1	1	0.60%	89.16%
	ARTSC_PHYS_0175	BASC PHYS SCI & ENGR 2 (INTGD)	1	1	0.60%	89.76%
	ARTSC_ANTH_0620	BIOCULTURAL ANTHROPOLOGY	1	1	0.60%	90.36%
	ARTSC_MATH_0400	DISCRET MATHEMATCL STRUCTURES	1	1	0.60%	90.96%
	ARTSC_RUSS_0850	EARLY RUSSIAN CULTURE	1	1	0.60%	91.57%
	ARTSC_BIOSC_0150	FOUNDATIONS OF BIOLOGY 1	1	1	0.60%	92.17%
	ARTSC_MUSIC_0100	FUNDAMENTALS OF WESTERN MUSIC	1	1	0.60%	92.77%
	ARTSC_CHEM_0110	GENERAL CHEMISTRY 1	1	1	0.60%	93.37%
	ARTSC_AFRCNA_0639	HISTORY OF JAZZ	1	1	0.60%	93.98%
	ARTSC_GER_0004	INTERMEDIATE GERMAN 2	1	1	0.60%	94.58%
	ARTSC_PHYS_0110	INTRODUCTION TO PHYSICS 1	1	1	0.60%	95.18%
	ARTSC_THEA_0840	INTRODUCTION TO THEATRE DESIGN	1	1	0.60%	95.78%
	ARTSC_PSY_0405	LEARNING AND MOTIVATION	1	1	0.60%	96.39%
	ARTSC_RUSS_0800	MASTERPIECES 19THC RUSSIAN LIT	1	1	0.60%	96.99%
	ARTSC_HPS_0611	PRINCPL OF SCIENTIFIC REASNING	1	1	0.60%	97.59%
	ARTSC_CHEM_0760	UHC GENERAL CHEM FOR ENGINRS 1	1	1	0.60%	98.19%
	ARTSC_CHEM_0710	UHC GENERAL CHEMISTRY 1	1	1	0.60%	98.80%
	ARTSC_ENGLIT_0354	WORDS AND IMAGES	1	1	0.60%	99.40%
	ARTSC_ENGLIT_0628	WORKING CLASS LITERATURE	1	1	0.60%	**100.00%**
		Total	**166**	**166**	**100.00%**	

APPENDIX C. NEO4J SCRIPTS

This appendix includes the scripts used to create objects and load data into NEO4J. Comma delimited files are produced using ORACLE SQL in the relational schema discussed in Section 4.2.

COURSE nodes

```
USING PERIODIC COMMIT 1000
LOAD CSV WITH HEADERS FROM "file:C:/courses.csv" AS csvLine
CREATE (c:course { ARCHIVE_PERIOD: toInt(csvLine.ARCHIVE_PERIOD),
            ACAD_GROUP_CD: csvLine.ACAD_GROUP_CD,
            SUBJECT_CD: csvLine.SUBJECT_CD,
            CATALOG_NBR: toInt(csvLine.CATALOG_NBR),
            ACAD_GROUP_DESCR: csvLine.ACAD_GROUP_DESCR,
            SUBJECT_DESCR: csvLine.SUBJECT_DESCR,
            COURSE_DESCR: csvLine.COURSE_DESCR,
            SECTIONS: toInt(csvLine.SECTIONS),
            ENROLLED_CRSE: toInt(csvLine.ENROLLED_CRSE),
            CAP_CRSE: toInt (csvLine.CAP_CRSE)
            })
//////////////////////////
```

COURSE_SET nodes

```
USING PERIODIC COMMIT 1000
LOAD CSV WITH HEADERS FROM "file:C:/course_sets.csv" AS csvLine
CREATE (c:course_set { ARCHIVE_PERIOD: toInt(csvLine.ARCHIVE_PERIOD),
              ITEMSET_ID: toInt(csvLine.ITEMSET_ID),
              ITEMSET: csvLine.ITEMSET,
              ENROLLED: toInt(csvLine.ENROLLED),
              MAX_SEATS_LEFT: toInt(csvLine.MAX_SEATS_LEFT),
              MAX_INIT_CAPACITY: toInt(csvLine.MAX_INIT_CAPACITY),
              INTEREST_REASON: csvLine.INTEREST_REASON
              })

////////////////////////////////
```

SECTION nodes

```
USING PERIODIC COMMIT 1000
LOAD CSV WITH HEADERS FROM "file:C:/sections.csv" AS csvLine
CREATE (c:section { ARCHIVE_PERIOD : toInt(csvLine.ARCHIVE_PERIOD),
             ACAD_GROUP_CD: csvLine.ACAD_GROUP_CD,
             SUBJECT_CD: csvLine.SUBJECT_CD,
             CATALOG_NBR: toInt(csvLine.CATALOG_NBR),
             CLASS_NBR: toInt(csvLine.CLASS_NBR),
             MON: csvLine.MON,
```

```
              TUES: csvLine.TUES,
              WED: csvLine.WED,
              THURS: csvLine.THURS,
              FRI: csvLine.FRI,
              STARTT: csvLine.STARTT,
              STOPT: csvLine.STOPT,
              STARTNUM: toInt(csvLine.STARTNUM),
              STOPTNUM: toInt(csvLine.STOPTNUM),
              FACILITY_ID: csvLine.FACILITY_ID,
              ENRL_TOT: toInt(csvLine.ENRL_TOT),
              ENRL_CAP: toInt(csvLine.ENRL_CAP),
              NBR_MEETINGS: toInt(csvLine.nbr_meetings)
             });

////////////////////////////////
```

Relationship between courses (ENROLLED_TUPLE edges)

```
USING PERIODIC COMMIT 1000
LOAD CSV WITH HEADERS FROM "file:C:/Tuples_enrollment_2091.csv" AS csvLine
MATCH (c1: course {ARCHIVE_PERIOD: toInt(csvLine.ARCHIVE_PERIOD),
          ACAD_GROUP_CD: csvLine.ACAD_GROUP_CD1,
          SUBJECT_CD: csvLine.SUBJECT_CD1,
          CATALOG_NBR: toInt(csvLine.CATALOG_NBR1)
         }),
    (c2: course {ARCHIVE_PERIOD: toInt(csvLine.ARCHIVE_PERIOD),
          ACAD_GROUP_CD: csvLine.ACAD_GROUP_CD2,
          SUBJECT_CD: csvLine.SUBJECT_CD2,
          CATALOG_NBR: toInt(csvLine.CATALOG_NBR2)
          })
CREATE UNIQUE (c1)-[e:ENROLLED_TUPLE {enrolled: toInt(csvLine.ENROLLED)}
o]->(c2);

////////////////////////////
```

Relationship between course and course set (BELONGS_TO_SET edges)

```
USING PERIODIC COMMIT 1000
LOAD CSV WITH HEADERS FROM "file:C:/courseitemset2091.csv" AS csvLine
MATCH (c: course { ARCHIVE_PERIOD: toInt(csvLine.ARCHIVE_PERIOD),
          ACAD_GROUP_CD: csvLine.ACAD_GROUP_CD,
          SUBJECT_CD: csvLine.SUBJECT_CD,
          CATALOG_NBR: toInt(csvLine.CATALOG_NBR) }),
   (s: course_set { ARCHIVE_PERIOD: toInt(csvLine.ARCHIVE_PERIOD),
            ITEMSET_ID: toInt(csvLine.ITEMSET_ID) })
CREATE UNIQUE (c) -[b:BELONGS_TO_SET]-> (s);


////////////////
```

STUDENTS

```
LOAD CSV WITH HEADERS FROM "file:C:/students.csv" AS csvLine
CREATE (c:student { EMPLID: toInt(csvLine.EMPLID)});
```

```
--/////////////////
```
Relationship Students ENROLLED IN sections

```
USING PERIODIC COMMIT 1000
LOAD CSV WITH HEADERS FROM "file:C:/Transactions_2131_with_CLASS_NBR.csv" AS csvLine
MATCH (s: section { ARCHIVE_PERIOD: toInt(csvLine.ARCHIVE_PERIOD),
          CLASS_NBR: toInt(csvLine.CLASS_NBR) }) -[:BELONGS_TO_COURSE]-> c,
   (i: student { EMPLID: toInt(csvLine.EMPLID)})
CREATE UNIQUE (i) -[b:ENROLLS_IN]-> (s);

--///////////////
```

APPENDIX D. STATISTICS ON COURSE NODES ENROLLMENT, DEGREE, AND ENROLLMENT WEIGHTED DEGREE CENTRALITY

Table 43 below shows a summary of statistics on EWDC by academic term for the six fall terms between 2091 and 2141. Table 44 and figures 36, 37 and 38 show detail statistics on Enrollment, Degree and Weighted Degree Centrality of course nodes for academic term 2141. The statics for other terms show similar patterns and are thus omitted for brevity. While the Degree distribution is highly skewed, the Weighted Degree Centrality presents a more normalized pattern as it corrects the distortion caused by the number of students enrolled in a course.

Table 43 Statistics for course Enrollment Weighted Degree Centrality (EWDC) by academic term

	2091	2101	2111	2121	2131	2141
Maximum	10.000	7.000	6.500	7.000	7.000	7.000
Minimum	0.051	0.049	0.049	0.060	0.048	0.042
Mean	2.492	2.477	2.485	2.534	2.539	2.562
Standard Deviation	1.144	1.144	1.117	1.158	1.172	1.168
EWDC Percentile 0.1	0.792	0.729	0.801	0.823	0.747	0.818
EWDC Percentile 0.9	3.750	3.786	3.776	3.842	3.938	4.000

Table 44 Statistics on enrollment, Degree and Enrollment Weighted Degree Centrality (EWDC) for academic term 2141

Statistics for Archive Period 2141

		ENROLLMENT	DEGREE	EWDC
N	Valid	1534	1534	1534
	Missing	0	0	0
Mean		62.60	95.97	2.56181397725
Std. Error of Mean		3.579	2.509	.029824413389
Std. Deviation		140.166	98.256	1.168112281668
Variance		19646.587	9654.336	1.364
Skewness		8.145	2.227	-.241
Std. Error of Skewness		.062	.062	.062
Kurtosis		90.844	6.226	-.493
Std. Error of Kurtosis		.125	.125	.125
Range		2357	736	6.958333333
Minimum		1	2	.041666667
Maximum		2358	738	7.000000000
Percentiles	10	5.00	12.00	.81857474200
	20	11.00	26.00	1.47619047600
	25	14.00	34.00	1.69895725325
	30	16.00	40.00	1.99275362300
	40	22.00	52.00	2.38461538500
	50	30.00	65.00	2.73252032500
	60	37.00	81.00	3.04166666700
	70	49.00	109.00	3.30429292950
	75	57.00	122.00	3.42155870475
	80	70.00	139.00	3.53846153800
	90	126.00	214.50	4.00000000000

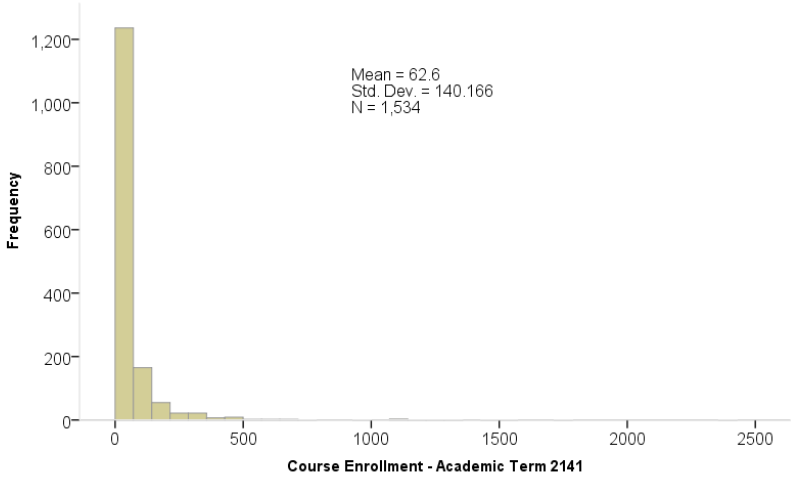


Figure 36 Course enrollment frequency – academic term 2141

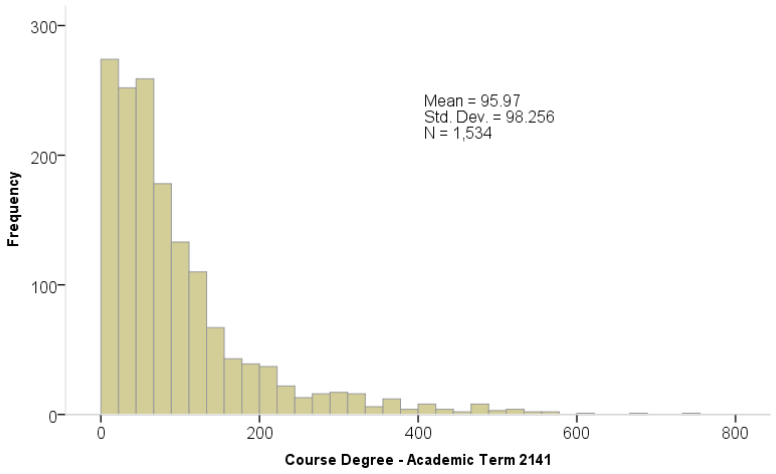


Figure 37 Course Degree frequency – archive period 2141

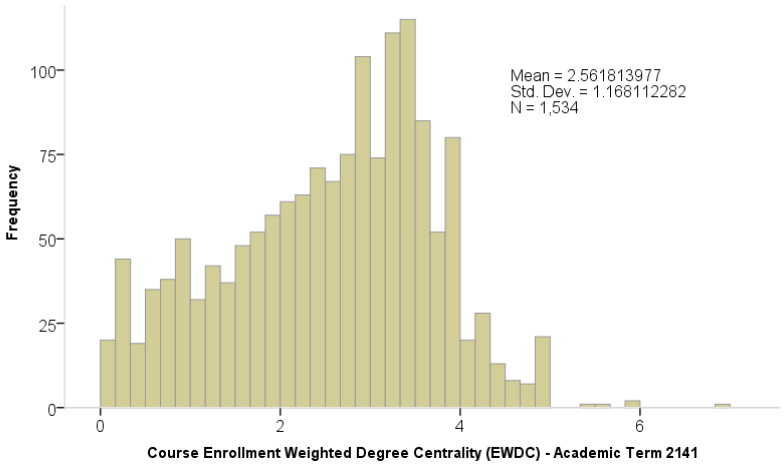


Figure 38 Course Enrollment Weighted Degree Centrality (EWDC) frequency – archive period 2141

APPENDIX E. STATISTICS ON CLIQUE GRAPH EDGE ATTRIBUTES OVERLAP AND WEIGHTED OVERLAP

This appendix presents descriptive statistics on the distribution of the metrics *overlap* and *weighted overlap* discussed in Section 5.4 as they result from the implementation of GCPM on the clique graph. The distributions are equivalent for the six term under analysis. Thus, for brevity only the results for academic term 2141 are shown (academic term from September to December 2013). Table 45 and Figures 39, 40 and 41 below illustrate highly skewed distributions for overlap and weighted overlap. The majority of course itemsets (cliques) that have courses in common have an overlap of two (i.e. they have two courses in common).

Table 45 Summary of statistics for Overlap and Weighted Overlap measures in clique graph -academic term 2141

		WEIGHTED_OVERLAP	OVERLAP	LOG10_WO
N	Valid	4429996	4429996	4429996
	Missing	0	0	0
Mean		.00330424	1.76	-2.7743
Std. Error of Mean		.000003763	.000	.00021
Median		.00139000	2.00	-2.8570
Mode		.000870	2	-3.06
Std. Deviation		.007919834	.728	.43947
Variance		.000	.530	.193
Skewness		40.642	.947	.839
Std. Error of Skewness		.001	.001	.001
Kurtosis		6598.186	1.614	.721
Std. Error of Kurtosis		.002	.002	.002
Range		2.158040	6	3.89
Minimum		.000280	1	-3.55
Maximum		2.158320	7	.33
Sum		14637.786020	7808829	-12290091.28
Percentiles	25	.00080000	1.00	-3.0969
	50	.00139000	2.00	-2.8570
	75	.00295000	2.00	-2.5302

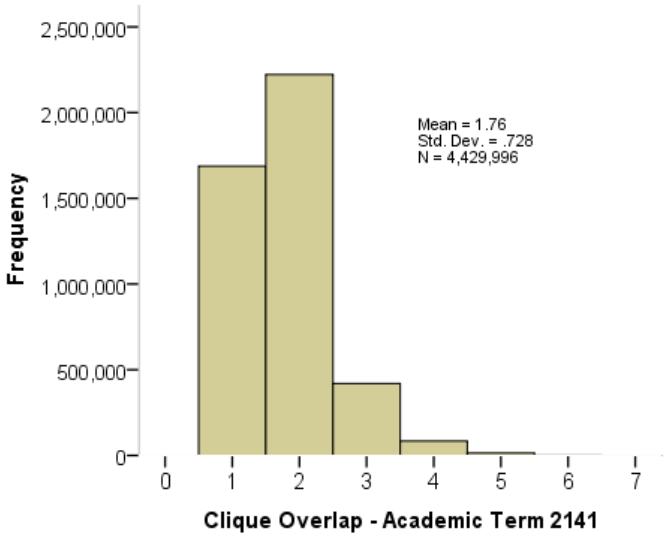


Figure 39 Clique Overlap frequency distribution – academic term 2141

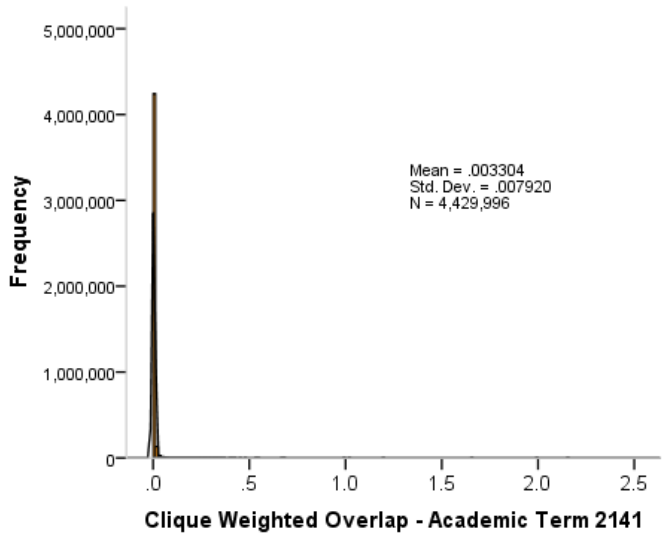


Figure 40 Weighted Clique Overlap frequency distribution – academic term 2141

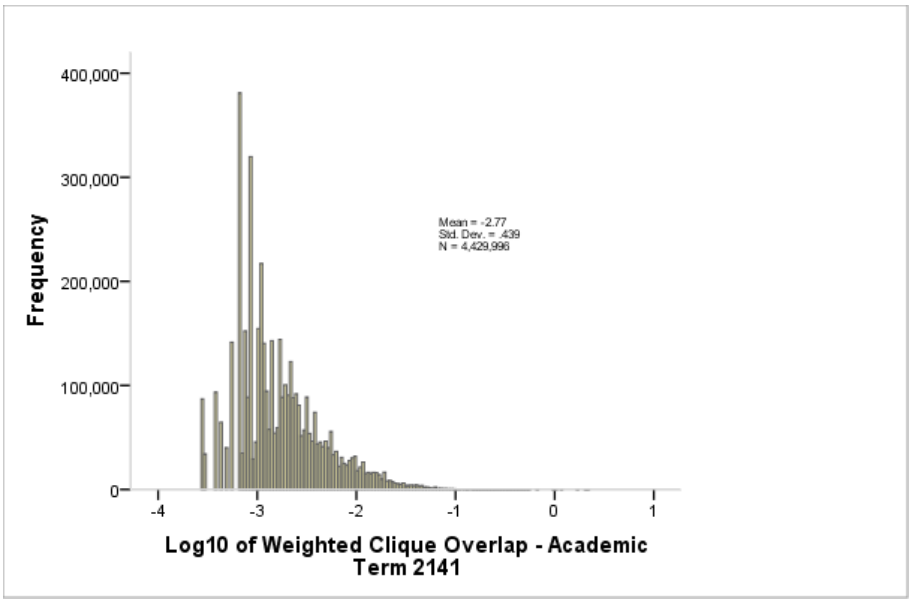


Figure 41 Log10 of Weighted Clique Overlap frequency distribution – academic term 2141

BIBLIOGRAPHY

Agrawal, R., Imieliński, T., & Swami, A. (1993). *Mining association rules between sets of items in large databases.* Paper presented at the ACM SIGMOD Record.

Agrawal, R., & Srikant, R. (1994). *Fast algorithms for mining association rules.* Paper presented at the Proc. 20th Int. Conf. Very Large Data Bases, VLDB.

Ahn, Y.-Y., Bagrow, J. P., & Lehmann, S. (2010). Link communities reveal multiscale complexity in networks. *Nature, 466*(7307), 761-764.

Barabási, A.-L. (2009). Scale-free networks: a decade and beyond. *science, 325*(5939), 412.

Barabási, A.-L., & Bonabeau, E. (2003). Scale-free networks. *Scientific American, 288*(5), 50-59.

Barraclough, E. D. (1965). The application of a digital computer to the construction of timetables. *The Computer Journal, 8*(2), 136-146.

Barreto, M., Szóstek, A., & Karapanos, E. (2013). An initial model for designing Socially Translucent systems for Behavior Change.

Begole, J. B., Tang, J. C., Smith, R. B., & Yankelovich, N. (2002). *Work rhythms: analyzing visualizations of awareness histories of distributed groups.* Paper presented at the Proceedings of the 2002 ACM conference on Computer supported cooperative work.

Blanchette, S. M. (2010). Space and Power in the Ivory Tower: Decision Making in Public Higher Education.

Bonutti, A., De Cesco, F., Di Gaspero, L., & Schaerf, A. (2012). Benchmarking curriculum-based course timetabling: formulations, data formats, instances, validation, visualization, and results. *Annals of Operations Research, 194*(1), 59-70.

Borgelt, C. (2012). Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2*(6), 437-456.

Braithwaite, P., Brauer, R., Bruhn, C., Gustafson, A., Kujak-Ford, N., McGlinn, E., . . . Rubow, P. (2012). Classroom Space Utilization. *Administative Excellence Project at University of Wisconsin-Madison.* from http://adminexcellence.wisc.edu/content/uploads/2012/08/Space-Utilization_Classroom_Campus-Forum-Presentation_2012_06_20.pdf

Bron, C., & Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM, 16*(9), 575-577.

Burke, E. K., De Causmaecker, P., Berghe, G. V., & Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of Scheduling, 7*(6), 441-499.

Burke, E. K., Kendall, G., Mısır, M., Özcan, E., Burke, E., Kendall, G., . . . Mısır, M. (2004). *Applications to timetabling.* Paper presented at the Handbook of Graph Theory, chapter 5.6.

Carter, M. W. (2001). A comprehensive course timetabling and student scheduling system at the University of Waterloo *Practice and Theory of Automated Timetabling III* (pp. 64-82): Springer.

CollegeNET. SCHEDULE25. from http://corp.collegenet.com/products/Schedule25_overview.html

Danisch, M., Guillaume, J.-L., & Le Grand, B. (2014). Multi-ego-centered communities in practice. *Social network analysis and mining, 4*(1), 1-10.

De Causmaecker, P., & Berghe, G. V. (2012). Towards a reference model for timetabling and rostering. *Annals of Operations Research, 194*(1), 167-176.

de Werra, D. (1985). An introduction to timetabling. *European Journal of Operational Research, 19*(2), 151-162.

De Werra, D. (1997). The combinatorics of timetabling. *European Journal of Operational Research, 96*(3), 504-513.

de Werra, D. (1997). Restricted coloring models for timetabling. *Discrete Mathematics, 165*, 161-170.

Derényi, I., Palla, G., & Vicsek, T. (2005). Clique percolation in random networks. *Physical review letters, 94*(16), 160202.

Di Gaspero, L., McCollum, B., & Schaerf, A. (2007). *The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (track 3).* Paper presented at the Proc. of the 14th RCRA workshop on Exper. Eval. of Algo. for Sol. Prob. with Combinatorial Explosion, Rome, Italy. Citeseer.

Díaz, O., & Puente, G. (2010). *Model-aware Wiki analysis tools: the case of HistoryFlow.* Paper presented at the Proceedings of the 6th international symposium on Wikis and open collaboration.

Dimopoulou, M., & Miliotis, P. (2001). Implementation of a university course and examination timetabling system. *European Journal of Operational Research, 130*(1), 202-213.

Dimopoulou, M., & Miliotis, P. (2004). An automated university course timetabling system developed in a distributed environment: A case study. *European Journal of Operational Research, 153*(1), 136-147.

Erickson, T., Halverson, C., Kellogg, W. A., Laff, M., & Wolf, T. (2002). Social translucence: designing social infrastructures that make collective activity visible. *Communications of the ACM, 45*(4), 40-44.

Erickson, T., & Kellogg, W. A. (2000). Social translucence: an approach to designing systems that support social processes. *ACM transactions on computer-human interaction (TOCHI), 7*(1), 59-83.

Erickson, T., Kellogg, W. A., Laff, M., Sussman, J., Wolf, T. V., Halverson, C. A., & Edwards, D. (2006). *A persistent chat space for work groups: the design, evaluation and deployment of loops.* Paper presented at the Proceedings of the 6th conference on Designing Interactive systems.

Erickson, T., Smith, D. N., Kellogg, W. A., Laff, M., Richards, J. T., & Bradner, E. (1999). *Socially translucent systems: social proxies, persistent conversation, and the design of "babble".* Paper presented at the Proceedings of the SIGCHI conference on Human Factors in Computing Systems.

Evans, T., & Lambiotte, R. (2009). Line graphs, link partitions, and overlapping communities. *Physical Review E, 80*(1), 016105.

Evans, T. S. (2010). Clique graphs and overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment, 2010*(12), P12037.

Everett, M. G., & Borgatti, S. P. (1998). Analyzing clique overlap. *Connections, 21*(1), 49-61.

Fernandes, P., Pereira, C. S., & Barbosa, A. (2015). A decision support approach to automatic timetabling in higher education institutions. *Journal of Scheduling*, 1-14.

Feuerstein, S., & Pribyl, B. (2005). Oracle pl/sql Programming.

Fink, I. (2002). Classroom Use and Utilization. *Facilities Manager, 18*(3), 13-24.

Fogarty, J., Lai, J., & Christensen, J. (2004). Presence versus availability: the design and evaluation of a context-aware communication client. *International Journal of Human-Computer Studies, 61*(3), 299-317.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports, 486*(3), 75-174.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 35-41.

Freeman, L. C. (1979). Centrality in social networks conceptual clarification. *Social networks, 1*(3), 215-239.

Fu, C., Kang, Z., Zhicun, F., Lansheng, H., & Jing, C. (2014). *K-clique community detection based on union-find.* Paper presented at the Computer, Information and Telecommunication Systems (CITS), 2014 International Conference on.

Geller, S. (2004). *Timetabling at the University of Sheffield, UK-hardening the incremental approach to timetable development.* Paper presented at the PATAT.

Gilbert, E. (2012). *Designing social translucence over social networks.* Paper presented at the Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems.

Granell, C., Darst, R. K., Arenas, A., Fortunato, S., & Gómez, S. (2015). A benchmark model to assess community structure in evolving networks. *arXiv preprint arXiv:1501.05808*.

Han, J., Kamber, M., & Pei, J. (2006). *Data mining: concepts and techniques*: Morgan kaufmann.

Harary, F. (1969). Graph theory: Addison-Wesley, Reading, MA.

Havemann, F., Gläser, J., & Heinz, M. (2015). Detecting Overlapping Link Communities by Finding Local Minima of a Cost Function with a Memetic Algorithm. Part 1: Problem and Method. *arXiv preprint arXiv:1501.05139*.

He, D., Liu, D., Jin, D., & Zhang, W. (2015). A Stochastic Model for Detecting Heterogeneous Link Communities in Complex Networks.

Henning, M., Brandes, U., & Pfeffer, J. (2012). Studying Social Networks M.

Hinkin, T. R., & Thompson, G. M. (2002). SchedulExpert: Scheduling courses in the Cornell University school of hotel administration. *Interfaces, 32*(6), 45-57.

Holzman, A. G., & Turkes, W. (1964). OPTIMAL SCHEDULING IN EDUCATIONAL INSTITUTIONS.

Jin, D., Gabrys, B., & Dang, J. (2015). Combined node and link partitions method for finding overlapping communities in complex networks. *Scientific reports, 5*.

Kanawati, R. (2015). Empirical evaluation of applying ensemble methods to ego-centred community identification in complex networks. *Neurocomputing, 150*, 417-427.

Kirshstein, R., & Wellman, J. (2012). Technology and the Broken Higher Education Cost Model: Insights from the Delta Cost Project. *Educause Review, 47*(5), 12-14.

Koh, Y. S., & Pears, R. (2007). Efficiently finding negative association rules without support threshold *AI 2007: Advances in Artificial Intelligence* (pp. 710-714): Springer.

Krajca, P., Outrata, J., & Vychodil, V. (2011). Using frequent closed itemsets for data dimensionality reduction. 1128-1133.

Kramer, M., Dutkowski, J., Yu, M., Bafna, V., & Ideker, T. (2014). Inferring gene ontologies from pairwise similarity data. *Bioinformatics, 30*(12), i34-i42.

Kristiansen, S., & Stidsen, T. R. (2013). A comprehensive study of educational timetabling-a survey.

Kumpula, J. M., Kivelä, M., Kaski, K., & Saramäki, J. (2008). Sequential algorithm for fast clique percolation. *Physical Review E, 78*(2), 026109.

Lam, S. S. K. (1997). The effects of group decision support systems and task structures on group communications and decision quality. *Journal of Management Information Systems, 13*(4), 22.

Lancichinetti, A., Fortunato, S., & Kertész, J. (2009). Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics, 11*(3), 033015.

Lancichinetti, A., Radicchi, F., Ramasco, J. J., & Fortunato, S. (2011). Finding statistically significant communities in networks. *PloS one, 6*(4), e18961.

Lee, C., Reid, F., McDaid, A., & Hurley, N. (2010). Detecting highly overlapping community structure by greedy clique expansion. *arXiv preprint arXiv:1002.1827*.

Lewis, R., Paechter, B., & McCollum, B. (2007). *Post enrolment based course timetabling: A description of the problem model used for track two of the second international timetabling competition*: Cardiff Business School.

Mani, T. (2012). Mining negative association rules. *IOSR Journal of Computer Engineering (IOSRJCE), 3*(6), 43-47.

Mathieu, P., & Verrons, M.-H. (2006). *How to solve a timetabling problem by negotiation?* Paper presented at the Proceedings of The 6th International Conference on the Practice and Theory of Automated Timetabling (PATAT 06).

McCollum, B. (1998). The implementation of a central timetabling system in a large british civic university. 237-253.

McCollum, B. (2007). A perspective on bridging the gap between theory and practice in university timetabling *Practice and Theory of Automated Timetabling VI* (pp. 3-23): Springer.

McCollum, B., McMullan, P., Burke, E. K., Parkes, A. J., & Qu, R. (2007). The second international timetabling competition: Examination timetabling track: Technical Report QUB/IEEE/Tech/ITC2007/Exam/v4. 0/17, Queens University, Belfast (UK).

McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A. J., . . . Burke, E. K. (2010). Setting the research agenda in automated timetabling: The second international timetabling competition. *INFORMS Journal on Computing, 22*(1), 120-130.

McDonald, D. W., Gokhman, S., & Zachry, M. (2012). *Building for social translucence: a domain analysis and prototype system.* Paper presented at the Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work.

Mikut, R., & Reischl, M. (2011). Data mining tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1*(5), 431-443.

Miller, J. J. (2013). Graph Database Applications and Concepts with Neo4j.

Moody, D. L. (2011). *A Tiling Approach to Producing High Quality Solutions to Real World Timetabling Problems.* City University of New York.

Müller, T. (2009). ITC2007 solver description: a hybrid approach. *Annals of Operations Research, 172*(1), 429-446.

Müller, T., & Barták, R. (2002). *Interactive timetabling: Concepts, techniques, and practical results.* Paper presented at the PATAT.

Müller, T., & Murray, K. (2010). Comprehensive approach to student sectioning. *Annals of Operations Research, 181*(1), 249-269.

Müller, T., & Rudová, H. (2012). *Real-life curriculum-based timetabling.* Paper presented at the Dag Kjenstad, Atle Riise, Tomas Eric Nordlander, Barry McCollum and Edmund Burke. Proccedings of the 9th International Conference on the Practice and Theory of Automated Timetabling. Son, Norway: SINTEF.

Murphy, J., Sutter, R., & Laboratories, E. F. (1966). *School Scheduling by Computer: The Story of GASP*: Educational Facilities Laboratories.

Murray, K., Müller, T., & Rudová, H. (2007). Modeling and solution of a complex university course timetabling problem *Practice and Theory of Automated Timetabling VI* (pp. 189-209): Springer.

NEO4J. Cypher Query Language.

The Neo4j Manual v2.2.0-M02. from http://neo4j.com/docs/milestone/cypher-query-lang.html

Oprea, M. (2007). MAS_UP-UCT: A multi-agent system for university course timetable scheduling. *International Journal of Computers, Communications & Control, 2*(1), 94-102.

Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social networks, 32*(3), 245-251.

Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature, 435*(7043), 814-818.

Paul, M., Anand, R., & Anand, A. (2015). Detection of highly overlapping communities in complex networks. *Journal of Medical Imaging and Health Informatics, 5*(5), 1099-1103.

Pennacchioli, D., Coscia, M., & Pedreschi, D. (2014). Overlap versus partition: marketing classification and customer profiling in complex networks of products. 103-110.

Piechowiak, S., Ma, J., & Mandiau, R. (2005). An open interactive timetabling tool *Practice and Theory of Automated Timetabling V* (pp. 34-50): Springer.

Pitt, T., & Tepper, S. (2012). Double majors: Influences, identities, and impacts. *Nashville, Tenn.: The Curb Center for Art, Enterprise and Public Policy at Vanderbilt University.*, Pages 9 and 11. Retrieved from: http://www.vanderbilt.edu/curbcenter/manage/files/Teagle-Report-Final-3-11-13-2.pdf

Porter, M. A., Onnela, J.-P., & Mucha, P. J. (2009). Communities in networks. *Notices of the AMS, 56*(9), 1082-1097.

Raeder, T., & Chawla, N. V. (2011). Market basket analysis with networks. *Social network analysis and mining, 1*(2), 97-113.

Rani, B. K., Srinivas, K., Reddy, B. R., & Govardhan, A. (2011). Mining Negative Association Rules. *International Journal of Engineering and Technology, 3*.

Reid, F., McDaid, A., & Hurley, N. (2012). *Percolation computation in complex networks.* Paper presented at the Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012).

Rubio, R. G., & Munoz, D. P. (2004). *A timetable production system architecture for course and exams.* Paper presented at the PATAT.

Rudová, H., Müller, T., & Murray, K. (2011). Complex university course timetabling. *Journal of Scheduling, 14*(2), 187-207.

Sánchez, C. A. (2014). *Timetabling in Higher Education: Considering the Combinations of Classes Taken by Students.* Paper presented at the 10th International Conference of the Practice and Theory of Automated Timetabling - PATAT, York, United Kingdom.

Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review, 13*(2), 87-127.

Schaerf, A., & Di Gaspero, L. (2007). Measurability and reproducibility in university timetabling research: discussion and proposals *Practice and Theory of Automated Timetabling VI* (pp. 40-49): Springer.

Sherman, G. R. (1958). *The Sequential Method of Scheduling Students*: Purdue Research Foundation.

Srikant, R., Vu, Q., & Agrawal, R. (1997). *Mining association rules with item constraints.* Paper presented at the KDD.

Stallaert, J. (1997). Automated timetabling improves course scheduling at UCLA. *Interfaces, 27*(4), 67-81.

Standish, T. A. (1995). Data structures, algorithms, and software principles in C.

Steiner, I. (1972). Group process and productivity.

Stuart, H. C., Dabbish, L., Kiesler, S., Kinnaird, P., & Kang, R. (2012). *Social transparency in networked information exchange: a theoretical framework.* Paper presented at the Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work.

Szostek, A. M., Karapanos, E., Eggen, B., & Holenderski, M. (2008). *Understanding the implications of social translucence for systems supporting communication at work.* Paper presented at the Proceedings of the 2008 ACM conference on Computer supported cooperative work.

Tao, H., Wu, Z., Shi, J., Cao, J., & Yu, X. (2014). *Overlapping Community Extraction: A Link Hypergraph Partitioning Based Method.* Paper presented at the Services Computing (SCC), 2014 IEEE International Conference on.

Thompson, G. L., & Thore, S. A. (1996). *Computational economics* (Vol. 68): Baltzer Science.

UNITIME. University Timetabling, Comprehensive Academic Timetabling Solutions. from http://www.unitime.org/

Van Bruggen, R. (2014). Graphs for HR Analytics

Wattenberg, M., Viégas, F. B., & Hollenbach, K. (2007). Visualizing activity on wikipedia with chromograms *Human-Computer Interaction–INTERACT 2007* (pp. 272-287): Springer.

Webber, J. (2012). *A programmatic introduction to Neo4j.* Paper presented at the Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity.

Wu, X., Zhang, C., & Zhang, S. (2004). Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems (TOIS), 22*(3), 381-405.

Xie, J., Kelley, S., & Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys (CSUR), 45*(4), 43.

Yellen, J., & Wehrer, A. (2013). The Design and Implementation of an Interactive Course-Timetabling System.

Yuan, X., Buckles, B. P., Yuan, Z., & Zhang, J. (2002). *Mining negative association rules.* Paper presented at the Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on.

Zaki, M. J., & Ogihara, M. (1998). *Theoretical foundations of association rules.* Paper presented at the 3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery.

Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997). *New Algorithms for Fast Discovery of Association Rules.* Paper presented at the KDD.

Zeising, M., & Jablonski, S. (2012). *A Generic Approach to Interactive University Timetabling.* Paper presented at the ACHI 2012, The Fifth International Conference on Advances in Computer-Human Interactions.

Zhang, Z., & Wang, Z. (2015). Mining overlapping and hierarchical communities in complex networks. *Physica A: Statistical Mechanics and its Applications, 421*, 25-33.