MTH3022 Project

Candidate numbers : 196722, 100612

1 Introduction

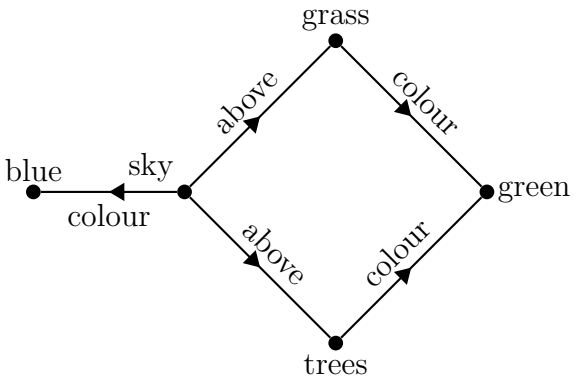
Cathegorising knowledge is a real-word problem. An example is to compare the semantic content of two articles related to a same topic. In fact, two networks can represent the same field and yet have different kinds of classifications and different cathegories. In this paper, we study two different ways to cathegorise mathematical knowledge based on Exeter University's mathematics bachelor programme. We chose to generate two graphs : one based on a semantic analysis of the syllabus plan of each module, and the other based on the academic trajectory of five students who linked the modules that they followed to their content. We explore different ways to measure the "similarity" of these graphs while underpinning the difference between a "subjective" and "objective" graph production of the mathematical knowledge.

2 Semantic Web

The semantic web is a set of standards defined by the World Wide Web Consortium[1] with the goal of allowing computers to parse and understand internet data. The primary technologies they developed for this are the Resource Description Framework[2] (RDF) and the Web Ontology Language[3] (OWL).

2.1 RDF

The Resource Description Framework is designed to standardise links between subjects and objects, and while it was initially designed for metadata, it is now used generally across a wide variety of domains. At its core are RDF graphs, sets of `(subject, predicate, object)` triples. `subject`s and `object`s are called resources (typically either text or IRIs[4]) and denote something that exists, while `predicate`s denote properties and are always IRIs. These depict a relationship between the `subject` and the `object`, with `predicate` indicating the type of relationship. This can be depicted as a directed graph, with `subject`s and `object`s being vertices, and `predicate`s being edge labels. For example the RDF graph {`(sky, colour, blue)`, `(sky, above, grass)`, `(sky, above, trees)`, `(grass, colour, green)`, `(trees, colour, green)`} could be depicted as below.



2.2 OWL

The Web Ontology Language (OWL) and its successor (OWL2) are designed to be consistent frameworks to describe a variety of concepts related to the internet. It does this by providing a set of axioms which constrain types of items (called "classes") and what relationships are permitted between them. This makes it possible to reason with data created by someone else because they will all use the same terms. OWL generally uses RDF (or a similar format) for describing the actual layout of the data, instead concerning itself with what sort of constraints those relationships model.

There are multiple different types of both OWL and OWL2, designed for different use cases. OWL2 has a general specification *OWL2 DL*[5] and then there exist profiles which remove some of the complexity to make them easier to use for particular domains, while still remaining fully compatible with *DL*. These profiles are *OWL2 EL*, designed for relatively simple reasoning over very large datasets, *OWL2 QL* which allows queries about relationships between items to be answered very efficiently, for example in databases, and *OWL2 RL* which allows the user to reason directly on RDF triples.

3 Methods

Let $A = (V_A, E_A, \varepsilon_A)$ and $B = (V_B, E_B, \varepsilon_B)$ be two graphs with respective adjacency matrices $\mathbf{A}_A$ and $\mathbf{A}_B$. Let $n_A = |E_A|$, $n_B = |E_B|$ and $m_A = |V_A|$, $m_B = |V_B|$

3.1 Similarity

Each vertex has a unique id (a unique name). Two vertices of different graphs *match* when they have the same id. We say that the edges $e_A \in E_A$ and $e_B \in E_B$ are *matching* if $\varepsilon_A(e_A)$ and $\varepsilon_B(e_B)$ map to matching vertices.

According to [6], the *similarity* between two networks is the mean of four measurements:

Edge strength similarity L_S :

$$L_S = \frac{\sum_{i=1}^{n} |S_i^A - S_i^B|}{\sum_{i=1}^{n} |S_i^A + S_i^B|},$$

where S_i^A and S_i^B are the strength (or weights) of the edge i in both graphs.

Matching edge ratio L_M :

$$L_M = \frac{N_E}{n},$$

where N_E is the number of edges matching in both networks, and n is the total number of edges, $n = \max(n_A, n_B) = \max(|E_A|, |E_B|)$.

Vertex ratio V_M :

$$V_M = \frac{N_V}{m},$$

where N_E is the number of matching vertices between networks, and m is the total number of vertices, $m = \max(m_A, m_B) = \max(|V_A|, |V_B|)$.

Matching Cluster Ratio V_C :

$$V_C = \frac{N_C}{m},$$

where N_C is the number of vertex clusters shared between the graphs and m is the number of vertices, as above.

According to [6], the **total similarity** is :

$$S = \frac{(1 - L_S) + L_M + V_M + V_C}{4}, \qquad S \in [0, 1].$$

Note that the studied and produced graphs of this paper do not use weighted edges. Thus, to compute the total similarity we do not take into account the **edge strength similarity**, i.e.

$$S = \frac{L_M + V_M + V_C}{3}, \qquad S \in [0, 1].$$

Two identical graphs will have a total similarity of one, so the further it is from that (and the closer to zero) the less similar the graphs are.

3.2 Connectivity and communities

The *vertex connectivity* $\kappa(A)$ of a connected graph A refers to the least cardinality $\kappa(A) = |S|$ of a subset of the vertex set $S \subset V_A$ such that the new graph $\tilde{A} = (\tilde{V}_A, \tilde{E}_A, \tilde{\varepsilon}_A)$ (with $\tilde{V}_A = V_A \backslash S$) is either disconnected or trivial. Such set S is called a *minimum vertex cut*.

The *strong vertices* are the vertices with a maximum vertex connectivity.

Computation of the connectivity of a graph : Let $N(v_1)$ be the set of all adjacent vertices to v_1 (note that if $v_2 \in N(v_1)$, then we have $v_1 \in N(v_2)$). Then, for an arbitrary

vertex v_1, if $\exists S$ s.t. $v_1 \notin S$, i.e. there is a minimum vertex-cut that does not contain v_1 then, $\kappa(A) = \min\{\kappa(v_1, v_2) \mid v_1 \in V_A,\ v_2 \in V_A \backslash \{v_1\},$ and v_2 is not adjacent to $v_1\}$; if $v_1 \in S$, $\forall S$ then, $\kappa(A) = \min\{\kappa(v_2, v_3) \mid v_2, v_3 \in N(v_1),$ and v_2, v_1 are not adjacent$\}$.

We can compute the connectivity by the maximum flow algorithm proposed by A.H. Esfahanian [7, Algorithm 11].

The aim of computing the connectivity of a graph is to compare the strong vertices of different graphs. There are two questions to ask: Do both graphs have the same strong vertex? If they do, how different is their connectivity? If the difference between the connectivity of the strong vertex of each graph is close to zero, both graphs will be considered to have a similar structure.

In this paper we will focus on the communities, they are closely related to the strong vertices since two communities are separated by a strong vertex. They will be computed with the `Mathematica` function `FindGraphCommunities`.

3.3 Spectral distances

Following [8], the $i, j-$th component of an *adjacency matrix* is defined as

$$\mathbf{A}_{i,j} = \begin{cases} 1 & \text{if } i \sim j, \\ 0 & \text{otherwise.} \end{cases}$$

The fact that $A_{i,j} = \{0, 1\}$ means that there is at most one edge between two vertices. Thereby we can define the *degree* d_i of the vertex i as the number of edges connected to the i.

The *degree matrix* $\mathbf{D}$ is a diagonal matrix whose diagonal contains the degree of each vertex, $D_{i,j} = d_i$ if $i = j$ and $D_{i,j} = 0$ otherwise.

Spectrum of a matrix : The *spectrum of a matrix* is a sorted sequence of eigenvalues. Let us denote the i-th eigenvalue of the adjacency matrix by $\lambda_i^{\mathbf{A}}$, and the i-th eigenvalue of the Laplacian matrix by $\lambda_i^{\mathbf{L}}$

Adjacency and Laplacian spectral distance : Let $\lambda^{\mathbf{A}_A}$ and $\lambda^{\mathbf{A}_B}$ be the adjacency spectra of the graphs A and B and let $\lambda^{\mathbf{L}_A}$ and $\lambda^{\mathbf{L}_B}$ be their Laplacian spectra. Then, the **Adjacency spectral distance** is defined as

$$d_{\mathbf{A}}(A, B) = \left(\sum_{i=1}^{n} (\lambda_i^{\mathbf{A}_A} - \lambda_i^{\mathbf{A}_B})^2 \right)^{1/2}$$

The **Laplacian spectral distance** is defined equivalently, but using the Laplacian spectra.

In this paper, the Adjacency and Laplacian spectral distances will be computed as defined in the `Week8-ComparingGraphs2023-4` worksheet.

4 Graph Generation

To generate our RDF graph we chose to automatically link the maths modules to msc2020[9] codes. This allowed both a greater degree of objectivity and to include all modules on our graph rather than only those we have taken. Notice that each module descriptor has a similar style (except Group Project which we didn't classify as it is too varied). Thereby we were able to parse their PDF descriptors and only take the "Syllabus Plan".

The syllabus plan is then split into words which are classified according to the msc2020 code using Mathematica's built-in classifier trained on the entire subject classification (imported as a csv). One of problems we encountered was the use of pronouns and prepositions ('a', 'the', 'as', 'and', 'also', etc) as well as 'general' maths vocabulary ('mathematics', 'model', 'theorem', etc). As these words were used in several if not all syllabus plans, they can be omitted. To resolve this problem generally (rather than needing to create a list of ignored words) we used the probability that Mathematica assigns to its classifications. For each module we then summed up the probabilities for any repeated codes and ordered them in a decreasing order.

Let us consider the first 6 words of MTH1000 (Foundations) as an example :

1. We have the following list : {`Functions, logarithmic, exponential, trigonometric, hyperbolic, Partial`}
2. Then we use `Classifier` to convert the initial list into a list of tuples containing the msc2020 code assigned to the word and its probability : {{`26Axx, 0.423802`}, {`11Axx, 0.760049`}, {`34Axx, 0.282966`}, {`42Axx, 0.968698`}, {`35Axx, 0.994287`}, {`35Axx, 0.342177`}}
3. If the tuples have the same msc code (first entry) they are grouped together and their probabilities (second entry) are summed up. In this case only the `35Axx` elements are combined, resulting in the list: {{`35Axx, 1.336464`}, {`42Axx, 0.968698`}, {`11Axx, 0.760049`}, {`26Axx, 0.423802`}, {`34Axx, 0.282966`}}
4. We then normalise these values so that they sum to 1, ensuring consistency across module descriptors of different lengths: {{`35Axx, 0.354314`}, {`42Axx, 0.256814`}, {`11Axx, 0.201499`}, {`26Axx, 0.112355`}, {`34Axx, 0.0750179`}}
5. Finally, we connect the module code to the msc code if its probability is greater than the cutoff value of 0.05. In this example, all words are connected, but in general this averages out to about 5 connections per module, although it varies between 2 and 8.

4.1 Loading other graphs

In order to compare our graph to the ones generated manually, we import and merge all 5 of the public graphs shared to the forum (thanks to Ben White, Holly Barber, Lucie Johnson, Luke Garner and Donovan Tran). While these all do roughly follow our shared ontology (notably the relationship we want is always marked as `ex[uses]`), they use different urls to refer to MSC

codes so can not directly be combined. Most of these graphs also link to the fine-grained codes like `03D65` however there wasn't enough text in these to classify so we instead only link to the more general `03Axx` codes. Therefore to combine these graphs we first filter them to remove all non-`ex[used]` relations and use regex to convert all msc codes to a shared format.

The advantage of combining these graphs rather than comparing against any single one of them is that each graph only contains those modules the author has taken and so by combining them we get a more well-rounded view of the modules as a whole. Despite this there are still some modules none of these graphs classify such as `MTH1000`, and so we removed these nodes from our graph for most comparisons.

5 Comparison

When discussing generating graphs (section 4) we picked an arbitrary probability cutoff of 0.05 to define whether a particular edge is included in our final graph. Using the Adjacency and Laplacian spectral distances we can instead compare a variety of different cutoffs and plot them, showing whether our picked value is reasonable.

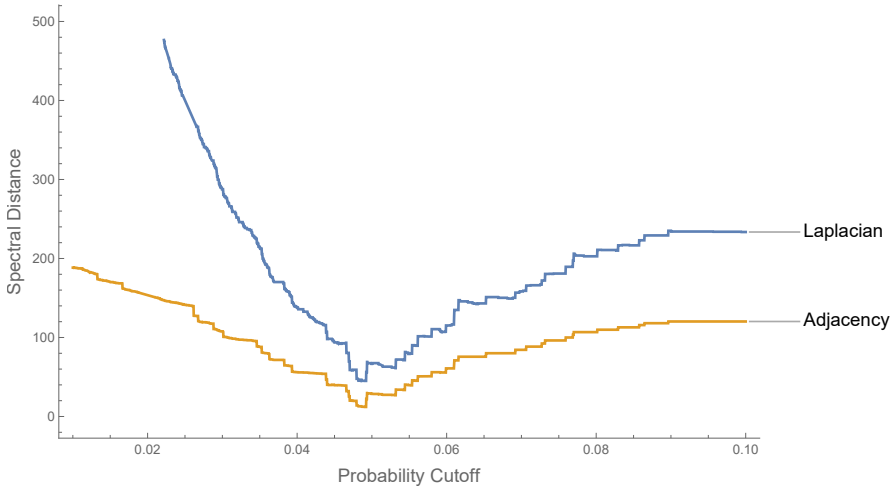
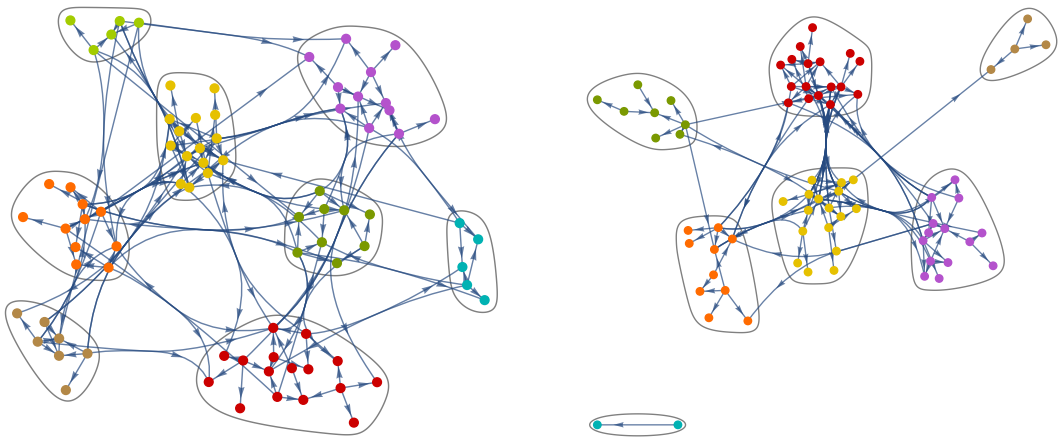


Figure 1: Variation of the Laplacian and Adjacency coefficients as a function of the cutoff value.

As we can see from Figure 1, our probability value of 0.05 is very close to the "best" value which — based on this test data — would be between 0.045 and 0.049. Notice also that both curves have a single minimum which occurs around the same cutoff value. This indicates that as we increase the cutoff from 0.05, most of the edges we remove are 'useful' to the score and as we decrease the cutoff most of the new ones we are adding are not useful. In this sense, the cutoff value is appropriate.

Let **RDF ours** denote our graph and **RDF other** denote the graph generated by loading the public graphs shared in the forum.

We compute their respective communities using mathematica's built in function `FindGraphCommunities` and obtain the graphs displayed in Figure 2.



(a) RDF ours (see A.2.1) (b) RDF other (see A.2.2)

Figure 2: Communities comparison

First, we note that both graphs don't have the same number of communities. In fact, **RDF ours** is divided into 8 communities (see A.2.1), whereas **RDF other** has 7 communities (see A.2.2) including one which is disconnected. Hence, we are comparing two graphs, the first one with 8 communities against the second with 6. We observe that in **RDF ours** the communities have a more varied range of mathematical topics, while **RDF other**'s correspond to more general yet more distinct mathematical contents. In addition, **RDF other**'s disjoint community corresponds to History of mathematics (MTH3019) which in **RDF ours** is not only part of a connected community (Light Green) but forms part of the strongly connected vertices.

We use the built-in mathematica function `KCoreComponents` to compute the groups of vertices that are strongly connected (see A.3). The common strong vertices to both graphs are MTH1001, MTH1002, MTH1003, MTH2003, MTH2006, MTH2009, MTH3022, MTH3024, 34-ODEs, 40-Sequences series summability, 41-Approximations and expansions. These vertices relate to calculus and analytical methods (e.g. Differential equations) and mathematical modeling. We interpret this result as an information on what are the modules and subjects that most students take and enjoy.

We remark also that **RDF ours** has much more strongly connected vertices (51 vertices against 34) but a maximal connectivity lower than **RDF other** (3 against 4). Thus we can conclude that **RDF ours** is more connected than **RDF other**.

Regarding similarity measurments (3.1), we obtain : $S = \frac{1417}{3210} \simeq 0.44$. This tells us that both graphs are not as similar as we would expect. We expected at larger value of similarity, since they are modelling the same content.

In Figure 3 we notice that the cutoff value of 0.05 is the value for which the similarity is at its maximum. This is consistent with the results illustrated in Figure 1.

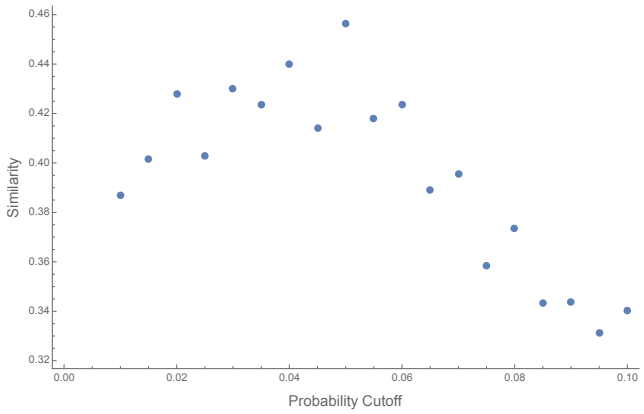


Figure 3: Variation of similarity coefficients as a function of the cutoff value.

6 Conclusion

To compare the "similarity" of **RDF ours** and **RDF other** we considered three methods : the similarity coefficient, the connectivity coefficients and analysis of communities, and the spectral distances between both graphs. The spectral distances allowed us to confirm our choice for the cutoff value to generate **RDF ours**. Even thought both graphs model the same courses, they have a low similarity : 0.44. We attribute this to the structural difference between the graph construction. Moreover, **RDF other** has a disconnected community which is not the case for **RDF ours**. We believe this is a consequence of the subjectivity when constructing **RDF other**: even thought **RDF other** was based on the academic experience of five students (not just one), the fact that the vertices are modules they selected and the mathematical content they associate to them makes the classification of mathematical knowledge more subjective.

References

1. Available from: `https://www.w3.org/` [Accessed on: 2024 Apr 24]
2. Available from: `https://www.w3.org/TR/rdf12-concepts/` [Accessed on: 2024 Apr 24]
3. Available from: `https://www.w3.org/TR/2012/REC-owl2-overview-20121211/` [Accessed on: 2024 Apr 24]
4. Available from: `https://datatracker.ietf.org/doc/html/rfc3987` [Accessed on: 2024 Apr 24]
5. Available from: `https://www.w3.org/TR/owl2-overview/` [Accessed on: 2024 May 02]
6. Brown NJK, Frazier CR, Cauthen KR, and Nozick LK. A Unique Graph Similarity Metric for Anomaly Detection. 2019 Apr. Available from: `https://www.osti.gov/biblio/1639686`
7. Esfahanian AH. Connectivity Algorithms. *Topics in structural graph theory*. Ed. by Lowell W. Beineke RJW. Cambridge University Press, 2013. Chap. 12:268–81. Available from: `https://www.cse.msu.edu/~esfahani/book_chapter/Graph_connectivity_chapter.pdf`

8. Wills P and Meyer FG. Metrics for graph comparison: A practitioner's guide. PLOS ONE 2020 Feb; 15. Ed. by Chen PY:e0228728. DOI: 10.1371/journal.pone.0228728. Available from: http://dx.doi.org/10.1371/journal.pone.0228728

9. Available from: https://cran.r-project.org/web/classifications/MSC-2010.html [Accessed on: 2024 May 02]

A Appendix

A.1 Code

All of the code and data files we used can be found at https://github.com/13ros27/mth3022-project, with the primary Mathematica project being mathematica/Project.nb.

A.2 Communities

A.2.1 Our Graph

Red MTH1001, MTH2008, MTH2009, MTH3006, MTH3022, MTH3040, 03-Mathematical logic and foundations (05-Combinatorics, 26-Real functions, 28- Measure and integration, 32-Several complex variables and analytic spaces, 40-Sequences, series, summability, 46-Functional analysis, 54-General topology, 70-Mechanics of particles and systems, 94-Information and communication theory, circuits).
Yellow MTH1000, MTH2003, MTH2005, MTH3004, MTH3026, MTH3039, (65-Numerical analysis, 97-Mathematics education, 11-Number Theory, 42-Harmonic analysis on Euclidean spaces, 34-ODEs, 35-PDEs, 68-Computer science, 16-Associative rings and algebras, 55-Algebraic topology)
Purple MTH1003, MTH2004, MTH3001, MTH3007 (31-Potential theory, 00-General and overarching topics; collections, 83-Relativity and gravitational theory, 39-Difference and functional equations, 76-Fluid mechanics, 91-Game theory, economics, finance, and other social and behavioral sciences, 80-Classical thermodynamics, heat transfe, 01-History and biography).
Orange MTH2006, MTH3030, MTH3011, MTH3045 (41-Approximations and expansions, 30-Functions of a complex variable, 81-Quantum theory, 82-Statistical mechanics, structure of matter, 93-Systems theory; control, 49-Calculus of variations and optimal control; optimization, 74-Mechanics of deformable solids).
Dark Green MTH1002, MTH2011, MTH3008, MTH3042 (45-Integral equations, 15-Linear and multilinear algebra; matrix theory, 44-Integral transforms, operational calculus, 47-Operator theory, 37-Dynamical systems and ergodic theory).
Gold MTH1004, MTH3024, MTH3041 (60-Probability theory and stochastic processes, 62-Statistics, 52-Convex and discrete geometry, 18-Category theory; homological algebra).
Teal MTH2010, MTH3038 (20-Group theory and generalizations, 13-Commutative algebra, 12-Field theory and polynomials).
Light Green MTH3013, MTH3019, MTH3028 (51-Geometry, 86-Geophysics).

A.2.2 Other Graph

Red MTH1004, MTH2006, MTH3022, MTH3024, MTH3028 (03-Mathematical logic and foundations, 05-Combinatorics, 15-Linear and multilinear algebra, 41-Approximations and expantions, 60-Probability theory and stochastic processes, 62-Statistics, 68-Computer science, 82-structure of matter, 97-Mathematics education).
Yellow MTH1002, MTH2003, MTH2008, MTH3030, MTH2009, MTH3040 (26-Real functions, 30-Functions of a complex variable, 32-Several complex variables and analytic spaces, 33-Special functions, 34-ODEs, 40-Sequences, series, summability, 42-Harmonic analysis on Euclidean spaces, 51-Geometry, 54-General topology, 86-Geophysics, 91-Game theory, economics, finance, and other social and behavioral sciences).
Purple MTH1001, MTH2010, MTH2011, MTH3004, MTH3026, MTH3038, (06-ordered algebraic structures, 08-General algebraic systems, 11-Number theory, 12-Field theory and polynomials, 13-Commutative algebra, 14-Algebraic geometry , 20-Group theory and generalizations, 94-Information and communication theory, circuits)
Orange MTH1003, MTH3006, MTH2005, (00-General and overarching topics; collections, 35-PDEs, 37-Dynamical systems and ergodic theory, 46-Functional analysis, 65-Numerical analysis, 83-Relativity and gravitational theory, 92-Biology and other natural sciences).
Green MTH2004, MTH3007, MTH3001, (53-Differential geometry, 74-Mechanics of deformable solids, 76-Fluid mechanics, 78-Optics, electromagnetic theory, 80-Classical thermodynamics, heat transfer).
Brown MTH3042, (44-Integral transforms, operational calculus, 45-Integral equations, 47-Operator theory).
Cyan MTH3019, (01-History and biography).

A.3 Strongly Connected vertices

A.3.1 Our Graph

Maximal connectivity 3;

List of strongly connected vertices MTH1000, MTH1001, MTH1002, MTH1003, MTH1004, MTH2003, MTH2004, MTH2005, MTH2006, MTH2009, MTH2010, MTH3001, MTH3004, MTH3007, MTH3008, MTH3011, MTH3013, MTH3019, MTH3022, MTH3024, MTH3026, MTH3028, MTH3030, MTH3038, MTH3039, MTH3041, MTH3042, MTH3045, 01, 11, 12, 20, 30 31, 32, 34, 35, 39, 40, 41, 44, 45, 47, 51, 60, 62, 65, 68, 76, 91, 94.

A.3.2 Other Graph

Maximal connectivity 4;

List of strongly connected vertices MTH1001, MTH1002, MTH1003, MTH2003, MTH2006, MTH2009, MTH3022, MTH3024, 03, 15, 26, 34, 40, 41, 97.