Finding Deeper Community Structures: The Networks of Portuguese Universities

Bernardo Silva[a,b] **and Maria Costa**[a,c]

[a]Técnico Lisboa; [b]MSc. in Electrical and Computer Engineering; [c]MSc. in Data Science Engineering

November 9, 2021

Community finding is a complex problem that is relevant to many different areas, as these partitions usually contain information on the network's structure. One common drawback is the lack of information on that aspect on most networks, leading to a difficulty in the interpretation of quality of the results. We start by obtaining real world networks with a clear partition known beforehand, the networks of Portuguese University applicants through several years. We then compare algorithms designed for the purpose of community finding on these networks, allowing us to evaluate their performance quantitatively and qualitatively. We also analyse the resolution problem associated with the algorithms used, by taking two different approaches. Our results allow us to have a better understanding of how the theoretical models perform in real life , and which should be picked when applying to a network with unknown portioning. The use of a network analysis approach to the study of the college applications in Portugal could also provide a better understanding of the dynamics of the supply and demand of courses versus candidates, as well as tendencies regarding the minimum grade for each field of study.

It has been shown that nowadays most systems can be described as networks, i.e., assemblies of nodes and edges containing information about the structure, with examples ranging from biological all the way to social phenomena. In this article we set to analyze of its topological characteristics, the surge of communities.

Community finding has been a widely discussed topic, with many research works being dedicated to it due to the its usefulness in the most diverse areas. The concept of a community in a network is nothing but a group of nodes more strongly connected with which other than with the "outside", and its often linked with classification of objects in categories for the sake of retrieval of information.

Quantitatively speaking this is a fairly simple concept to grasp and there have been a significant amount of algorithms designed towards it. Given all that, our goal here is to analyse some of them as well as their meaning given real world networks, while establishing some comparisons between the results obtained.

The networks we chose to gather consist of college applications in Portugal, in the years 2018 to 2021, with nodes being the courses available to the applicants and the link between a pair of them the simultaneity in applications by a given applicant (more detail on this process in Methods section). Besides the general interest in studying these networks, they will be a fundamental tool in understanding the methods of finding structures within communities. This is because we can, *a priori*, divide the networks according to several attributes (fields of study and location, for example), which is not very common in network analysis.

Background

A community can be qualitatively defined as a subset of nodes of a graph with denser connections between them than with the rest of network in hand. The process of finding a community usually involves identifying its inner hierarchical structure, by mapping the nodes and edges into a tree. The algorithms here explored use a particular parameter of a network, modularity, to find that structure, and involve the maximization of said parameter. These types of algorithms, called agglomerative, start with all nodes as communities and progress by joining them if that leads to a increase of the parameter (1).

The modularity of a network's partition is usually given by the following sum over all n communities (2)

$$Q = \sum_{s=1}^{n} \left[\frac{w_{ss}}{w} - \left(\frac{w_s}{2w} \right)^2 \right] \quad , \qquad [1]$$

which is equivalent to

$$Q = \frac{1}{2w} \sum_{ij} \left(w_{ij} - \frac{w_i w_j}{2w} \right) \delta(C_i, C_j) \quad , \qquad [2]$$

where w_{ij} represents the weight of the edge between nodes i and j (adjacency matrix), w_i the degree of node i and w the total degree of the network. The function $\delta(C_i, C_j)$ is 1 when both nodes belong to the same community and 0 otherwise. This metric is often used as a way to find communities within a network. Maximizing the modularity means finding the n communities with the most inter-community edges and the fewest intra-community edges.

Despite leading to good results, there is a resolution problem with this method (3), that is, this method does not allow for the finding of smaller sub-communities or even bigger ones.

Two solutions for this problem have been proposed. The first, by Reichardt and Bornhold (4), consisting in the addition of a resolution parameter, r, in the formula for the modularity (R&B resolution parameter),

$$Q_{\mathrm{R\&B}} = \sum_{s=1}^{n} \left[\frac{w_{ss}}{w} - r \left(\frac{w_s}{2w} \right)^2 \right] \quad . \qquad [3]$$

This method corresponds to varying the relative weight of the comparison between the network and the random graph.

The other method was proposed by Arenas *et al.* (5), consisting in the addition of a self-loop at every node with a given weight r (Arenas resolution parameter), that is, the adjacency matrix of the network becomes

$$W' = W + rI \quad , \qquad [4]$$

which corresponds to a shift of the degrees of the nodes by $2r$.

Methods

Building the Network. The data used to build the networks was obtained through Web scrapping, using *Python*'s *request* library, to access the data publicly available at DGES.

For each year, from 2018 to 2021, it was created a list of all the faculties of the several Portuguese universities and respective codes (there is a unique code associated with each faculty). With this information, it was then created a list of courses per faculty and respective codes (course code). Finally, accessing the list of all courses' candidates, and cross-checking between every pair of courses, a link was added between the pairs if a common candidate was found. The weight of these links was set to represent the number of common candidates. The candidates' data was deleted after this process to ensure anonymity.

Furthermore, a similar method was applied to obtain some statistics relating to each course, such as the number of available and filled slots, and the minimum grade to enroll the course (corresponding to the grade of the last student to enroll).

Classifying by Fields of Study. Using the official 2021 list of courses per field of study, each course was classified. Since some of the official fields of study were too narrow (total of 23 fields), we merged a few of them to create broader areas, namely: Education; Arts and Architecture; Human Sciences; Social and Behavioral Sciences, and Law; Business Sciences; Life Sciences; Exact Sciences; Social and Personal Services; Health.

This classification was achieved by comparing each course's name with the names on the list, finding the most similar and picking its area. Due to the unavailability of such a list for each year, and the fact that some courses went through name changes, this classification might not have been totally accurate. This does not affect the analysis of fields of study in a noticeable degree.

Community Finding.

Greedy Modularity Maximization. Communities were found, using Clauset-Newman-Moore greedy modularity maximization (1), implemented in the `networkx` library. The algorithm first considers all nodes to be communities and maps the ΔQ for the merge of each pair, choosing the largest of them and proceeding to it. In every iteration, it combines communities in a way that maximizes modularity, until only one community remains.

Louvain Communities. Communities were found, using the Louvain modularity maximization (6), implemented in the `python-louvain` library. This algorithm also starts with each node as a community and computes the modularity gain of merging adjacent nodes, choosing the higher value. Once a maximum is reached, the same is applied to the new formed network, with communities being the nodes; the process uses modularity as its stopping criteria, it only stops when there's no increase in its value.

Matching Communities in Different Years. After finding communities on the different years, it is useful to display them side by side, trying to show which community represents the "same" group of courses in different years. For that, the year with the most communities was picked as the baseline (suppose m communities). These communities where then numbered from 1 to m.

For all other years, it was evaluated the similarity between each community and the baseline communities. The similarity score between communities A and B is computed by:

$$S = \frac{\#\{\text{common courses}\}^2}{1 + \#\{\text{unique courses}\}} \quad . \qquad [5]$$

So, for a year with $n \leq m$ communities, S yields a $n \times m$ matrix of scores. We then try to number the n communities with numbers between 1 and m to maximize the sum of the scores, that is, for each row we must pick the index that maximizes the score, without repeating indexes. For that, we found the maximum element of the matrix, say S_{ij}, and number community i with index j. Then, we set row i and column j to a negative number to ensure that that community or index are not picked again. The process is repeated until all n communities are numbered. Communities in different years with the same numbering should represent approximately the same groups.

Results and Discussion

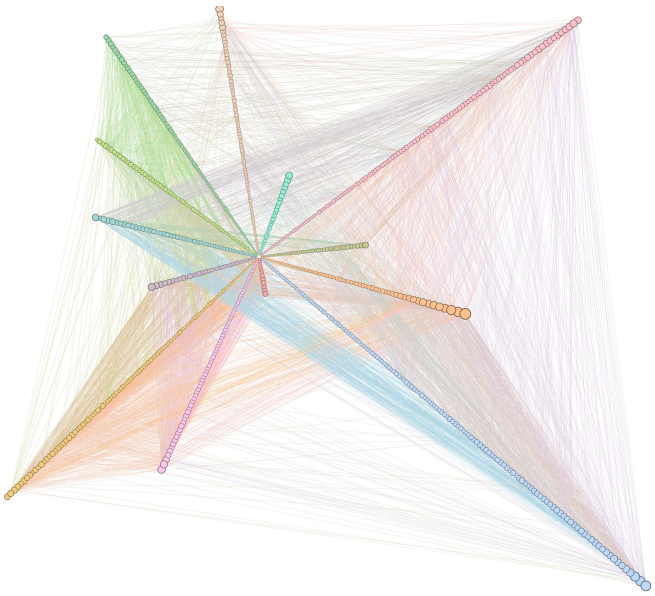

Fig. 1. Network for 2021, with edge filter for weights above 10. The colors represent different communities.

Network characterization. Given the networks created as described in , regarding the years of 2018 to 2021, some measures were taken to characterize them. The table that follows contains some relevant information:

Table 1. Some metrics of each year's network

Year	N	E	$\langle k \rangle$	w	# Candidates
2018	1063	77248	957.87	506771	49362
2019	1056	76547	962.98	506703	51036
2020	1063	81027	1214.81	644401	62561
2021	1065	81032	1243.28	660132	63878

To be noted the high value of the network's average degree, $\langle k \rangle$, placing these in the connected regime, as $\langle k \rangle \gg \ln(N)$ for every year.

To further analyse the networks obtained, it's useful to plot the cumulative probability degree distribution, to which we proceeded.

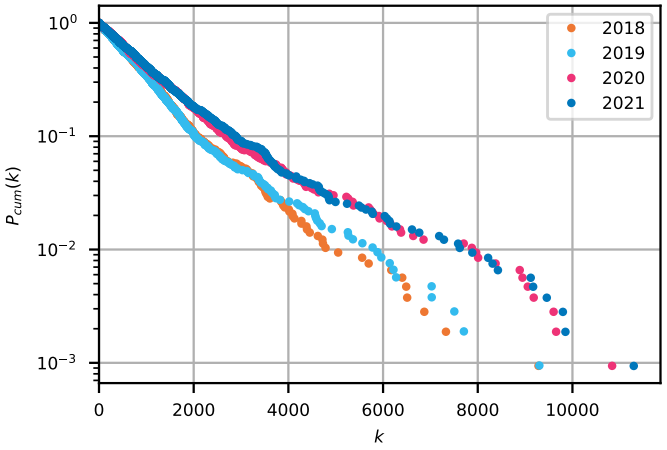

Fig. 2. Degree distribution for years 2018 to 2021

Firstly, we can observe the similarity that all four years share regarding the distribution's shape, to be expected as all describe the same phenomena. In addition to that, it's worth noting the difference in the latest two years for larger values of k, showing higher probabilities, in agreement with the $\langle k \rangle$ values for these networks.

R&B Resolution Parameter.

Greedy Modularity Maximization. We start by analysing the effect of varying the R&B resolution parameter using the Clauset-Newman-Moore greedy modularity maximization algorithm (1) to find the communities. Considering the parameter varying from 0 to 30, we see that the number of communities varies approximately linearly, and a linear regression yields a slope of around 8 and an interception at around -5 for all years. The detailed results can be checked on table 2.

Fig. 3. Greedy Modularity Maximization: Number of communities found in function of the R&B resolution parameter. It appears to follow a linear growth.

Despite showing values of r^2 close to 1, the value of the intercept is different from 1, which we would expect with a resolution parameter of 0. This fact will be discussed further when we look at the asymptotic behaviour of the method.

Table 2. Results of the linear regressions of the size of the communities found in function of the resolution parameter

Year	Slope	Intercept	r^2
2018	8.867 ± 0.007	-9 ± 1	0.9978
2019	8.92 ± 0.09	-9 ± 2	0.9967
2020	8.34 ± 0.08	-7 ± 1	0.9974
2021	8.28 ± 0.08	-7 ± 1	0.9973

The average size of each community, as expected, varies with the inverse of the number of communities. These results allow us to get a first insight on how varying the parameter can be useful for finding the underlying structures of the network at different scales.

For a more qualitative analysis we resort to the know divisions of the networks and see how well the algorithm performs at finding them. Using attributes associated with each course, we might get a better understanding of what composes a community. The most obvious one is the field of study corresponding to each course, two courses in the same area will be attractive to the same person. Using the communities previously obtained, counting the courses belonging to each area, and then trying to match all four years' communities, we can understand not only how the communities are structured each year, but also how they evolved in time. Figure 4 shows the results for five different values of the resolution parameter. A set of four bars represents the communities of the four years that matched, and every bar is divided by the fraction of elements in each field, ordered from the greatest to the smallest fraction.

Starting with the null resolution parameter, the algorithm results in every node belonging to the same community. This is useful for showing the division of the networks in fields of study.

For a resolution parameter of 0.2, we get only two communities with very small overlap in fields of study (the elements of common fields of study represent less than 5% of the community size). This division in two is very close to what we would obtain if we were to divide the fields in a subjective manner.

The standard choice or resolution parameter equal to one, although displaying a well defined division in fields of study, would not be in itself enough to explain the full structure of the network, emphasizing this idea that there exist deeper structures.

Looking at parameter 1.5, we get (excluding the smallest communities) nine well defined sets of communities. This coincides with the number of fields of study chosen *a priori* to classify the courses that, despite being a common way of dividing courses, it is also an arbitrary choice. The structure of this graph indicates that maybe theses boundaries do not exactly represent the divisions in interests of future students.

We also tried to find a correlation between communities and the university of the course, without success, as the correlation is not as strong as when using the fields of study.

Louvain Modularity Maximization. The other modularity maximization method used was the Louvain Modularity Maximization (6). For the same range of resolution parameter, this method shows the same linear behaviour in the number of

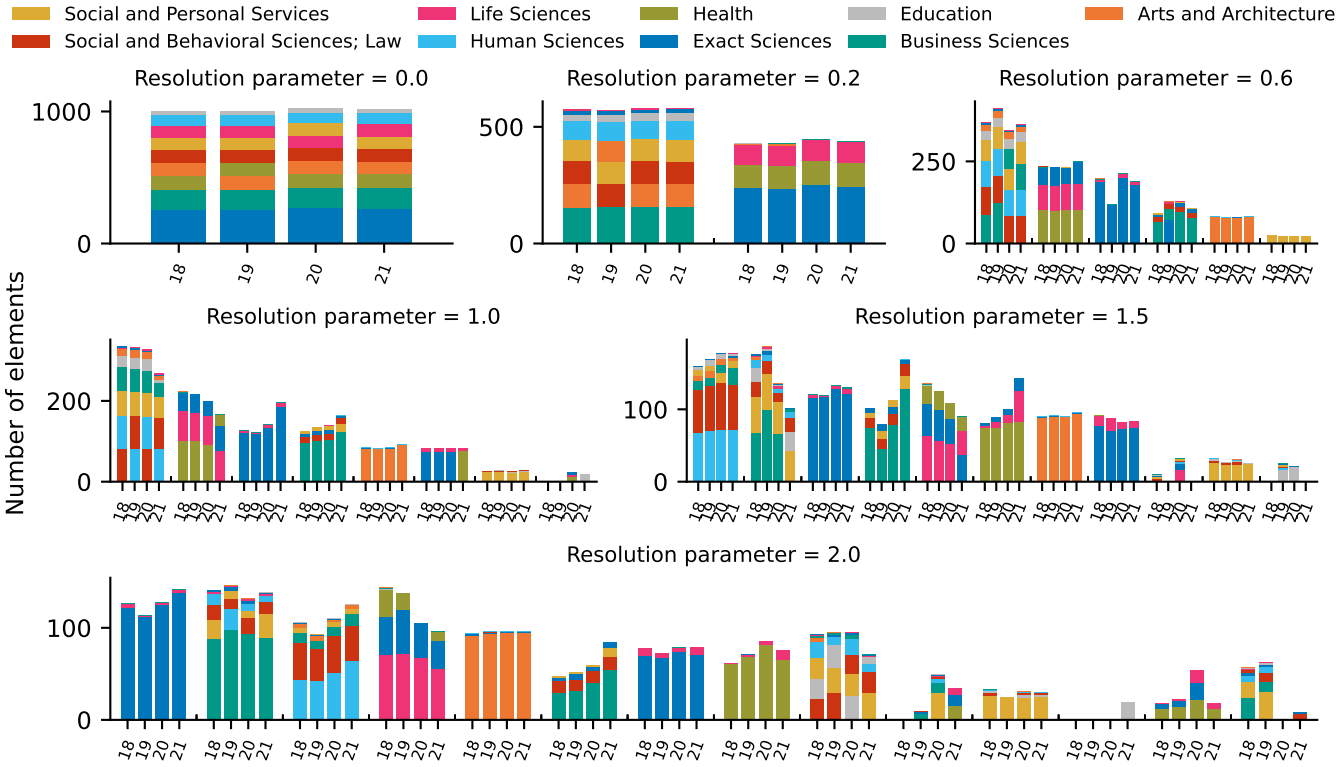

Fig. 4. Greedy Modularity Maximization: Several partitions of the networks divided by fields of study. Each bar represents a community found in a given year, and groups of four bars should represent the same community through the years. Each bar is divided in the fractions representing each field of study.

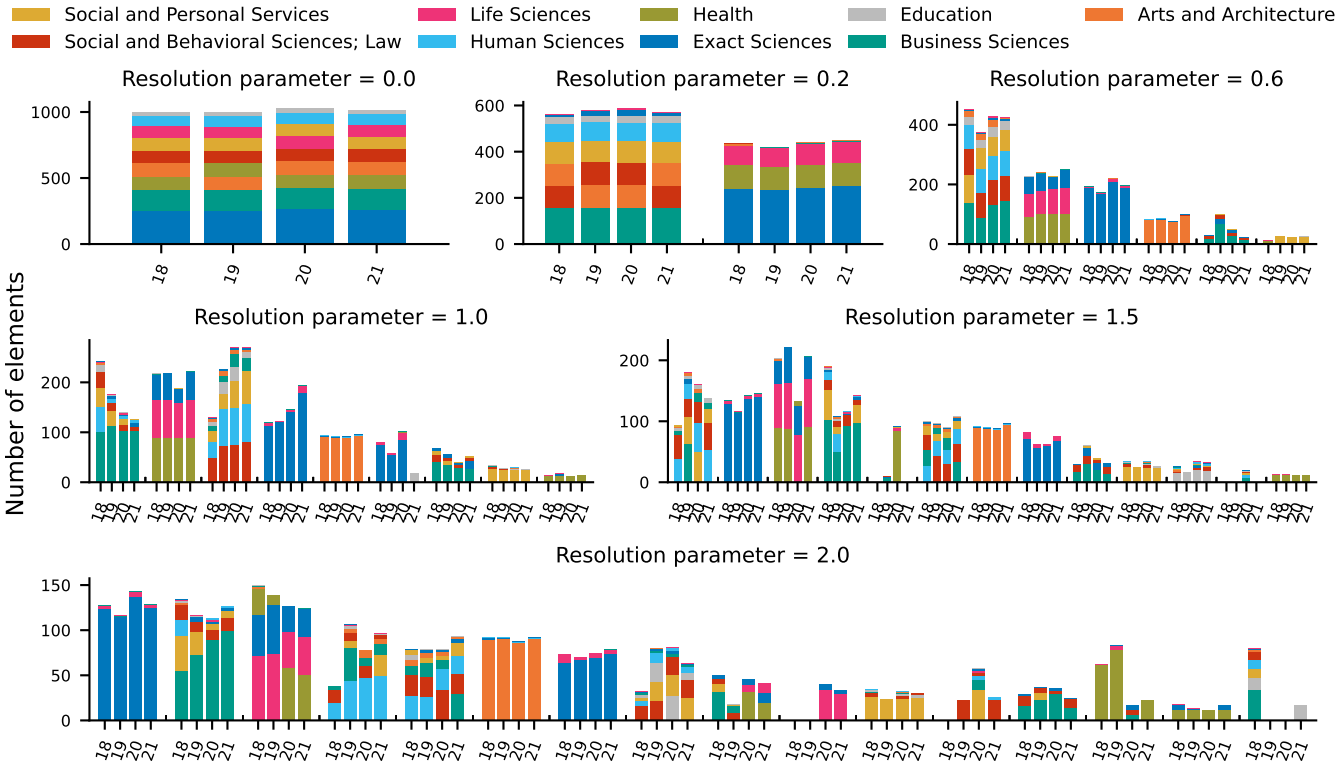

Fig. 5. Louvain Modularity Maximization: Several partitions of the networks divided by fields of study. Each bar represents a community found in a given year, and groups of four bars should represent the same community through the years. Each bar is divided in the fractions representing each field of study.

communities generate. This is plotted in figure 6. Repeating

Fig. 6. Louvain Modularity Maximization: Number of communities found in function of the R&B resolution parameter. Once again, we see the linear behaviour.

the linear regressions, we get the results shown on table 3.

Table 3. Results of the linear regressions of the size of the communities found in function of the resolution parameter

Year	Slope	Intercept	r^2
2018	8.67 ± 0.08	-6 ± 1	0.9974
2019	8.67 ± 0.09	-6 ± 1	0.9963
2020	8.13 ± 0.08	-4 ± 1	0.9972
2021	8.07 ± 0.08	-4 ± 1	0.9970

The values are similar with the ones obtained before, being the intercepts slightly closer to the expected value, but still not completely right.

Once again, we represent some of the resulting partitions of the networks classified by field of study, in figure 5

Comparative Analysis. Given that both algorithms work towards the maximization of the same parameter, modularity, as previously discussed, but taking different routes, we found it useful to make some comparisons between the two. For this we used the network relative to the year of 2021.

In terms of performance, the Louvain algorithm had an inferior time of execution, an important aspect on networks as connected as the ones analysed here.

In terms of community size and number the graphs in figure 7 show how these evolved with the resolution parameter ranging from 0.2 to 25

One can start by observing the similarity between both methods, either when it comes to the number of communities as well as the average community size, emphasizing how alike these algorithms perform. The seemingly big difference in the average community size for the first value is only due to the fact that the Greedy Modularity algorithm finds two communities while Louvain finds three.

When it comes to the modularity score associated with these partitions, we can calculate it using the same parameter used during the partitioning algorithm. We show this comparison in figure 8.

If we then look at the difference between the two curves, in figure 9, we find that Louvain performs better, even though it is with a small margin, and as expected, as we increase the resolution parameter the two get closer.

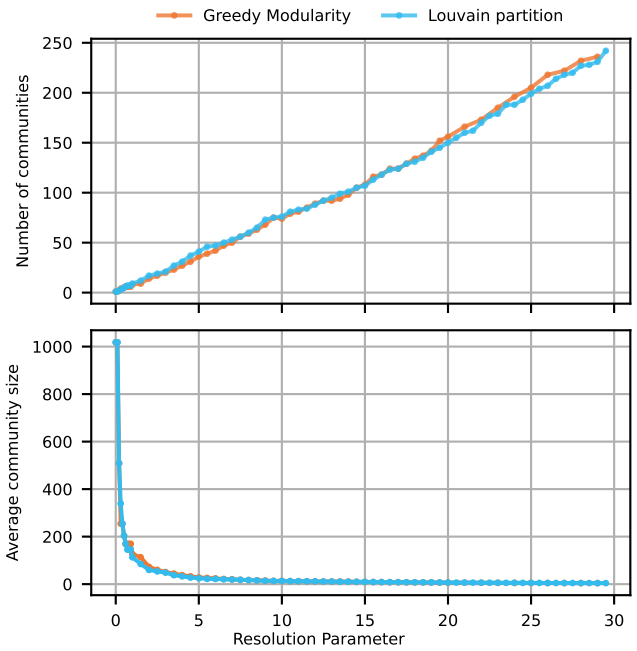

Fig. 7. Number of communities and average size for the Greedy Modularity Maximization and the Louvain Modularity Maximization

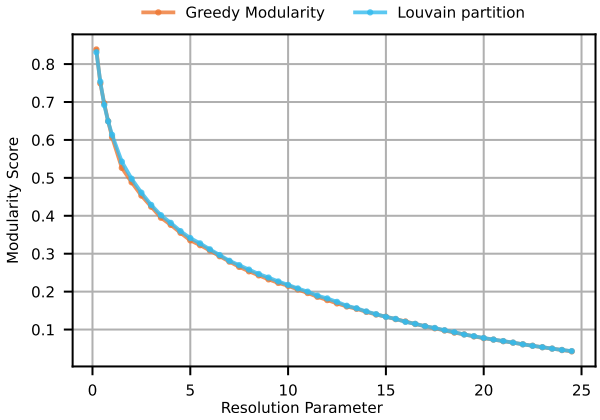

Fig. 8. Modularity score using the respective resolution parameter

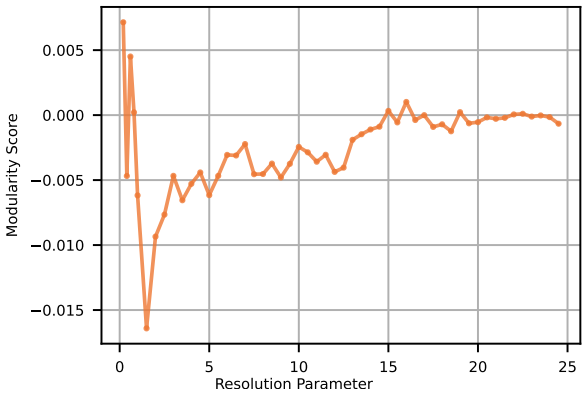

Fig. 9. Difference between greedy modularity and Louvain modularity scores

Arenas Resolution Parameter. The other method proposed for revealing the different structures of the networks relies on adding self-loops with a given weight. This weight is by default 0, so this is the baseline. For obtaining bigger communities, we must pick a negative parameter, and a positive one results in smaller communities.

In this section, we resort to the Greedy Modularity Maximization algorithm since the Louvain method implementation used is incompatible with negative node degrees (which is a consequence of a negative enough resolution parameter).

With this method, there is a negative lower bound which represents the value of the parameter after which all nodes belong to the same community. This lower bound is given by $r_{\min} = -\frac{2w}{N}$ (5). For our networks, this yields -506, -507, -628 and -648 from the first to last year.

To reach a number of communities similar to the one obtained before of around 300, the parameter must reach a value of around 5000. We show the number of communities in function of the resolution parameter on figure 10.

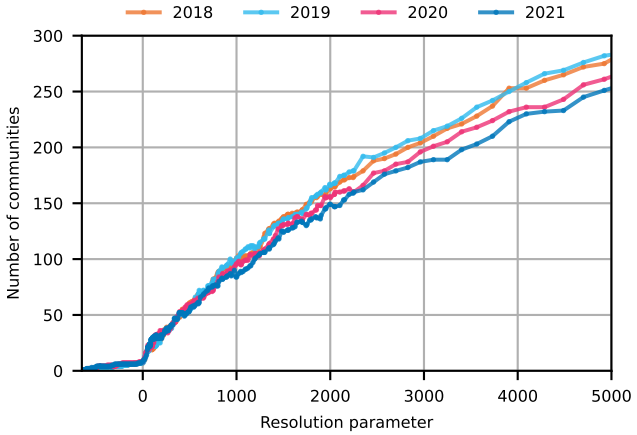

Fig. 10. Arenas Resolution Parameter: Number of communities in function of the Arenas parameter.

We show a more detailed view of how the number of communities grows from 1, for negative values of the parameter, in figure 11.

We see that the number of communities grows slowly from 1, at $r_{\min}$, to the numbers previously obtained, with parameter 0, of 7 and 8 communities for the first and last two years, respectively. With a parameter greater than 0, the number of communities grows slowly when compared to the R&B parameter. We see that the obtained values of $r_{\min}$ do not coincide precisely with what was predicted, being greater by about 50.

Because the effect of the parameter is adding a self-loop at every node with the parameter as the weight, its effect then depends on the scale of the weights of the networks' edges. Our networks show an average degree in the order of a thousand, so this turns out to be the scale of the parameter. We see here a first disadvantage of this method, being less general and predictable, as the linear behaviour is less predominant.

Other important fact for community finding is how uniform are the communities found. We can have an idea of this by analysing the standard deviation of the size of the communities, a big standard deviation will mean that there are communities

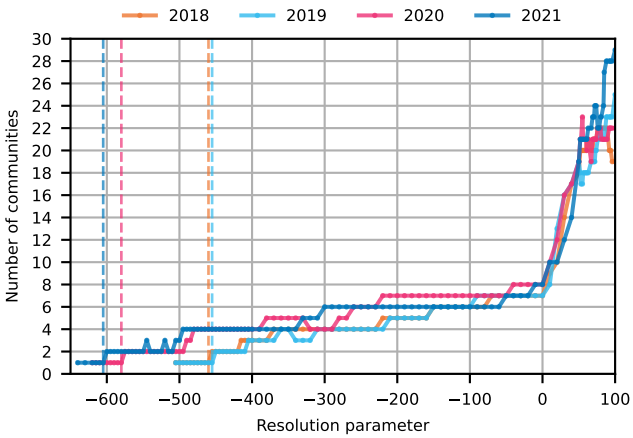

Fig. 11. Arenas Resolution Parameter: Number of communities for negative and small parameter. The dashed lines represent the values obtained for $r_{\min}$ and the leftmost point represents its expected value.

much bigger than the smaller ones. So, to compare the two resolution parameters, we plot the standard deviation versus the number of communities found (figure 12).

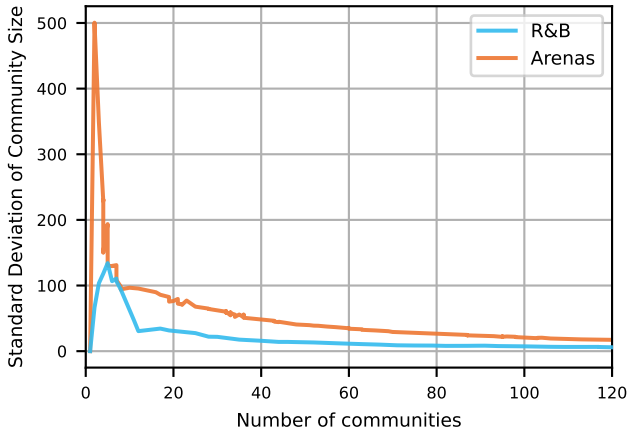

Fig. 12. Comparison of the standard deviations of the communities sizes obtained with each method.

Varying the Arenas resolution parameter generates less uniform communities when considering a small number of communities.

Asymptotic Behaviour. We have already shown how the limit of the methods where the result is every node being in the same community. It is also of interest to study how the other limit, every node in its own community, is reached.

For the R&B resolution parameter, considering equation 1, we see that all nodes will be isolated when $\delta(C_i, C_j) = 0, \forall i \neq j$, which happens when $w_{ij} < \gamma w_i w_j / 2w$, that is, for $\gamma > 2ww_{ij}/w_i w_j$. The right-hand side is maximum for two connected nodes, each with a small degree. Thus, the last two nodes to remain in a community when using this methods are the two most isolated but also connected between themselves.

Applying the method for increasingly values of the resolution parameter lead us to the expected results, being the

value after which all nodes isolated the same as predicted theoretically. Table 4 shows these results.

Table 4. R&B resolution parameter: Last two nodes in the same community, for $r = r_{\max} - 1$, with the respective degrees and weight of the edge connecting them. These are the nodes that are connected and present the smallest product of degrees. More information about these courses can be found here.

Year	$r_{\max}$	$2ww_{ij}/w_iw_j$	Course i	w_i	Course j	w_j	w_ij
2018	7508	7507.718	1307_A011	30	1307_A010	9	2
2019	19303	19302.971	3062_L163	5	3131_9878	21	2
2020	4439	4157.426	3141_L169	62	3141_8419	5	1
2021	3171	3070.38	3152_9123	215	3154_9090	2	1

Plotting the number of communities up to the values of $r_{\max}$, figure 13, shows that the linear growth previously found was only local, and does not hold for all resolution parameters. The negative values of the intercept that were found can be explained by this fact, as we might have considered a too wide interval of parameters to approximate as a line. The further we increase the resolution parameter, the slower the growth of the number of communities becomes, up to the point where all nodes are isolated.

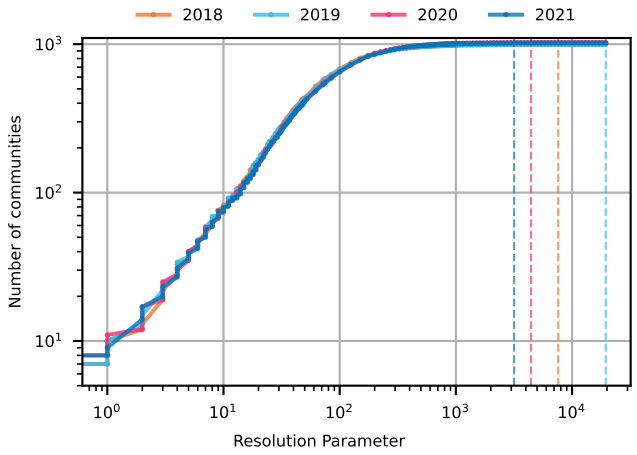

Fig. 13. R&B resolution parameter: Asymptotic behaviour of the number of communities. The dashed lines represent the value of r after which all nodes are isolated.

Considering the other method of finding structures, Arenas *et al.* presents a set of inequalities (one for every pair of nodes) whose minimum value of r that satisfies them corresponds to the value of $r_{\max}$, that is, the value after which all nodes are isolated [*]. We took an experimental approach and ran the algorithm until the value of $r_{\max}$ was found. Here, the last two nodes that remain in the same community, in contrast with the previous method, are the ones which are the most connected between themselves.

[*] (5) considers that a self-loop of weight r increases the node's degree by r. We chose to consider that the degree is increased by $2r$, and so the parameter obtained with the inequalities should be halved to obtain our corresponding parameter

Table 5. Arenas resolution parameter: Last two nodes in the same, for $r = r_{\max} - 1$, with the respective degrees and weight of the edge connecting them. These are the nodes connected by the edge with the greatest weight. More information about these courses can be found here.

Year	$r_{\max}$	Course i	w_i	Course j	w_j	w_{ij}
2018	592 672	0901_9813	5609	1507_9813	6428	1197
2019	593 102	1504_9078	6804	0911_9078	4980	1200
2020	848 585	0901_9813	8156	1507_9813	9938	1674
2021	860 300	0901_9813	8219	1507_9813	10395	1709

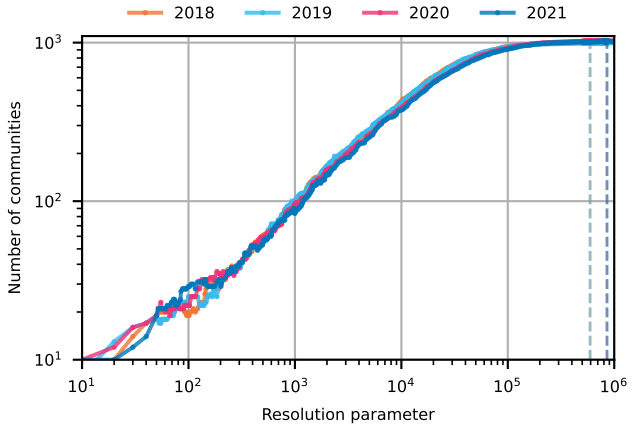

Fig. 14. Arenas resolution parameter: Asymptotic behaviour of the number of communities. The dashed lines represent the value of r after which all nodes are isolated.

Conclusion

The networks considered showed to be very interesting, as the resulting partitions matched very well with the divisions expected considering the fields of study. The algorithms used were able to identify, given different resolution parameters, communities in conformity with the proposed classification, which also shared a lot of similarities between the four years analysed.

When considering the two algorithms for modularity maximization (Greedy and Louvain), we found that the results are very similar, both in terms of number of communities as well as average community size, being the implementation used for the Louvain algorithm the fastest of the two. When computed the difference in the modularity score of the two, we found Louvain to have a marginal advantage over Greedy. In addition to this, it's worth noting that the Louvain algorithm performed with smaller time of execution.

After analysing the two ways proposed of solving the resolution problem intrinsic to community finding via modularity maximization, we conclude that the R&B parameter yields the best results. Namely, it is easier to control to number of expected communities due to the near linear dependence at low values of the parameter. We also concluded that the communities found by this method seem to be more uniform in size. As for the asymptotic behaviour, the Arenas method kept together the strongest connections in the network, which might be seen as more intuitive than keeping most isolate nodes in the same community. This would seem like an advantage for

this method, but, since this is an asymptotic behaviour, there is no real use case for this, as there should be no interest in finding a number of communities near the size of the network.

For further study of these communities, if other year's data was made available, it would be interesting to study the dynamics of these networks, of how the courses are grouped and if we could use attributes such as the minimum grade of the course or the number of candidates as a metric of the attractiveness of the nodes. This approach could also prove useful in the elaboration of a common strategy for fulfilling the needs and demands of college courses.

1. A Clauset, ME Newman, C Moore, Finding community structure in very large networks. *Phys. review E* **70**, 066111 (2004).
2. MEJ Newman, *Networks: an introduction.* (Oxford Univ. Press), (2011).
3. S Fortunato, M Barthélemy, Resolution limit in community detection. *Proc. Natl. Acad. Sci.* **104**, 36–41 (2007).
4. J Reichardt, S Bornholdt, Statistical mechanics of community detection. *Phys. Rev. E* **74**, 016110 (2006).
5. A Arenas, A Fernández, S Gómez, Analysis of the structure of complex networks at different resolution levels. *New J. Phys.* **10**, 053039 (2008).
6. R Lambiotte, JC Delvenne, M Barahona, Random walks, markov processes and the multiscale modular organization of complex networks. *IEEE Transactions on Netw. Sci. Eng.* **1**, 76–90 (2014).
7. R Lambiotte, JC Delvenne, M Barahona, Comparative analysis of community detection algorithms. *2017 Conf. on Inf. Commun. Technol. (CICT)* **1** (2017).