Network models development and community detection in identifying similar socioeconomic profiles

Rodrigo de A. Porto[1], Rafael de M. D. Frinhani[1], Felipe R. M. Mendes[1], Vanessa C. O. de Souza[1]

Universidade Federal de Itajubá (UNIFEI), Brazil
rodrigoandradeporto@hotmail.com; {frinhani, felipe.mendes, vanessasouza}@unifei.edu.br

Abstract. The growing demand for scholarships in higher education and the limited financial resources make it challenging to select candidates with the most significant socioeconomic vulnerability. To assist in this task, we developed three graph-based models to represent the network of candidates for student aid. We applied the Community Detection methods *Fast Greedy*, *Multilevel*, and *Walktrap* to the networks generated by each model to identify groups of candidates with the most significant similarity in socioeconomic characteristics. Computational experiments were performed with real data from three years of aid applications from a Brazilian federal university to validate the models. We analyzed the generated solutions for cohesion and external isolation between communities based on metrics such as modularity and the number of edges between communities. Model 2 (Multirelational) obtained the best results in the three datasets analyzed, with the *Fast Greedy* method delivering more cohesive communities than the others. We noticed that the Community Detection method greatly influences the obtaining of better quality solutions, which in some cases reduced the inter-community edges by about eight times in cases with the same number of communities. The work brings contributions by presenting three network models for academic management, a context that still needs to be explored in the literature, which, together with Community Detection methods, can potentially classify data in an unsupervised manner.

CCS Concepts: • **Information systems** → **Clustering**; • **Computing methodologies** → *Network science*.

Keywords: Academic Management, Community Detection, Network Models, Student Aid, Unsupervised Learning.

1. INTRODUCTION

Student assistance programs in Brazilian federal universities provide scholarships to students facing financial or academic challenges. A dedicated administrative unit, the Student Affairs Directorate (SAD), is responsible for managing aid and assessing applications at the institution where this study was conducted. Currently, this process is entirely manual and performed on a case-by-case basis. However, the increasing number of applicants and limited financial resources have made individual evaluation a complex and time-intensive task, hindering the effective prioritization of candidates who best meet the eligibility criteria. In this scenario, computational methods can obtain a prior classification of applicants to assist professionals' decision-making. Such methods can identify patterns in the socioeconomic characteristics of applicants, enabling their comparison based on a measure of similarity. Several works have addressed the identification of profiles, for example, those based on statistical methods, heuristics, or unsupervised learning [Hamim et al. 2021] [Yang et al. 2004], but few have addressed the topic in the context of student assistance.

Given this context, this study aims to model and explore graph-based computational methods to classify candidates for student assistance according to their socioeconomic characteristics. Three similarity models between candidates were represented as graphs and submitted to Community Detection algorithms to support prioritization in the selection process and decision-making by the University's social workers.

Experiments with real data from three academic years highlighted the Multirelational Model as the most effective, especially when combined with the *Fast Greedy* Community Detection algorithm. These results demonstrate the approach's potential as a decision-support tool for identifying groups of applicants with similar socioeconomic profiles and streamlining student aid allocation decisions. The models' application can be extended to other areas that need profile identification.

This article is organized as follows: Section 2 covers concepts related to profile identification, graphs, social network modeling, and Community Detection methods. Section 3 details the research method, models developed, and computational experiments carried out. Section 4 contains the results obtained, and finally, Section 5 presents conclusions and suggestions for future work.

2. THEORETICAL REFERENCE

Profile identification helps to understand behaviors, such as consumer preferences, in criminal investigations and service demand, which leads to the development of methods that use images or behavioral data. For example, the use of *Genetic Algorithm* and *Nearest Neighbor* to match products to customer preferences [Yang et al. 2004], or the analysis of the willingness of Europeans to adopt sustainable travel habits [Bassi and Vera 2023]. Although these methods are suitable for feature matching, they overlook relational structures, thus limiting the identification of interaction-based profile patterns.

Graph-based networks represent systems through interacting elements. Network Science researchers developed topological metrics to describe networks like social or biological ones [Yang et al. 2016]. Metrics fall into three groups: distance (e.g., diameter, betweenness), connection (e.g., degree, clustering coefficient), and spectral (e.g., algebraic connectivity). These measures reveal structural features (e.g., size, density) and interaction patterns (e.g., influence or community structure). For example, modularity detects groups with shared traits or behaviors [Yang et al. 2016].

Graphs can represent social networks, where vertices typically represent individuals, and edges represent relationships. A simple graph model uses undirected, unweighted edges when symmetrical relationships are equally intense. Alternatively, a weighted graph represents the intensity of relationships, with a graph $G = (V, E, f)$ consisting of a set V of vertices, a set E of edges, and a function f mapping edges to weights. In this model, an edge $e_{ij} \in E$ with $e_{ij} = w$ represents a relationship between vertices n_i and n_j with weight w. A weighted digraph can represent the flow of participant comments in a social network. Multi-relational models consider different relationships between individuals with varying relevance [Ramesh et al. 2017]. The number and significance of interactions between individuals determine the strength of a relationship.

Graph clustering identifies groups of vertices with shared interactions or characteristics. Traditional methods include graph partitioning, hierarchical, and partitional clustering [Fortunato 2010]. In a network represented by graphs, a community is a locally dense subgraph where vertices share common traits or interactions [Barabási and Pósfai 2016]. Community detection reveals network organization, relationships between actors, and dynamic processes [Yang et al. 2016]. The key difference between clustering and Community Detection is that the former focuses on similarity or distance between data points, and the latter is based on edge density within versus outside the community [Fortunato 2010].

According to [Fortunato 2010], a community is a subgraph g of a graph G with $|g| = q_g$, where q_g is the number of vertices in g. A community is considered quality if $k_v^{ext} = 0$ and strong if $\delta_{int}(g)$ exceeds the average density $\delta(G)$ of G. For g to be a community, it must have significantly higher internal density than the average density of the entire graph. The internal degree k_v^{int} is the number of edges connecting vertex v to g, and the external degree k_v^{ext} is the number of edges connecting v to the rest of the graph.

A quality assessment is key for evaluating clustering algorithms and understanding network relationships. Modularity is a widely used quality measure in Community Detection, with Q-Modularity

[Newman and Girvan 2004] being particularly common. It uses a null model where edges are reconnected while respecting vertex degrees, selecting the partition with the highest modularity.

Yang [Yang et al. 2016] compares Community Detection methods, including *Fast Greedy*, *Multilevel*, and *Walktrap*, for accuracy and computational efficiency. The *Multilevel* method uses a bottom-up approach, iteratively moving vertices between communities to maximize modularity. The process continues until no further improvements are possible, with a $O(|V| \log |V|)$ complexity. *Fast Greedy* merges communities to maximize modularity, with a complexity of $O(|V| \log^2 |V|)$. *Walktrap* employs an agglomerative approach based on random walks, with a complexity of $O(|E| \cdot |V|^2)$, improving to $O(|V|^2 \log |V|)$ for sparse graphs. Further details are available in [Diboune et al. 2024].

3. DEVELOPMENT

The research carried out in this work is experimental, with a qualitative-quantitative approach and exploratory objective, which aims to identify groups of student aid applicants with similar socioeconomic profiles through graph-based models and Community Detection methods. Figure 1 illustrates the model of the solution. In step 1, the raw data were collected via an online request form. In step 2, for each student (S), a matrix stores the numerical attributes per capita income, per capita expenditure, and total value of family assets, which were transformed into ordinal attributes to reduce variability. An additional motivation for the transformation is to enable prioritization of classes considered to be more vulnerable. After the pre-processing, the dataset used in the experiments only contains the attributes deemed relevant in a first analysis, detailed in the Data Dictionary of Table I. The datasets are available at the link `http://dx.doi.org/10.13140/RG.2.2.19808.49926`.

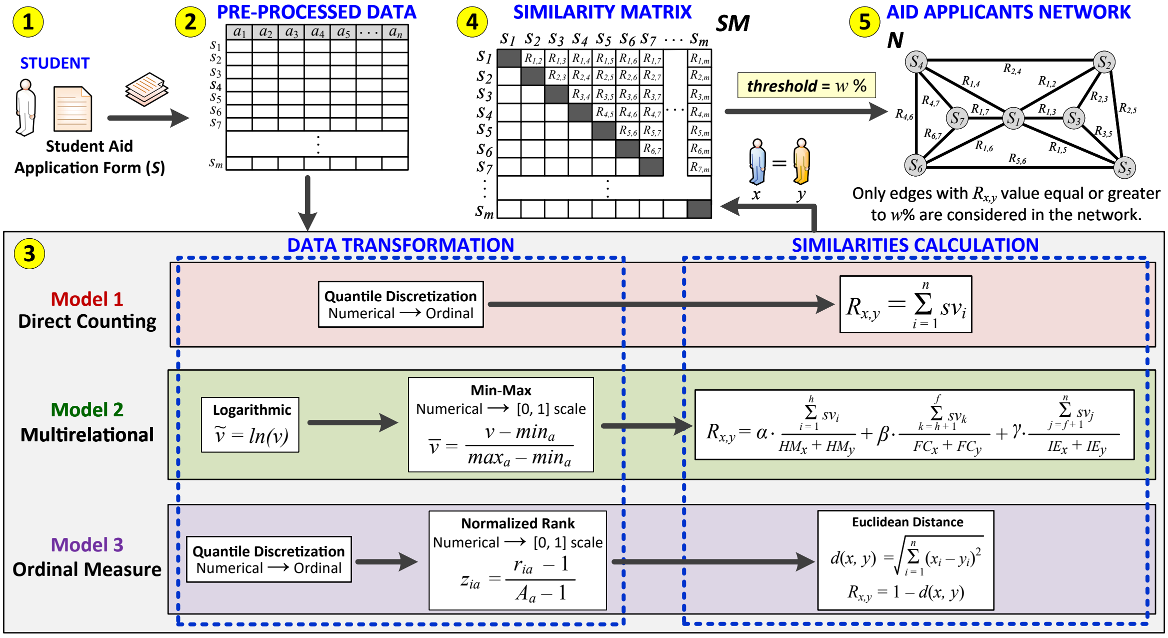
1
STUDENT
Student Aid Application Form (S)
2
PRE-PROCESSED DATA
a_1 a_2 a_3 a_4 a_5 ⋯ a_n
s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_m
4
SIMILARITY MATRIX
SM
s_1 s_2 s_3 s_4 s_5 s_6 s_7 ⋯ s_m
$R_{1,2}$ $R_{1,3}$ $R_{1,4}$ $R_{1,5}$ $R_{1,6}$ $R_{1,7}$ $R_{1,m}$
threshold = w %
x y
5
AID APPLICANTS NETWORK
N
S_1 S_2 S_3 S_4 S_5 S_6 S_7
$R_{2,4}$ $R_{1,4}$ $R_{1,2}$ $R_{4,7}$ $R_{1,7}$ $R_{1,3}$ $R_{2,3}$ $R_{2,5}$ $R_{4,6}$ $R_{6,7}$ $R_{3,5}$ $R_{1,6}$ $R_{1,5}$ $R_{5,6}$
Only edges with $R_{x,y}$ value equal or greater to w% are considered in the network.
3
DATA TRANSFORMATION
SIMILARITIES CALCULATION
Model 1
Direct Counting
Quantile Discretization
Numerical ⟶ Ordinal
$R_{x,y} = \sum_{i=1}^{n} sv_i$
Model 2
Multirelational
Logarithmic
$\tilde{v} = ln(v)$
Min-Max
Numerical ⟶ [0, 1] scale
$\overline{v} = \frac{v - min_a}{max_a - min_a}$
$R_{x,y} = \alpha \cdot \frac{\sum_{i=1}^{h} sv_i}{HM_x + HM_y} + \beta \cdot \frac{\sum_{k=h+1}^{f} sv_k}{FC_x + FC_y} + \gamma \cdot \frac{\sum_{j=f+1}^{n} sv_j}{IE_x + IE_y}$
Model 3
Ordinal Measure
Quantile Discretization
Numerical ⟶ Ordinal
Normalized Rank
Numerical ⟶ [0, 1] scale
$z_{ia} = \frac{r_{ia} - 1}{A_a - 1}$
Euclidean Distance
$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
$R_{x,y} = 1 - d(x, y)$

Fig. 1. Model that details building the aid applicants network.

In step 3, one of the models is run to build the aid applicant network. Each model takes a different approach to determining the similarity between individuals based on their characteristics (attributes). All models generate static networks with no direct interaction between individuals. The three models differ in the data transformation operations applied and how the similarity between pairs of individuals is calculated. The ***Model 1* - Direct Counting** considers a simple, undirected, valued graph whose edge weights represent the degree of resemblance between a pair of students requesting student aid. The resemblance degree is given by the total number of attributes with equal values between two applicants. Quantile-based discretization was used for categorization, with the records divided into ten classes with similar quantities. Then, the values of each attribute of the feature vectors of applicants are compared, and a similarity vector sv stores the value 1 if the attributes are the same value or 0 otherwise. Concluding the analysis of all attributes, the values in sv are summed to obtain the resemblance degree $R_{x,y}$ of the pair of candidates, and stored in SM (step 4).

Table I. Dictionary of the socioeconomic data set of scholarship applicants.

ATTRIBUTE	DESCRIPTION	TYPE	VALUES
school_origin	Type of school of origin of the applicant.	Categorical	Public; Philanthropic; Private with scholarship $<$ 50%; Private with scholarship $\geq$ 50%; Private without scholarship.
housing	Applicant's housing situation.	Categorical	Lives with Family, Relatives, Third Parties; Shares rent with Colleagues; Alone in Rented or Financed property; Alone in owned property paid off.
housing_family	Housing situation of the applicant's family.	Categorical	Rented; Assigned or Inherited; Owned in payment; Owned paid off.
sons	Number of sons the applicant has.	Integer	Integer values greater than or equal to 0
income	Family per capita incomes.	Integer	Integer values between 0 and 2,902.
expense	Family per capita expenses.	Integer	Integer values between 0 and 1,659.
disease	Number of individuals with serious disease in the family group.	Integer	Integer values greater than or equal to 0.
family_education	Number of family members with completed higher education or postgraduate studies.	Integer	Integer values greater than or equal to 0.
family_assets	Total value of family assets.	Integer	Integer values between 0 and 700,000.
transport	Means of transport used to get to the University.	String	Public transport; Ride; On foot or Bicycle; Car or Motorcycle (own); Rent transport.

The ***Model 2* - Multirelational** considers an undirected and valued graph in which the attributes of the applicants are organized into three group types: *Housing and Mobility* (HM), *Family Condition* (FC), and *Income and Expenses* (IE). The organization of the attributes into groups allows the application of different weights to represent a priority or greater relevance. For the case, the HM, FC, and IE groups of attributes have α, β, and γ weights, respectively. Each type of group brings together attributes with similar purposes; the organization defined for the groups was also based on the institution's student aid selection notices. The HM group comprises the aid applicant's housing situation, the family housing situation, and the transportation used to get to the university. The FC group includes the following applicant attributes: type of school origin, number of sons, number of individuals in the family with a severe disease, and number of family members with a college degree or postgraduate degree. Finally, the IE group comprises the per capita income, total value of family assets, and per capita expenses.

In the data transformation stage, the logarithmic transformation is applied to the pre-processed data, which reduces the number of null values due to the amplitude between the minimum and maximum values. Next, Min-Max normalization produces a linear transformation in the original data and preserves the existing relationships. In the Similarities Calculation stage, the resemblance degree R_{xy} between an applicant pair is calculated, with the variables α, β, and γ representing the weights of the attribute groups HM, FC, and IE, respectively. For each attribute i of a given group, the number of equal features between applicants x and y is obtained, divided by the sum of the total equalities of the same attribute both applicants have. This aims to relativize the strength of the equalities of the applicant's pair, considering the total of equalities that each one has. The total value of the number of equalities is multiplied by the respective attribute group weight predefined. Then, adds up the values of each group of attributes to obtain the R_{xy} between requesters x and y.

The ***Model 3* - Ordinal Measure** relies on a similarity measure for ordinal data, organizing them into hierarchical classes. Ordinal data helps reduce the diversity of attribute values by representing those of near magnitude by the same category, allowing ranking. This data can be obtained by quantile discretizing numerical attributes to ordinal values with minimum and maximum values. Next, the

Normalized Rank z_{ia} of the values of each attribute is calculated, and the data are transformed to a value between [0, 1]. The resemblance degree is calculated based on distance measures (e.g., Euclidean, City Block) or similarity measures (e.g., Cosine, Pearson).

In step 4, each applicant pair's $R_{x,y}$ is stored in a similarity matrix SM of dimension $m \times m$, where m is the number of applicants. The aid applicant's network N is generated in step 5 from the SM, considering the edges that meet the threshold w; that is, if $w = 50\%$, only edges with $R_{x,y}$ equal to or greater than this percentage will be considered. Finally, the network is subjected to Community Detection algorithms to identify groups based on socioeconomic similarities.

3.1 Design of Experiments

For each of the previously described models, three Community Detection algorithms - *Fast Greedy*, *Multilevel*, and *Walktrap* - were applied. Additionally, networks were generated with a similarity threshold between individuals ranging from 0% to 90%, in 5% increments. As a result, each model–method pair was tested 19 times, totaling 513 experiments considering the three years.

Based on the standard of the student assistance program of the institution studied, the weights of the attribute sets of the Multirelational Model were defined as follows: the total value 6 was considered as the sum of the weights of all attribute sets, with the attribute set Housing and Mobility weighting 2 (corresponding to 33%), Family Condition weighting 1 (17%), and Income and Expenses having a weight of 3 (50%).

Table II shows the criteria used to rank the best models, defined together with the business specialist. Considering the study scenario as a reference, networks between 2 and 5 communities will be selected.

Table II. Criteria for selecting the most promising networks.

PRIORITY	**CRITERIA**	**DESCRIPTION**
1	Number of communities ($\#C$)	2 to 5
2	Minimum number of individuals in the community	$\geq$ 5% of the total
3	Modularity (M)	highest
4	Number of edges between communities ($\#EC$)	lowest
5	Threshold similarity between individuals ($w\%$)	highest

The model's development and experimentation environment is a notebook with an Intel Core i7-10510U processor at 2.30GHz, 8GB of RAM, and Windows 11 Pro 64-bit operating system version 21H2. The language used is *Python* 3.10.4, with the libraries *igraph* 0.10.2 and *networkx* 2.8.8 for graph manipulation and use of the Community Detection algorithms *Fast Greedy*, *Multilevel*, and *Walktrap*. The following *Python* packages were used in the data preprocessing, graph generation and reporting tasks: *Pandas* 1.4.2, *numpy* 1.22.3, *scikit-learn* 1.1.3, *Matplotlib* 3.5.1, *openpyxl* 3.0.9, *xlrd* 1.2.0 and *xlwt* 1.3.0. The *software Gephi* 0.9.6 was used to illustrate the networks.

4. RESULTS AND DISCUSSION

Table III describes the results of the experiments, organized by year. The models are organized into three primary columns, followed by subcolumns with the two methods that presented the best results for each data year and model combination. The column $w\%$ is the minimum percentage of resemblance threshold between individuals, $\#EC$ is the total number of edges between communities, Q is the modularity, and $\#C$ is the number of communities found.

Regarding the number of communities obtained by each method, each model's behavior remains the same with data from different years (e.g., *Model 1* $w\%$ close to 50%, *Model 2* 5%, and *Model 3* 60%). *Model 2* stands out for its greater sensitivity to changes in $w\%$, considering that small increases significantly impact the $\#EC$ reduction. By gradually increasing the value of $w\%$, the number of communities could be adjusted accurately, avoiding subgrouping and overfragmentation.

Table III. Experiment results with data from 2018, 2019, and 2020, of the networks generated by each model, community detection methods, and resemblance threshold.

	2018																	
	Model 1 (Direct Counting)						Model 2 (Multirelational)						Model 3 (Ordinal Measure)					
	Walktrap			*Fast Greedy*			*Walktrap*			*Fast Greedy*			*Multilevel*			*Fast Greedy*		
$w\%$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$
0	273,828	0.072	3	205,732	0.053	2	275,346	0.434	3	295,178	0.451	3	388,073	0.000	7	381,492	0.000	6
5	215,358	0.075	2	200,171	0.053	2	**1,820**	**0.475**	**5**	**2,454**	**0.482**	**5**	358,362	0.000	5	302,496	0.000	3
10	215,358	0.075	2	200,171	0.053	2	239	0.488	6	293	0.488	6	336,423	0.001	5	241,483	0.001	3
15	175,099	0.080	2	176,730	0.061	2	239	0.488	6	293	0.488	6	266,372	0.003	3	208,439	0.002	2
20	175,099	0.080	2	176,730	0.061	2	239	0.488	6	293	0.488	6	254,941	0.007	3	198,385	0.006	3
25	175,099	0.080	2	176,730	0.061	2	217	0.488	7	272	0.488	7	160,707	0.021	2	162,646	0.020	3
30	104,982	0.141	2	108,175	0.099	2	40	0.523	717	43	0.530	716	132,437	0.038	2	134,821	0.035	3
35	104,982	0.141	2	108,175	0.099	2	19	0.487	763	20	0.490	763	124,588	0.064	3	98,444	0.062	2
40	39,961	0.222	2	45,902	0.189	2	19	0.487	763	20	0.490	763	71,164	0.095	2	71,234	0.092	2
45	39,961	0.222	2	45,902	0.189	2	19	0.487	763	20	0.490	763	57,950	0.151	3	43,073	0.149	3
50	**39,961**	**0.222**	**2**	**45,902**	**0.189**	**2**	31	0.561	782	31	0.569	782	37,025	0.211	3	23,762	0.205	2
55	15,257	0.417	10	14,024	0.371	6	2	0.499	862	2	0.499	862	**19,247**	**0.282**	**3**	11,327	0.267	2
60	15,257	0.417	10	14,024	0.371	6	2	0.499	862	2	0.499	862	10,463	0.351	6	**4,239**	**0.321**	**4**
	2019																	
	Walktrap			*Fast Greedy*			*Multilevel*			*Fast Greedy*			*Multilevel*			*Fast Greedy*		
$w\%$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$
0	76,351	0.073	3	57,770	0.089	2	72,607	0.374	3	74,313	0.358	3	85,241	0.000	4	82,638	0.000	4
5	73,266	0.073	3	54,786	0.089	2	**6,591**	**0.417**	**4**	2,057	0.394	2	85,004	0.000	4	75,872	0.000	3
10	73,266	0.073	3	54,786	0.089	2	4,282	0.427	6	**833**	**0.418**	**5**	78,533	0.000	4	71,163	0.001	3
15	42,592	0.086	2	44,903	0.104	3	3,693	0.432	16	210	0.421	15	75,992	0.004	4	52,181	0.003	2
20	42,592	0.086	2	44,903	0.104	3	3,687	0.428	15	159	0.421	15	63,900	0.009	3	48,294	0.006	2
25	42,592	0.086	2	44,903	0.104	3	3,540	0.433	21	58	0.422	20	55,925	0.018	3	41,347	0.014	2
30	24,761	0.126	2	25,980	0.160	3	3,540	0.433	21	58	0.422	20	47,370	0.029	3	35,520	0.026	2
35	24,761	0.126	2	25,980	0.160	3	185	0.758	87	173	0.685	82	26,951	0.061	2	26,953	0.060	2
40	18,785	0.211	5	9,011	0.258	2	0	0.749	455	0	0.749	455	19,182	0.095	2	19,314	0.091	2
45	18,785	0.211	5	9,011	0.258	2	0	0.586	465	0	0.586	465	12,493	0.140	2	12,463	0.141	2
50	**18,785**	**0.211**	**5**	9,011	0.258	2	0	0.431	469	0	0.431	469	7,648	0.182	2	7,604	0.184	2
55	4,359	0.371	17	2,332	0.360	5	0	0.000	475	0	0.000	475	6,479	0.229	3	4,275	0.226	2
60	4,359	0.371	17	**2,332**	**0.360**	**5**	0	0.000	475	0	0.000	475	**4,199**	**0.293**	**4**	**2,663**	**0.263**	**3**
	2020																	
	Walktrap			*Fast Greedy*			*Walktrap*			*Fast Greedy*			*Multilevel*			*Fast Greedy*		
$w\%$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$	$\#EC$	Q	$\#C$
0	396,221	0.080	2	468,520	0.073	3	448,402	0.473	3	403,921	0.468	2	689,092	0.000	7	715,036	0.000	8
5	384,226	0.080	2	458,422	0.073	3	**1,361**	**0.496**	**4**	1,140	0.489	2	630,228	0.000	5	509,916	0.000	3
10	384,226	0.080	2	458,422	0.073	3	120	0.499	6	206	0.497	3	468,272	0.002	3	400,233	0.001	3
15	325,709	0.091	2	409,215	0.082	3	119	0.499	6	205	0.497	3	413,497	0.005	3	367,307	0.004	3
20	325,709	0.091	2	409,215	0.082	3	119	0.499	6	205	0.497	3	411,470	0.011	3	347,921	0.007	3
25	325,709	0.091	2	409,215	0.082	3	118	0.499	6	204	0.497	3	363,438	0.020	3	266,075	0.017	2
30	194,261	0.145	2	211,742	0.129	2	118	0.499	6	204	0.497	3	206,571	0.049	2	206,823	0.048	2
35	194,261	0.145	2	211,742	0.129	2	118	0.499	6	**204**	**0.497**	**3**	151,370	0.083	2	154,303	0.078	3
40	112,500	0.255	3	94,360	0.261	3	114	0.499	13	101	0.491	6	95,609	0.146	3	91,288	0.142	2
45	**112,500**	**0.255**	**3**	94,360	0.261	3	5	0.714	1042	5	0.714	1042	46,055	0.220	2	46,383	0.215	2
50	112,500	0.255	3	94,360	0.261	3	5	0.706	1048	5	0.706	1048	20,117	0.278	2	20,341	0.275	2
55	30,820	0.418	9	25,833	0.395	5	5	0.706	1048	5	0.706	1048	21,521	0.323	4	8,764	0.302	2
60	30,820	0.418	9	**25,833**	**0.395**	**5**	4	0.695	1076	4	0.695	1076	**9,057**	**0.397**	**5**	**3,375**	**0.322**	**3**

Through modularity (Q), it is possible to identify the level of cohesion within the communities. According to the literature [Newman and Girvan 2004], modularity values close to 0 indicate a weak or non-existent community structure, between 0.3 and 0.7 represent a good division, and values close to 1.0 are rare in real networks. The data show that *Model 2*, especially when combined with the *Fast Greedy* algorithm, presented the highest modularity values in the three years analyzed, reaching 0.497 in 2020. These values indicate that the model was the most effective in identifying cohesive and well-defined communities, which is desirable in studies that seek to group similar socioeconomic profiles. This result is attributed to the model's weights, which help discretize individuals by valuing common characteristics, as expected.

In *Models 1* and *3*, some Community Detection methods could not fully meet all the criteria. For example, the highest modularity of *Model 1* + *Walktrap* is 0.255 in 2020, and this variant cannot obtain the number of groups within the limits of the $\#C$ criterion and within the range of Q values that indicate a good division of communities. Similar cases are also observed in *Model 3* (e.g., 2018 *Multilevel* and 2019, both methods). The $\#EC$ analysis allows us to verify the external isolation of communities. Also in this aspect, *Model 2* + *Fast Greedy* obtained the best results, reaching only 204 external edges in 2020, much smaller than other combinations, which exceeded hundreds of thousands of connections in many cases.

About the execution time, *Models 1* and *2* required an average of 5.64 seconds, with *Model 3* requiring a time approximately thirteen times greater (73.46 seconds on average). A likely explanation is due to the calculation of the Euclidean distance during the construction of the network and execution of the methods. The time decreases with the increase of $w\%$ due to the reduced density of the network with the smaller number of edges. Among the methods, *Walktrap* presented the longest times, justified by their quadratic asymptotic complexity compared to a linear one of other methods.

In summary, the results in Table III point to the superiority of *Model 2* over the *Fast Greedy* algorithm, both in terms of Community Detection quality, efficiency, and stability. Its high modularity, low number of external edges, good adaptability to the similarity threshold, and temporal consistency make it the most robust choice for identifying similar socioeconomic profiles in complex networks.

For practical purposes, Figure 2 better clarifies the cohesion of the communities. The network contains 122,717 edges, with 204 of them external. Community $C1$ contains 67.15% of all applicants who live in rented accommodation, and 97.93% use bicycles or carpooling as transportation or none at all. 56.25% of the individuals have higher incomes (8 to 10), and 63.04% have higher expenses (6 to 10). The characteristics of this group indicate the profile of individuals who share the rent of a residence close to the institution, sometimes in the form of a student republic.

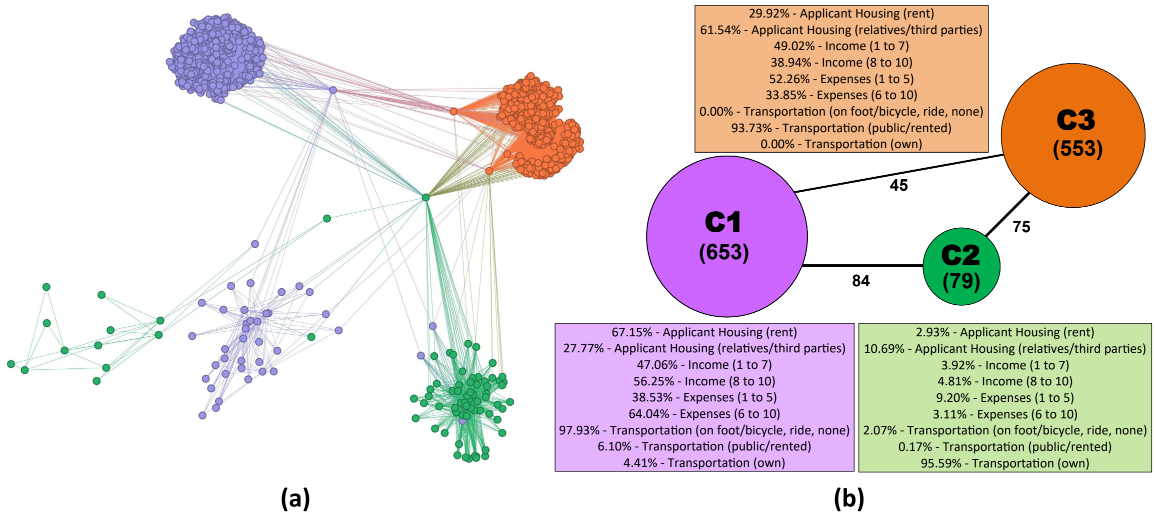
29.92% - Applicant Housing (rent)
61.54% - Applicant Housing (relatives/third parties)
49.02% - Income (1 to 7)
38.94% - Income (8 to 10)
52.26% - Expenses (1 to 5)
33.85% - Expenses (6 to 10)
0.00% - Transportation (on foot/bicycle, ride, none)
93.73% - Transportation (public/rented)
0.00% - Transportation (own)
C3
(553)
45
C1
(653)
75
C2
(79)
84
67.15% - Applicant Housing (rent)
27.77% - Applicant Housing (relatives/third parties)
47.06% - Income (1 to 7)
56.25% - Income (8 to 10)
38.53% - Expenses (1 to 5)
64.04% - Expenses (6 to 10)
97.93% - Transportation (on foot/bicycle, ride, none)
6.10% - Transportation (public/rented)
4.41% - Transportation (own)
2.93% - Applicant Housing (rent)
10.69% - Applicant Housing (relatives/third parties)
3.92% - Income (1 to 7)
4.81% - Income (8 to 10)
9.20% - Expenses (1 to 5)
3.11% - Expenses (6 to 10)
2.07% - Transportation (on foot/bicycle, ride, none)
0.17% - Transportation (public/rented)
95.59% - Transportation (own)
(a)
(b)

Fig. 2. View of (a) the 2020 network of *Model 2* and (b) the mesoscopic view, *Fast Greedy* and $w = 35\%$.

Community $C2$ contains the lowest number of members, bringing together 95.59% of all applicants with their transportation. In $C2$, 10.69% live with family or third parties, 3.92% have lower incomes (1 to 7). In $C3$, 61.54% of the cluster live with family or third parties, and 93.73% of the applicants use public transportation, which may indicate housing in peripheral neighborhoods or even neighboring cities. About half of the individuals in the group (49.92%) have lower incomes (1 to 7), and 52.26% have lower expenses (1 to 5). The profile of individuals in group 3 is made up of students who live with relatives or friends in areas further away from the university and who have low incomes. Low income is suspected to be one factor that justifies living with thirdies, often due to the inability to cover the costs of housing and food. This community brings together the most socioeconomically vulnerable.

5. CONCLUSIONS

Federal higher education institutions' assistance programs provide student aid scholarships to alleviate the difficulties students face in situations of socioeconomic vulnerability. The increase in the number of applicants for aid scholarships, together with the limitations of financial resources, makes the task of moderately ranking those who will be awarded time-consuming and complex. To assist in decision-making on granting aid scholarships, we use graphs to model the network of student aid applicants and Community Detection methods to identify those with similar characteristics.

Computational experiments were carried out to validate the cohesion and assertiveness of the proposed models. The focus is on analyzing the external isolation between the communities in the network and where the students who received the scholarships in 2018 are located within the communities. The cohesion of the groups in each model can be analyzed throughout the experiment by examining the number of edges between the groups. The mesoscopic view from the network contributes to a better understanding of the student communities found. The results confirmed the viability of the models, which were able to represent the candidate network. Community Detection is a viable method, capable of identifying groups with similar socioeconomic profiles, allowing social workers to prioritize the analysis of the most vulnerable groups. In addition, it was possible to obtain more cohesive groups by removing attributes that do not discretize the applicants well and may confuse the methods. All models obtained interesting results with potential for application. *Model 3* stood out in the 2018 and 2019 databases, stood out in 2020, while *Model 2* was relevant in all years studied. The *Fast Greedy* method had better results than the others analyzed.

The application of complex network methods and tools to identify profiles is little explored in the literature, and this article demonstrates the viability of the models and methods used. Further studies may indicate an ideal value for the minimum similarity threshold between applicants, which contributes to obtaining more cohesive communities. Other studies may better elucidate the overlap of communities, indicating a more specialized group of individuals in a larger community. The analysis of other measures for categorical data, such as LIN, Smirnov, and Anderberg, may contribute to the refinement of the models. Using methods to calculate the similarity of mixed-type attributes may generate more cohesive and isolated communities. Finally, graph-based models allow the application of other methods and tools, such as centrality, which can help identify specific cases of applicants who require constant and individualized monitoring due to greater socioeconomic vulnerability.

REFERENCES

Barabási, A.-L. and Pósfai, M. *Network science.* Cambridge University Press, Cambridge, 2016.

Bassi, F. and Vera, José Fernando nd Martín, J. A. M. Profile-based latent class distance association analyses for sparse tables: application to the attitude of european citizens towards sustainable tourism. *Advances in Data Analysis and Classification* vol. 18, pp. 1–28, 2023.

Diboune, A., Slimani, H., Nacer, H., and Beghdad Bey, K. A comprehensive survey on community detection methods and applications in complex information networks. *Social Network Analysis and Mining* 14 (1): 1–47, 2024.

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

Hamim, T., Benabbou, F., and Sael, N. Survey of machine learning techniques for student profile modeling. *International Journal of Emerging Technologies in Learning (iJET)* 16 (04): 136–151, 2021.

Newman, M. E. and Girvan, M. Finding and evaluating community structure in networks. *Physical review E* 69 (2): 026113, 2004.

Ramesh, A., Rodriguez, M., and Getoor, L. Multi-relational influence models for online professional networks. In *Proceedings of the International Conference on Web Intelligence.* WI '17. Association for Computing Machinery, New York, NY, USA, pp. 291–298, 2017.

Yang, H.-W., Pan, Z.-G., Wang, X.-Z., and Xu, B. A personalized products selection assistance based on e-commerce machine learning. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (ICMLC).* Vol. 4. IEEE, Shanghai, China, pp. 2629–2633, 2004.

Yang, Z., Algesheimer, R., and Tessone, C. J. A comparative analysis of community detection algorithms on artificial networks. *Nature Scientific Reports* vol. 1, pp. 1–16, 2016.