Graphical tools for analysing the development of competencies in an engineering program

Rein Vesilo
Macquarie University
Corresponding Author Email: rein.vesilo@mq.edu.au

ABSTRACT

CONTEXT

Engineering competencies play an important role in planning curriculum goals and course design. The traditional focus has been on technical skills; however, more recently, human-orientated skills have increased in importance using terms such as generic, professional, and transferable skills with these skills now firmly embedded in the curriculum. In addition, the management of engineering programs is constrained by factors such as the cost of innovation, the cost of scaling up, the diversity of student intake, limited access to industry partners, and so on. This increase in the number and complexity of program requirements has made it more difficult to extract the vital indicators of how competencies are developed, which are essential for closing the loop program aims and outcomes.

PURPOSE OR GOAL

The aim of the work is to develop concepts, techniques, and tools that focus on the structural aspects of the curriculum, such as the sequencing of courses and interactions between courses, to determine the influence that different components have in developing competencies. Influence here refers to courses that are strongly connected to other courses to mediate knowledge development and requirements satisfaction.

APPROACH OR METHODOLOGY/METHODS

The approach used for analysing a program is to apply graphical network analysis and theory. The relationship between courses and competencies was represented as a bipartite graph with link weights calculated using data from the Course Handbook. Dependency analysis and the BiRank measure allowed the influence of courses and competencies to be determined that also included the effect of neighbouring nodes. Course prerequisite graphs were used to model knowledge flow using the perspectives of knowledge development and requirements satisfaction.

ACTUAL OR ANTICIPATED OUTCOMES

The analysis was applied to a mechanical engineering program that had a strong component of professional skills. The analysis of the bipartite graph determined how competencies were developed depending on the type of course (technical or generic skills) and the type of competency (discipline, design, or professional). Dependency analysis identified which courses could act as crossover points between different streams. Inclusion of competency alignment in the prerequisite graphs showed a spreading effect of influence between courses.

CONCLUSIONS/RECOMMENDATIONS/SUMMARY

The results verified the effectiveness of using graph theory to extract meaningful information from complex program and course data using structural information. Influential components of the program could be determined as well as hidden aspects of the curriculum such as cross-dependencies. This information can be used to identify critical courses that require more teaching attention, facilitate course redesign, and aid resource allocation.

KEYWORDS

Curriculum, competencies, graph theory, centrality, dependence

Introduction

The rigorous specification of engineering competencies is vital for curriculum planning and course design. The traditional focus has been on technical skills. However, engineering practice requirements have broadened due to increased demands for interdisciplinarity, sustainability, globalisation, workforce mobility, and related human-orientated and employability skills. The imagining of the future engineering graduate is one who is "T-shaped," with a long vertical line of depth on technical skills and knowledge and a wide horizontal line of breadth in generic skills. By the term generic skills, we include a broad range of skills including writing, communication, critical thinking, self-learning, leadership, teamwork, business skills, and professional skills such as professional behaviour, ethics, demeanour, and identity. In this paper, we also consider transferrable engineering skills that include systems thinking, design, and engineering analysis. Such capabilities are now firmly established in the engineering curriculum, accreditation requirements, and national engineering policy documents. Despite the increased inclusion of generic skills in programs, the expectations for technical ability remain high (Burnett et al., 2021).

However, the increasing breadth and diversity of requirements in the curriculum introduces several problems. One problem is how to achieve an integrated program so that different parts of the program combine coherently. Another problem is that, with different criteria used in different knowledge areas, it is more difficult to measure competency development, which is essential to close the loop between curriculum objectives and actual outcomes. Engineering programs are also experiencing a range of factors that the constrain the options available for designing and administering programs and courses that address these problems. These include the cost of innovation, the cost of scaling up, limited access to industry partners, limited availability of appropriately qualified staff, competing pressures on staff, the diversity of student intake, the quality of student intake, and accreditation difficulties (Crosthwaite, 2019).

The aim of this paper is to develop concepts, techniques, and tools to make the management of the complexity of multiple requirements and contextual factors easer; so that when faced with large amounts of data the detection of patterns and contributing factors is facilitated. The approach is to focus on the development and measurement of competencies, as these encapsulate the learning outcomes and accreditation requirements of a program. The paper focusses on the structural aspects of the curriculum as expressed through the sequencing of courses to achieve knowledge development and interactions between courses, using graph theory to extract key competency measures. The main areas of graphical network theory used are the concepts of centrality and dependence to determine which courses and competencies are the most influential and to unearth relationships between courses and sources of influence that may otherwise be hidden. A significant contribution of the work is the inclusion of competencies and the increased understanding this provides to curriculum analysis.

Background

There is extensive literature now on engineering competencies. (Male, Bush, & Chapman, 2011) rank engineering competencies in the following order: communication, working in diverse teams, self-management, professionalism, creativity/problem-solving, management leadership, business, practical engineering, innovation, contextual responsibility (social global environment, etc.), applying technical theory. Other lists are given by (Chan, Zhao, & Luk, 2017), (Boelt, Kolmos, & Holgaard, 2022), (Sharma, De Costa, & Heyzer, 2014). Although the use of graph theory for curriculum analysis has been conducted by other researchers, this paper expands on these applications of graph theory by explicitly including standard competencies in the analysis. Some of these other studies include the following. (Lightfoot, 2010) uses graph theory to determine favourable places in the curriculum for topic introduction, reinforcement, and assessment in a Business Administration program. (Stavrinides & Zuev, 2023) produced a course network graph with 771 nodes and 772 links in their analyses of courses at the California Institute of Technology (Caltech) to improve the prediction of enrolments, progression rates, graduation time, and student performance, enhance student learning experiences, assist program navigation, improve resource allocation, and facilitate interactions between departments. (Pavlich-Mariscal, Curiel, & Chavarro, 2019) examined the CDIO approach to curriculum design to instil a desired body of knowledge and use a backward design process (Wiggins & McTighe, 2005) to determine desired outcomes, define specific prerequisite topics, and identify improvement opportunities. (Aldrich, 2015) analyses a biochemistry and molecular biology program. Courses with many prerequisites are places of integration, whereas courses with many successors are information sources. Forward flow coherence occurs when the courses use information from the prerequisite courses effectively. Lateral coherence refers to connections between courses across different streams, where no formal prerequisites are specified. (Lie, Brennan, & Nygren, 2018) use graph theory to connect graduate attributes, attribute indicators, program course, learning outcomes, and grading components. (O'Meara & Vaidya, 2021) represent the connections between topics in a textbook as a complex system where ideas and concepts are interconnected and use a constructivist approach to personalise the curriculum.

Concepts of graph theory

A graph is defined formally by a set of nodes (sometimes called vertices) and a set of links (sometimes called edges), where a link connects two nodes. The nodes represent a set of entities of interest and the links represent relationships between the nodes (the prerequisite graph for the case study in the paper is shown in Figure 1). In a directed graph, each link has a specified direction from a source node to a destination mode. This can be used to indicate a flow or movement (e.g., by clicking on a hyperlink on a web page to transfer control to another web page). In an undirected graph, the links have no directionality. This is often used to indicate an association between two entities or their or co-occurrence. Graphs in which all nodes belong to the same class of objects are called unimodal. For a graph with N nodes, the adjacency matrix $A = \left(A_{to,from}\right)_{N\times N}$, represents the graph as a matrix where the rows and columns correspond to nodes and the matrix entries represent the links. In an unweighted graph, the matrix entries describe the presence or absence of a link between two nodes. If there is a link to node "*to*" from node "*from*" then $A_{to,from} = 1$, otherwise $A_{to,from} = 0$. In a weighted graph $A_{to,from}$ can be any nonnegative number that indicates the strength of the link.

Centrality analysis determines which nodes are the most influential in a graph. In the context of curriculum analysis, influence refers to how strongly connected courses are to other courses to mediate knowledge development and requirements satisfaction. For an unweighted undirected graph, the degree of a node is the number of links joined to a node. For an unweighted directed graph, the outdegree of a node is the number of links that originate from the node and the indegree is the number of links that terminate at the node. For weighted graphs, degrees are obtained by summing the weights of links correspondingly. Nodes with high degree have greater connectedness and are often considered more influential.

PageRank and related centrality measures extend degree centrality by taking into consideration the influence of neighbouring nodes: Being directly connected to another node with high influence can increase one's own influence, giving the node an influence out of proportion to the number of direct connections it has. Calculating the PageRank centrality involves adding two terms. The first term is the influence of a node that is acquired from neighbouring nodes (network effect) and the second term is the inherent influence of a node. A parameter α specifies the contribution of the network effect and $1-\alpha$ specifies the inherent influence. If $x = (x_1, x_2, \cdots, x_N)$ is the vector of node centralities and x_0 is the vector of inherent node centralities, the formula for computing centrality is $x = \alpha W x + (1-\alpha)x_0$ where $0 < \alpha < 1$, and $W$is obtained from the adjacency matrix A by dividing the columns by the corresponding outdegree of a node. PageRank is the basis of the Google search engine, where the outdegree of a page equals the number of hyperlinks on a page.

Bipartite graphs

A bipartite graph is a graph with two types of nodes. The only links are from a node of one type to a node of the other type. In this paper, one node type represents courses, and the other node type represents shared competencies (other examples include patient-doctor, agent-event relationships etc). The links considered here are undirected, and their weights (between 0 and 1) represent how much of a course is devoted to developing a particular competency. The rows in the adjacency matrix represent competencies and the columns represent courses. The projection of a bipartite graph involves the derivation of a unimodal graph from the initial bimodal graph. Two projections are possible. In the course projection, the bipartite graph is projected onto a graph of courses by reflecting competency contributions back to the course nodes in proportion to the link weights and indicates the strength of the association between two courses expressed via shared competencies.

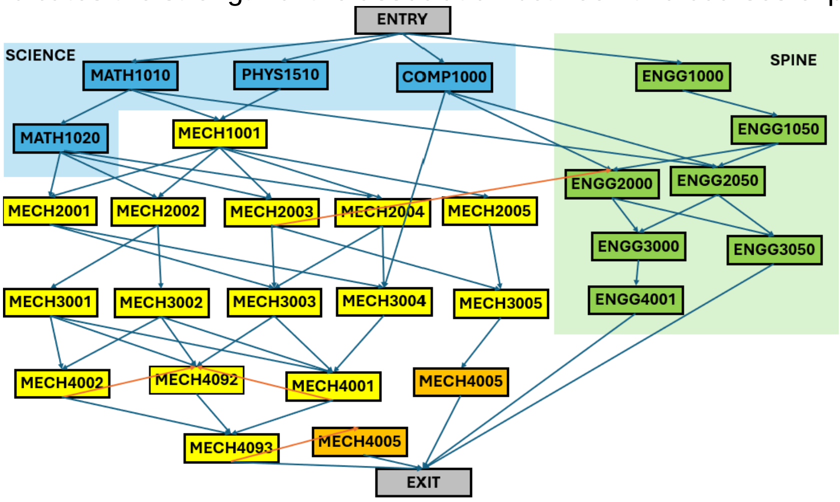

Figure 1: Graph for program case study

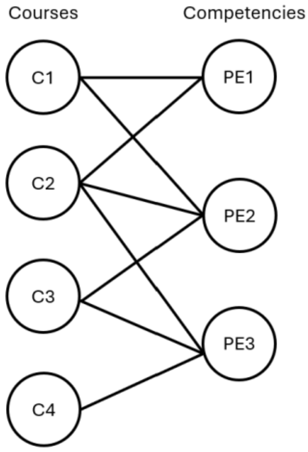

Figure 2 Bipartite graph

In the competency projection, the bipartite graph is projected onto a graph of competencies; this indicates the association of competencies through shared courses. The course projected graph can be used to determine the dependency between two courses. A course C1 (leader) is said to depend on another course C2 (follower) if C1 develops a competency, say PE1, then so does C2. In this case, C1 and C2 would provide useful lateral coherency links for PE1 in the curriculum, so that the common competency PE1 can be exploited to link content and learning in courses C1 and C2. Courses with low dependency are either developers of a restricted set of competencies or spread their contribution thinly over many competencies. A numeric value for the dependence of one course on another course is obtained using the formula given by (Gerdes, 2014). The dependency matrix expresses the dependence values over all possible pairwise combination of courses. In this matrix, if a row, representing say course C3, has multiple large values then many courses are dependent on it. The dependency coefficient of a course is the average of the pairwise dependencies in a row. In the example, C3 would have a high dependency coefficient. However, the projection process can blur important relations that may be present in the original bipartite graph and give misleading results. An alternative class of centrality measures for a bipartite graph produces two sets of centrality measures, one for each side of the graph. These algorithms can be obtained by simultaneously solving two equations: $c = \alpha W^T p + (1 - \alpha) = c_0$ and $p = \beta W c + (1 - \beta) p_0$, where c is the centrality of courses, p is the centrality of competencies and the suffix 0 indicates inherent centrality. Measures differ in the way W is determined. HITS and CoHITS are older algorithms but suffer from sensitivity to outliers and excessive influence of high-degree nodes. BiRank is a newer and more robust algorithm, and is the one used in this paper[1].

---

[1]In PageRank, $W = K_{PE}^{-1/2} A K_{CO}^{-1/2}$ where K_{PE} and K_{CO} are the diagonal outdegree matrices for competencies and courses, respectively. (For details, see (He, Gao, Kan, & Wang, 2016) and (Yang, Aronson, Odabas, Ahn, & Perry, 2022)).

Case Study

The Mechanical Engineering program at Macquarie University was used as a case study. The software used was MATLAB and Python with custom developed code. The program and course requirements data were taken from the publicly available course handbook. The structure of the program is shown in Figure 3 (see also Figure 1), where the number of courses for a component is in parentheses.

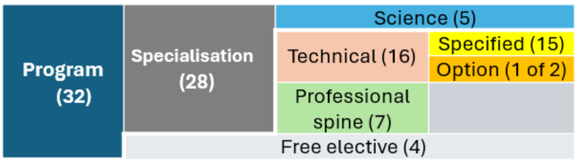

Figure 3: Structure of program

All courses were considered excluding free electives, except that one free elective was used for an option course. Most of the technical and spine courses included a mapping of learning outcomes to a set of standard competencies based on the Engineers Australia Stage One Competencies (see Table 1). The contribution a course made to the development of a competency was calculated as follows. First, the assessment weight was divided equally between each course learning outcome associated with the assessment item. Then the total contribution to each learning outcome was obtained by summing over the contributions of each assessment item. Then, the contribution of each learning outcome was divided equally between all the competencies that included it. Finally, the total course contribution for a competency was obtained by summing the competency contributions of each learning outcome. A bipartite graph was constructed to represent the mapping of courses to standardised competencies. The contribution of each course to each competency is shown on the heat map in Figure 4. The total program contribution to the development of each competency was obtained by summing the contributions of individual courses. The results are given in the final column of Table 1. The highest weights are for PE21, PE11, and PE32. The lowest weights are for PE14, PE34 and PE33. This can be used to make comparisons with the desired program outcomes and to make recommendations for change if necessary.

Table 1: Engineers Australia Stage One Competencies used in Course Handbook

Category	Code	Description	Contrib.(%)
Discipline knowledge	PE11	Comprehensive theory-based understanding of the underpinning fundamentals applicable to the engineering discipline	9.58
	PE12	Conceptual understanding of underpinning maths, analysis, statistics, computing	7.68
	PE13	In-depth understanding of specialist bodies of knowledge	7.33
	PE14	Discernment of knowledge development and research directions	2.38
	PE15	Knowledge of engineering design practice	5.61
	PE16	Understanding of the scope, principles, norms, and accountabilities of sustainable engineering practice	3.88
Generic design and project skills	PE21	Application of established engineering methods to complex problem solving	10.80
	PE22	Fluent application of engineering techniques, tools, and resources	8.57
	PE23	Application of systematic engineering synthesis and design processes	7.78
	PE24	Application of systematic approaches to the conduct and management of engineering projects	5.86
Profession-al skills	PE31	Ethical conduct and professional accountability	5.31
	PE32	Effective oral and written communication in the professional and lay domains	9.56
	PE33	Creative, innovative and pro-active demeanour	3.80
	PE34	Professional use and management of information	2.69
	PE35	Orderly management of self- and professional conduct	5.00
	PE36	Effective team membership and team leadership	4.06

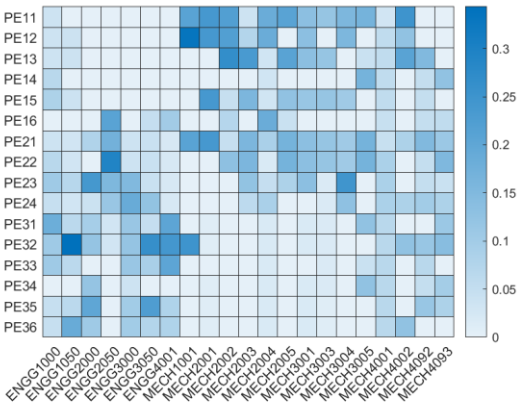

Figure 4: Mapping of courses to competencies

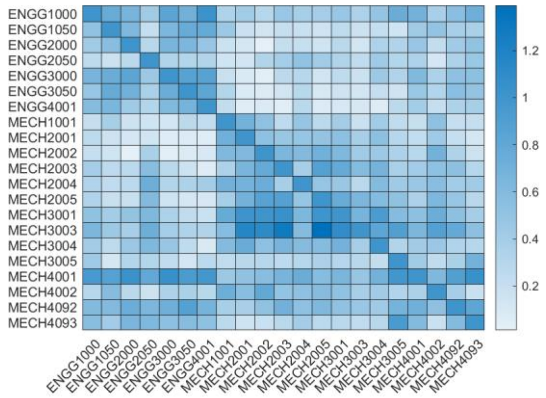

Figure 5: Dependency matrix courses

Competencies were categorised as discipline knowledge, generic engineering design and project knowledge, and professional skills. The courses were classified as technical and spine. The contributions different types of courses made to different types of competencies were determined by summing over sets of courses and competencies (see Table 2).

Table 2: Competency contributions to program

Competency Course t	Discipline	Design	Professional	Totals
Technical	32.97	22.12	11.12	66.22
Spine	3.54	10.92	19.33	33.78
Totals	36.54	33.04	30.45	100

Table 3: Competency contributions per course

Competency Course	Discipline	Design	Professional	Totals
Technical	48.79	32.74	16.46	98.00
Spine	10.47	32.31	57.22	100
Totals	36.02	32.60	30.05	98.67

The contribution of all courses to each competency was slightly more than 1/3 for the discipline competencies, approximately 1/3 for the design competencies, and slightly less than 1/3 for the professional competencies. A prominent feature was the small contribution the spine made to discipline competencies. In Table 2, there are more technical courses (14) than spine courses (7). Table 3 shows the percentage contributions per course. Unsurprisingly, technical courses contributed more to discipline competencies, and spine courses contributed more professional competencies. The technical and spine courses contributed roughly equally to the design competencies. The entries for the dependency matrix are shown in the heat map in Figure 5. Strong dependencies exist from spine courses to other spine courses, between early technical courses and from ENGG4001 to other spine courses. MECH4002 is an anomaly with low dependence. Figure 6 shows the dependency coefficients obtained from the projection onto the courses. The highest-scoring courses were MECH4001, MECH3003, MECH4092 (see the Appendix for the list of course names). Of the spine courses ENGG3000 stands out. These courses are leaders that develop competencies that follower courses develop (e.g. the followers of MECH4001 in Figure 7 have higher weights in the same row, such as the spine courses, MECH3005, MECH4093) and provide suitable places for lateral coherence. Low-dependence courses include MECH1001, MECH3005, ENGG4001 that contribute to a limited set of competencies. Figure 7 shows the competency dependency scores. The highest scoring competencies were PE34, PE24, and PE14. PE32 (communication) stands out as a professional competency with low dependence suggesting further investigation as to why this is so.

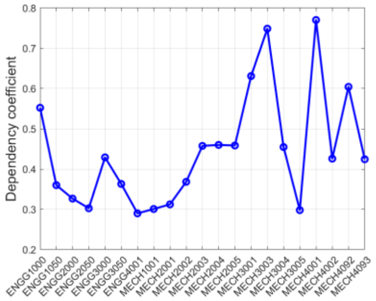

Figure 6: Course dependency scores

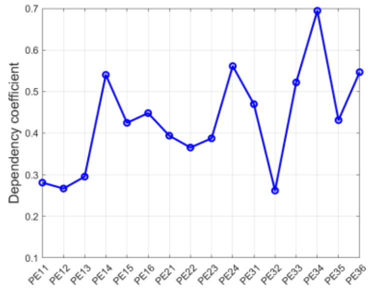

Figure 7: Competency dependency scores

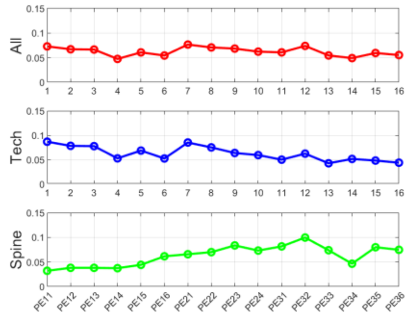

Figure 8: BiRank competency scores

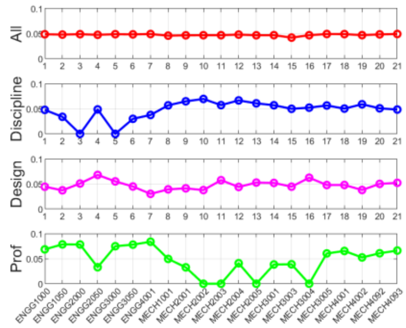

Figure 9: BiRank course scores

BiRank analysis was conducted on different groups of courses and different groups of competencies, by reducing the overall bipartite graph as appropriate. The values of the BiRank parameter were $\alpha = \beta = 0.7$. Figure 8 shows the BiRank scores for the group of all competencies in different course groups. The BiRank ordering is almost the same as that given by descriptive analysis, regardless of the group of courses. This is because the sum of competencies equals one for each course (except for one course that did not assign a learning outcome to any competency). In the group of all courses, the competencies with the highest ranking were PE21, PE11, and PE32. The lowest ranked were PE14, PE34, and PE33. Across the group of technical courses, discipline competencies were preferred, while within the group of spine courses, professional competencies were preferred. Figure 9 shows the BiRank scores for the group of all courses when the bipartite graph is restricted to different groups of competencies. In this case, the competency contributions for a course do not add up to one, and the BiRank scores are different for different groups of competencies. Among the group of discipline competencies, technical courses were favoured and spine courses had a much lower rank. Over the group of professional competencies, spine and later year technical courses were preferred while early year technical courses had a lower rank. For design competencies, both spine and technical courses contributed similarly, although there was variation within each group. When PageRank is compared to indegree, indirect network effects occur, typically producing a smoothing of scores (results not shown).

Competency development

The analysis of program-wide competency development was performed using unimodal course graphs in which knowledge is accumulated by progressing through courses. The graph links represent the prerequisite relationships between courses. Our focus is on the flow of knowledge where prerequisites specify the knowledge that is essential or desirable for success in a course (see Figure 1). (Although we recognise that prerequisites can be used to accomplish other goals such as helping instructors understand and manage student deficiencies, be used as used gatekeepers (Hriez & Al-Naymat, 2021), and be used by administrators for planning.). Two perspectives were used to construct the graphs. In the bottom-up perspective, earlier courses are a foundation upon which new knowledge is built. This leads to a directed graph where prerequisites point to the courses they support. We call this graph the forward graph. The indegree of a course is the number of prerequisites the course has. PageRank includes the indirect influence of earlier courses that have been passed through prerequisite courses. For high PageRank courses, there are challenges of integrating concepts or selectively using concepts from a large palette of possibilities (Aldrich, 2015). Such courses require greater use of high-order skills in Bloom's taxonomy to analyse, evaluate, select, and combine concepts (Lightfoot, 2010). General principles for organising sequences of courses include the following (Ornstein & Hunkins, 2018):

- Simple to complex learning
- Part-to-whole learning: The assumption is that certain bits of knowledge must be comprehended before other bits can be understood.
- Whole-to-part learning: The focus on giving an overview first.
- Chronological learning: Topics are presented in order of occurrence (e.g., as in history).
- Concept-related learning: The focus is on knowledge structure and interrelationships.
- Inquiry-related learning: Learning reflects the processes of scholarly investigation.
- Learner-related sequence: The focus is on student learning experience and activity.
- Utilisation-based learning: The focus is on how practitioners use knowledge in the world.

In the forward graph, the sequence of courses typically involves simple-to-complex, part-to-whole, and concept-related learning.

The top-down perspective is based on requirements satisfaction, where the program end goals and requirements are passed back to earlier courses. This is the basis for backward curriculum design. This leads to a directed graph where successor courses point back to the courses they follow. We call the resulting graph the backward graph. The indegree of a course is the number of successor courses that a course has. PageRank includes the influence that has been received indirectly from later courses. High PageRank courses are considered fundamental and provide a useful place to present introductory material and 'framework' concepts. (Lightfoot, 2010). (Aldrich, 2015) describes such courses as information sources. The sequence of courses is based on utilisation-based learning and whole-to-part learning. However, since these courses are required by a greater variety of successor courses, there is a pedagogical challenge in preparing students to learn successfully in a wider range of contexts, including the workplace context. In the transfer model of learning (National Research Council, 2000), new learning involves a transfer based on previous learning. Transfer is typically improved when students learn to use general principles. In an undirected graph. influence is acquired from all adjacent nodes and high connectivity is favoured. An analysis of the undirected graph is not included.

Prerequisite analysis

The prerequisite analysis was conducted for all courses in the program, except free electives. For simplicity, prerequisites and corequisites were treated equally. To ensure that there were loops in the graph, a link from EXIT to ENTRY was included. Figures 10 and 11 present results for the indegree and PageRank measures for the forward and backward graphs, respectively. The courses are shown on the horizontal axis, and are grouped from left to right into spine, science, and discipline courses. PageRank used the value $\alpha = 0.5$.

In the forward graph (Figure 10**Figure 10**), high PageRank courses were mostly technical courses of the next year, indicating the high level of integration of these courses. Spine courses had a moderate ranking, indicating the involvement of a variety of generic and design skills. The early science and discipline courses had lower PageRank values. These courses tended to focus on a single conceptual area. Anomalous courses were as follows. Courses with high PageRank and low degree included MECH4093, ENGG4093, and ENGG4001. These courses receive indirect influence and express the need to continue the integration of knowledge from prerequisite courses. Courses with low PageRank and higher indegree included MECH2004, MECH2003, MECH2002, and MECH2001. These courses have prerequisites that are shared by multiple other courses.

In the backward graph (Figure 11), higher values of PageRank are seen for foundation courses in science and the discipline. These courses need to be well designed so that knowledge can be effectively transferred. Later year technical courses tended to have lower PageRank. Anomalous courses were the following. High PageRank courses with low indegree included PHYS1510, MECH4092, and ENGG1000. These courses need to consider not only transfer to the successor courses, but also subsequent courses. Courses with low PageRank but with higher indegree included MECH3004, MECH3002, MECH3001, MECH4002. Low PageRank suggests that less consideration may be given to successors, allowing more flexible teaching.

Graphs for courses where a standard set of competencies is specified

The competency mapping information was then included in the study of prerequisite graphs. This information was only available for technical and spine courses, and the graphs were reduced to include only these courses. In this case, the influence of a prerequisite course in the forward graph and the influence of a successor course in the backward graph depended on the degree of alignment of the course competencies. The degree of alignment was represented as a link weight that was calculated as the cosine of the angle between the vectors of competency contributions for the two courses joined by the link. Unweighted versions of the graphs were used for comparison.

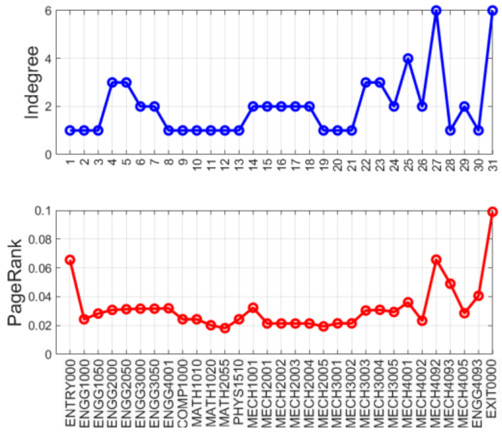

Figure 10: Forward graph for all courses

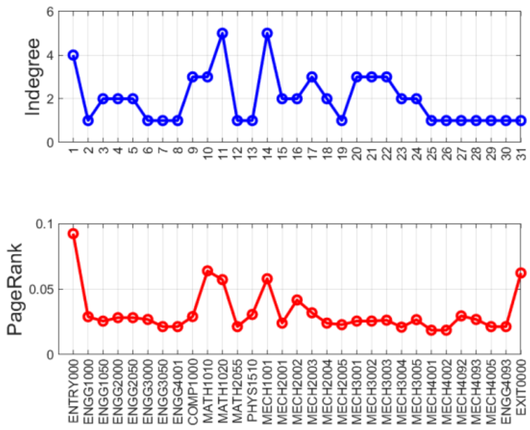

Figure 11: Backward graph for all courses

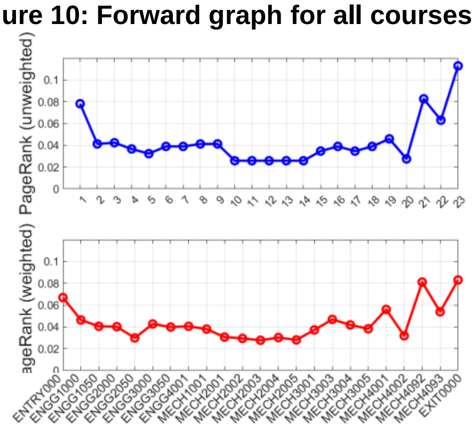

Figure 12: Forward graph with competency alignment

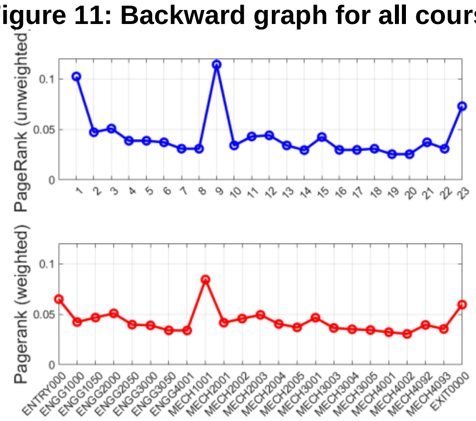

Figure 13: Backward graph with competency alignment

In the forward graphs (Figure 12), the unweighted and weighted graphs had a similar shape, with the later technical courses and the spine courses having greater influence. Somewhat surprisingly, the early MECH courses had the smallest ranks. When competencies were included, the greatest increases in ranks were for MECH3004 and MECH3003, indicating closer alignment with prerequisite courses. In the backward graph (Figure 13), unweighted and weighted graphs had a similar shape. MECH1001 stood out as one of the most influential courses by far. However, competency alignment reduced its rank, suggesting course designers need to consider carefully how successor courses continue knowledge development. The greatest increases in ranks when competencies were included were for MECH2005 and ENGG2000, indicating an increase in alignment with the successor courses.

Conclusion

The application of graph theory was found to be an insightful technique for determining the actual levels of competency development in a program, determining which courses contributed most to developing specific competencies and vice versa, and investigating the complex relationships in programs resulting from the sequencing of courses and the dependencies between courses. PageRank allowed the influences of non-immediately connected courses to be included. The tools developed can be used to assess actual outcomes against desired outcomes and enable anomalous courses and competencies, whose influence may have been under- or over- estimated, to be detected and be suggestive of further investigation and action. Software run times were only a few seconds allowing a quick turnaround for analysing different factors; however, data entry took longer, with a semi-automated conversion from the handbook being used. Analysis of knowledge growth and requirements and use of the forward and backward graphs, respectively, identified pedagogical issues and highlighted courses that require additional resources. Future work involves improving the software user interface and the comparison of multiple programs to understand structural relations that involve discipline, design, and professional skills.

Appendix

Table 4: List of courses in the Mechanical Engineering program

Science	Spine	Discipline		
COMP1000 Introduction to Computer Programming MATH1010 Calculus and Linear Algebra I MATH1020 Calculus and Linear Algebra II MATH2055 Engineering Mathematics II PHYS1510 Engineering Physics	ENGG1000 Introduction to Engineering ENGG1050 Engineering Design ENGG2000 Engineering Practice ENGG2050 Engineering Systems and Design Thinking ENGG3000 Engineering Project Practice ENGG3050 Engineering Leadership and Entrepreneurship ENGG4001 Professional Practice	MECH1001 Introduction to Mechanical Engineering MECH2001 Engineering Dynamics MECH2002 Fluid Mechanics MECH2003 Mechanical Design 1 MECH2004 Mechanics of Solids MECH2005 Engineering Materials	MECH3001 Thermo-dynamics MECH3002 Heat and Mass Transfer MECH3003 Mechanical Design 2 MECH3004 Applied Numerical Engineering MECH3005 Manufacturing Engineering	MECH4001 Product Design Engineering MECH4002 Energy Sustainable Design MECH4092 Mechanical Engineering Research Thesis A MECH4093 Mechanical Engineering Research Thesis B ENGG4093 Engineering Research Thesis Extension B MECH4005 Production Processes

References

Aldrich, P. R. (2015). The curriculum prerequisite network: Modeling the curriculum as a complex system. *Biochemistry and Molecular Biology Education, 43*(3), 168-180.

Boelt, A. M., Kolmos, A., & Holgaard, J. E. (2022). Literature review of students' perceptions of generic competence development in problem-based learning in engineering education. *European Journal of Engineering Education*, 1-22.

Burnett, I., Cameron, I., Crosthwaite, C., Foley, B., Goldfinch, T., Hargreaves, D., Wilson, J. (2021). Engineering Futures 2035 - 2021 Engineering Change - The future of engineering education in Australia. Australian Council of Engineering Deans (ACED).

Chan, C. K., Zhao, Y., & Luk, L. Y. (2017). A validated and reliable instrument investigating engineering students' perceptions of competency in generic skills. *Journal of Engineering Education, 106*(2), 299-325.

National Research Council, (2000). *How people learn: Brain, mind, experience, and school, Expanded edition* (A. B. J.D. Bransford, and R.R. Cocking Ed.): Commission on Behavioral and Social Sciences and Education. Washington, DC: National Academy Press.

Crosthwaite, C. (2019). Engineering Futures 2035: A scoping study. In: Australian Council of Engineering Deans (ACED).

Gerdes, L. M. (2014). Dependency centrality from bipartite social networks. *Connections, 34*, 14-29.

He, X., Gao, M., Kan, M.-Y., & Wang, D. (2016). Birank: Towards ranking on bipartite graphs. *IEEE Transactions on Knowledge and Data Engineering, 29*(1), 57-71.

Hriez, R. F., & Al-Naymat, G. (2021). A framework to capture the dependency between prerequisite and advanced courses in higher education. *Journal of Computing in Higher Education, 33*, 807-844.

Lie, S., Brennan, R., & Nygren, A. (2018). Graph-based approach to model the dependency information of graduate attributes for supporting the accreditation process. *Proceedings of the Canadian Engineering Education Association.*.

Lightfoot, J. M. (2010). A graph-theoretic approach to improved curriculum structure and assessment placement. *Communications of the IIMA, 10*(2), 5.

Male, S. A., Bush, M. B., & Chapman, E. S. (2011). An Australian study of generic competencies required by engineers. *European Journal of Engineering Education, 36*(2), 151-163.

O'Meara, J., & Vaidya, A. (2021). A network theory approach to curriculum design. *Entropy, 23*(10), 1346.

Ornstein, A., & Hunkins, F. (2018). Curriculum, foundations, principles, and issues global edition (Seventh Edition ed.): Pearson Education Limited.

Pavlich-Mariscal, J. A., Curiel, M., & Chavarro, G. (2019). *CDIO curriculum design for computing: A graph-based approach.* Paper presented at the Proceedings of the 15th International CDIO Conference.

Sharma, M., De Costa, G., & Heyzer, D. (2014). Analysis of employability skills for civil engineers in New Zealand.

Stavrinides, P., & Zuev, K. M. (2023). Course-prerequisite networks for analyzing and understanding academic curricula. *Applied Network Science, 8*(1), 19.

Wiggins, G. P., & McTighe, J. (2005). *Understanding by design*: ASCD (www.ascd.org).

Yang, K.-C., Aronson, B., Odabas, M., Ahn, Y.-Y., & Perry, B. L. (2022). Comparing measures of centrality in bipartite patient-prescriber networks: A study of drug seeking for opioid analgesics. *PLOS ONE, 17*(8), https://doi.org.1371/journal.pone.0273569.

Copyright statement