São José dos Campos - SP
LV SBPO
LV Simpósio Brasileiro de Pesquisa Operacional
São José dos Campos, SP - 6 a 9 de novembro de 2023

Identifying elements of solutions of combinatorial optimization problems using measures of graph centrality

Maria Claudia S. Boeres
Universidade Federal do Espírito Santo
Av. Fernando Ferrari, 514, Goiabeiras, Vitória, ES, CEP 29075-910
boeres@inf.ufes.br

Alexandre Plastino, Simone de Lima Martins
Universidade Federal Fluminense
Av. Gal. Milton Tavares de Souza, s/n, São Domingos, Niterói, RJ, CEP 24210-346
{plastino,simone}@ic.uff.br

RESUMO

Este trabalho visa dar evidências de que medidas de centralidade em grafos podem ser utilizadas para identificar elementos pertencentes a soluções de problemas de otimização combinatória de determinada categoria, mais especificamente aqueles que podem ser modelados por grafos. Para tanto, foram utilizadas cinco medidas de centralidade e o problema de minimização de vértices d-branch. Utilizando-se métricas comumente adotadas na área de Ciência de Dados para avaliar a acurácia de modelos preditivos multirrótulo, foi possível verificar uma alta capacidade das medidas de centralidades em corretamente identificar vértices d-branch para um grande conjunto de instâncias do problema de minimização de vértices d-branch.

PALAVRAS CHAVE. centralidades, grafos, otimização combinatória

Tópicos: Otimização combinatória, Inteligência Computacional

ABSTRACT

This work aims to show evidence that measures of centrality in graphs can be used to identify elements belonging to solutions of combinatorial optimization problems of a particular category, specifically those that can be modeled by graphs. Five centrality measures and the d-branch vertex minimization problem were used for that. Using metrics commonly adopted in the area of Data Science to assess the accuracy of multi-label predictive models, it was possible to verify a high capacity of the measures of centralities in correctly identifying d-branch vertices for a large set of instances of the d-branch vertices minimization problem.

KEYWORDS. centralities, graphs, combinatorial optimization

Paper topics: Combinatorial optimization, Computational Intelligence

1. Introduction

Several combinatorial optimization problems are related to finding specific vertices in a graph. This work aims to verify if vertex's centrality measures can identify these vertices, which could be helpful in the design of efficient heuristics for those problems. To evaluate the capacity of centrality measures in identifying these key vertices, we employed metrics commonly adopted in the Data Science area to assess the accuracy of multi-label predictive models.

Graph centralities are measures of vertex importance typically used in social network analysis. The link structure of the nodes in the network strongly influences these measures defined according to a particular criterion for classifying them. In this way, they can be used to classify the vertices according to their importance in the connection structure of the graph to which they belong. An invariant of a graph consists of a property preserved in isomorphic graphs, which presents the same structure of links between vertices. The centrality measures (or simply centralities) are invariants of the graph. The best-known graph centralities in the literature are degree, betweenness, closeness, eigenvector, and PageRank.

Heuristics have proven to be a handy tool in solving combinatorial optimization problems. They do not guarantee to obtain the optimum, but they are efficient algorithms that generate good-quality solutions for the most diverse applications. Knowledge of the characteristics of the problem to be solved and its solution space is very relevant for choosing and adapting the heuristic for its solution.

Centralities are a reasonable criterion for heuristics to solve optimization problems on graphs, as they are directly connected to the graph's topological structure. We present some works that use or evaluate the centralities of graphs to develop solutions for optimization problems in graphs.

[Sharma et al., 2016] performed an empirical study comparing five centrality measures in the context of Protein-Protein Interaction (PPI) networks. A PPI network (PPIN) represents the interoperability of proteins and how they coordinate to perform certain functions. It may be represented by a graph where the nodes are the proteins, and the edges are the interaction between them. Protein complexes are groups of similar proteins that work toward accomplishing specific metabolic functions. Determining these complexes helps to find the functions of unknown proteins. Several methods exist to detect these complexes, but their accuracy could be improved. This study performed an empirical analysis to verify whether these complexes are associated with measures of betweenness centrality, eigenvector centrality, PageRank centrality, closeness centrality, and radiality centrality. The authors measured some complexes' average centralities and showed they present high PageRank and radiality values. Therefore, they suggest using these measures to help find protein complexes.

[Herrmann et al., 2018] extend previous work showing that PageRank centrality measures in local optima network (LON) accurately predict the performance of local search-based metaheuristics. A LON is a directed graph where the vertices are the search space's local optima, and the edges model the transitions between the local optima whose weight is the transition probability of moving from one basin of attraction to another. Some works showed that the performance of first-improvement local search, Simulated Annealing, and Iterated Local Search heuristics could be well predicted using the PageRank centrality of the global optimum for LONs with edges representing basin transition probabilities or the number of escape edges. This work studies whether and how the definition of LON edges affects the ability of PageRank centrality to predict the performance of local search-based metaheuristics accurately.

[Frinhani et al., 2018] propose a heuristic using PageRank centrality to solve the mini-

mization of open stacks problem (MOSP) that aims to determine the ideal production sequence to optimize the occupation of physical space in manufacturing settings. At each stage of a production process, a large piece is processed from smaller pieces obtained from specific stacks close to the machine that produces the larger one. Physical constraints prevent space allocation for the simultaneous accommodation of stacks of all requested pieces. Once a stack is open, it can only be closed and open space in the room when the demand for pieces of the same type is no longer requested. Therefore, the objective of the problem is to determine the sequence of processing the larger pieces that minimizes the number of open stacks. This work proposes a PageRank-based heuristic to solve large instances modeled in graphs. The results showed that the heuristic is competitive regarding quality and computational time.

[Lozano and Trujillo, 2019] present an optimization problem that selects nodes in a network to link each of a particular number of newly infiltrated nodes to maximize the lower betweenness centrality value obtained by the infiltrated nodes. Betweenness measures the participation of the nodes in the shortest paths of the network, and it has been widely used as a centrality measure in analyzing social networks to identify key players in social networks, for example. They proposed a local-search-based heuristic to solve the problem. A construction phase generates a solution by connecting each infiltrated node to the node that maximizes its betweenness. Then, a local search procedure performs swap and interchange moves among the infiltrated nodes.

[Martin and Niemeyer, 2019] propose a method that allows researchers to estimate the robustness of a centrality measure in a specific network and can, therefore, use it as a basis for decision-making, with experiments applied to random graphs and real-world networks. They observed that their method brings a good approximation for the robustness of centrality measures. They also proposed a heuristic to decide whether the estimation procedure should be used. They analyzed, for certain networks, why the eigenvector centrality is less robust than, among others, the PageRank.

[Alozie et al., 2022] proposed a centrality-based heuristic to solve the distance-based critical node problem that aims to find a subset of nodes of cardinality at most B, whose removal minimizes the total number of pairs of nodes connected by a hop distance of at most k. The heuristic executes a construction phase that uses degree, e-Katz, and betweenness centralities to generate solutions later combined by a backbone-based crossover procedure to generate good-quality offspring solutions. The local search procedure tries to improve these solutions generated using neighborhoods based on these three centralities. Computational experiments showed that the proposed heuristic generated good-quality solutions compared to exact solutions, particularly for some challenging problem instances.

In this work, we intend to evidence that centralities measures can be used to identify elements that belong to solutions of graph combinatorial optimization problems. We conducted experiments using five centrality measures – degree, betweenness, closeness, eigenvector, and PageRank – and the d-branch vertex minimization problem (d-MBV). The d-MBV problem consists of finding in a connected and undirected graph G, a spanning tree with a minimum number of vertices with a degree strictly greater than d, for $d \geq 2$. The motivation is to minimize the number of switches in optical networks.

To show the centrality measures' ability to identify d-branch vertices correctly, we adopted measures commonly used in data science to assess multi-label predictive models. These measures are called meta-measures since they evaluate other measures, the centrality ones.

Section 2 describes the graph centralities and the meta-measures. Section 3 shows the results obtained evaluating centralities for the d-MBV problem, which is also described in this

section. Section 4 presents the conclusions and future works.

2. Centrality Measures and Meta-Measures

As graph centralities give information about the role of vertices in the graph structure, in this work, we investigate five well-known that may help in solving graph optimization problems. These centralities are degree, betweenness, closeness, eigenvector, and PageRank (Section 2.1).

In addition, we borrow data science measures to evaluate the centralities. They are F-score and four example-based ranking measures. Their definitions, reproduced from [Zhao et al., 2010] and [Pereira et al., 2018], are presented in Section 2.2.

2.1. Centrality Measures

Centrality measures rank the nodes of a graph based on their topological importance. It is a graph-invariant property preserved by isomorphism. Their objective is to classify the vertices from the most to the least central (important) according to a particular criterion. The most popular centralities are based on the number of connections like vertices degree, distance, such as closeness and betweenness, and spectral properties associated with the matrices of the graph, such as eigenvector and PageRank.

The mathematical definitions of these five centralities are described in an undirected graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges.

Degree: The degree $d(v)$ of a vertex $v \in V$ shows its communicativeness or popularity. It is defined as the number of vertices adjacent (or connected) to v and is given as $d(v) = k_v$, where k_v is the number of edges incident to v.

Betweenness: The betweenness $b(v)$ of a vertex $v \in V$ indicates its potential for communication control and behavior as a mediator in the network, quantifying the number of times v acts as a bridge on shortest paths connecting two other G nodes. Thus, it corresponds to the proportion of geodesics (shortest paths) that pass through the node v and is given by $b(v) = \sum_{a,b \neq v}(\frac{g_{avb}}{g_{ab}})$, where g_{ab} is the number of geodesics between vertices a and b and g_{avb} is the number of those geodesics that pass through v.

Closeness: The closeness centrality $c(v)$ of a vertex $v \in V$ determines how easily it is possible to reach other vertices of the graph from v, therefore, how close it is to the other vertices. A vertex closer to others can obtain information more efficiently. It corresponds to the inverse of the sum of the distances from one node to the others, given by $c(v) = (\sum_{a \in V, a \neq v} d(v, a))^{-1}$, where $d(\cdot, \cdot)$ is the distance between two nodes of G.

Eigenvector: The eigenvector centrality assigns relative scores to all nodes in a network based on the principle that connections to nodes with higher scores contribute more to the node's score in issue than connections to low-scoring nodes.

Let p be a n-dimensional column vector that contains the eigenvector centrality values associated with each vertice $v \in V$ and $A_{n \times n}$, the G adjacency matrix. The vector p is recursively updated for k iterations from an initial vector $(1, 1, ..., 1)^T \in \mathbb{R}^n$, being iteratively multiplied by A^T and normalized by the highest calculated value until convergence is reached. When k is large, $p_k = A^T p_{k-1} \approx \lambda p_{k-1}$. The process to obtain the eigenvector centrality vector p is performed with the power method [Zaki and Wagner Meira, 2020], which provides the dominant eigenvector and its associated eigenvalue λ as the answer.

PageRank: The PageRank centrality can be understood as a variant of the eigenvector algorithm used by the Google search engine. It was projected to be used on the Web graph, which consists of pages (vertices) and hyperlinks (graph arcs); thus, it is mainly applied on directed networks. The PageRank calculation for the Web graph considers the number of inbound links (i.e., sites pointing to a given site), the quality of the links (that is, the PageRank of the sites that point to

the given site (linkers)), and the link propensity of the linkers (i.e., the number of sites the linkers link to).

The computation of the PageRank values associated with the vertices of the Web graph uses the random surfing assumption that a person surfing the Web randomly chooses one of the outgoing links from the current page or, with some small probability $1 - damping$, randomly jumps to any of the other pages. Like eigenvector centrality, the PageRank of a vertice recursively depends on the PageRank of those pointing to it.

Considering undirected graphs, the PageRank computation uses the degree instead of in and out-degree vertices. Thus, the PageRank centrality $p(v)$ of a vertex $v \in V$ is given by $(1 - d)/n + d * \sum_{\{v,w\} \in E} p(v)/d(w)$, where d is the damping factor, usually set as 0.85 and $d(v)$ represents the degree of vertice v.

2.2. Meta-Measures

We chose the F-score measure and four multi-label ranking measures from the data science field to evaluate the influence of the centrality measures in identifying d-branch vertices. The F-score metric is the harmonic mean of the precision and recall metrics. Therefore, it represents both precision and recall in one metric. The four ranking measures are adaptations of well-known ranking measures used to evaluate multi-label classifiers [Pereira et al., 2018].

2.2.1. F-score

The F-score measure (or F-statistic) is used to test how well a feature discriminates samples from different classes, evaluating the relation between class variance and within class variance [Zhao et al., 2010]. The F-score is computed as

$$\text{F-score} = \frac{\text{variance between classes}}{\text{variance within classes}} = \frac{\sum_{i=1}^{c} \frac{N_i}{c-1}(\mu_i - \mu)^2}{\frac{1}{N-c}\sum_{i=1}^{c}(N_i - 1)\sigma^2},$$

where c is the number of classes, N is the total number of samples, N_i is the number of samples with class i, μ is the mean of the values from all samples, μ_i is the mean of the values from samples with class i and σ_i^2 is the variance of the values from samples with class i. The higher the F-score, the better the feature evaluated discriminates the classes compared.

2.2.2. Ranking Measures

This section presents three ranking measures adapted to evaluate the centralities applied to graph optimization problems. Their definitions are reproduced from [Pereira et al., 2018].

The notation adopted is the following. Given a multi-label data set D with N instances (examples), for each example (x_i, Y_i), $i = 1, \ldots, N$, x_i is the set of attributes values and $Y_i \subseteq L$, the set of true labels, each one belonging to the set of q labels $L = \{\lambda_j | j = 1, \ldots, q\}$. Given an instance x_i, the rank of labels predicted by a ranking method is denoted as r_i, and $r_i(\lambda)$ is the rank position of a label λ. The most relevant label receives the highest rank (1), while the least relevant one receives the lowest (q). Additionally, H is the model generated by the multi-label learning task, capable of predicting a subset of labels given an unseen instance.

All the ranking measures considered the label ranking generated by the classifier, averaging the results over all the instances.

One Error: evaluates how frequently the top-ranked label is not in the set of the relevant labels of the instance. It is defined by $OneError(H, D) = \frac{1}{N}\sum_{i=1}^{N} \delta(\text{argmin } r_i(\lambda))$ where $\lambda \in L$, $\delta(\lambda) = 1$, if $\lambda \notin Y_i$ and 0, otherwise.

Ranking Loss: indicates the number of times that irrelevant labels are ranked higher than relevant labels. It is given by $Ranking\_Loss(H, D) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{|Y_i||\bar{Y_i}|}|\{(\lambda_a, \lambda_b) | r_i(\lambda_a) > r_i(\lambda_b), (\lambda_a, \lambda_b) \in Y_i \times \bar{Y_i}\}|$.

Average Precision: computes for each relevant label the proportion of relevant labels that are ranked before and including it. Finally, averages over all relevant labels. It is given by $Average\_Precision = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i|}\sum_{\lambda\in Y_i}\frac{|\{\lambda'|\lambda'\in Y_i, r_i(\lambda')\leq r_i(\lambda)\}|}{r_i(\lambda)}$.

3. Evaluating centralities for the minimum d-branch problem

We applied our proposal to evaluate centralities' role in graph optimization problems in the minimum d-branch problem. We describe this problem, show the instantiation of the meta-measures to this problem, and discuss the results obtained by applying the meta-measures on centralities.

3.1. The minimum d-branch problem

Given a connected and undirected graph G, the Minimum Number of Branch Vertices (MBV) problem aims to obtain a spanning tree of G with the smallest number of vertices with a degree greater than 2. Such vertices are called branch vertices. This problem was formulated in Gargano et al. [2002] to determine the best allocation positions for *switches* in optical networks. The authors also proved that the MBV problem is NP-hard.

A generalization of MBV, the k-MBV, was proposed in Merabet et al. [2018], using the concept of k-branch, representing vertices with degrees strictly greater than $k+2$. The k value characterizes a tolerance parameter in optical network projects. If a light signal is split into k copies, the signal power of one copy is reduced by at least a factor of $1/k$ of the original signal power. The k-MBV was also proved to be *NP-hard* for any value of k.

The d-MBV problem [Moreno et al., 2018] was proposed to simplify the notation of the k-MBV problem, introducing the parameter $d = k+2$. The d-MBV problem consists of finding in an undirected graph G a spanning tree with a minimum number of vertices d-branch. A d-branch vertice is a vertice with a degree strictly greater than d, for $d \geq 2$. Moreno et al. [2018] formulate and solve the d-MBV as an Integer Programming (IP) problem. An Iterated Local Search (ILS) heuristic is also proposed for the problem solution.

3.2. Centralities and Meta-Measures adapted for the minimum d-branch problem

The d-MBV problem is defined over an undirected and connected graph G. As such, the five centralities presented in Section 2.1 were computed for the n vertices of G. The centralities vectors were generated using functions provided in the graph library Igraph [Csardi and Nepusz, 2006]. In this section, we also discuss the adaptations imposed on the definitions of the meta-measures presented in Section 2.2 for the minimum d-branch problem and propose an extra ranking measure called Spread Ranking.

3.2.1. F-score

The F-score was calculated for each instance of the minimum d-branch problem and for each of the five centralities computed. For each calculation, the set of vertices of G was considered as the set of elements, split into two classes: the set of vertices d-branch and the set of vertices non-d-branch. These classes were defined from the optimal solutions an IP solver generated for the problem instances.

3.2.2. Ranking Measures

Four different ranking measures were also applied to evaluate how the centralities are well-ranking the d-branch vertices.

Let $G = (V, E)$ be an undirected graph with n vertices and m edges that is the input of the minimum d-branch problem; the sets $Y = \{v \in V | v \text{ is d-branch}\}$ of k vertices d-branch, $0 \leq k \leq n$ and $\bar{Y} = V - Y$ of $n-k$ vertices non d-branch, where the k d-branch vertices are obtained from the optimal solution generated by an IP solver; r, an n-dimensional vector that stores

the ranking positions of the n vertices of G, according to their descending order relative to their centrality values. Thus, position 1 of r is associated with the highest centrality value, and position n with the lowest value. If there are ties in centrality values, their ranking positions all become the same, equal to the average of the original positions.

The ranking measures are defined for each centrality in the following.

k_Error (RM1): evaluates how often k top-ranked vertices are not d-branch. It is given by **k**_Error $= \frac{1}{k}\sum_{i=1}^{k}\delta(\lambda)$, where $i = 1, \ldots, k$ corresponds to the first k ranking positions of the vertices of G. The function $\delta(\lambda) = 1$, if $\lambda \notin Y$ and 0 otherwise. So, if all the first k vertices of the ranking are d-branch, **k**-Error is 0. However, if all are not d-branch, the result is 1. The **k**-Error measure is an adaptation of One Error, presented in [Pereira et al., 2018]. For example, suppose that an instance with 20 vertices has 5 d-branch vertices. A centrality measure ranks its vertices, and k equals 5. The first five best-ranked vertices are checked in the exact solution if they are d-branch, the first three are d-branch vertices, and the other two are not. So, the k_Error is equal to $(0+0+0+1+1)/5$ or 0.4.

Spread Ranking (RM2): this measure considers the average of ranking positions relative to all d-branch vertices. Its name was inspired by how the d-branch vertices are spread across the ranking positions. If they are together in the first positions, the measured value tends to be smaller. However, on the contrary, if the d-branch vertices occupy scattered positions or are at the end of the ranking vector, the measured value tends to be greater. It is given by Spread Ranking $= \frac{1}{k}\sum_{i=1}^{n}\delta'(\lambda) * r(i)$. The function $\delta'(\lambda) = 1$, if $\lambda \in Y$ and 0 otherwise. For example, consider an instance with 20 vertices and 5 d-branch vertices. A centrality measure ranks its vertices, which are checked in the exact solution if they are d-branch. The first two and the last three ranked vertices are d-branch vertices. So, the Spread Ranking value equals $(1*1+1*2+1*18+1*19+1*20)/5$ or 12.

Ranking Loss (RM3): indicates the number of times non-d-branch vertices are ranked before d-branch vertices. It is given by Ranking Loss $= \frac{1}{|Y||\bar{Y}|}|\{(v,w)|r(v) > r(w), (v,w) \in Y \times \bar{Y}\}|$. In this case, if no non-d-branch vertices are in front of the d-branches, the answer is 0. Otherwise, if all non-d-branch vertices are in front of d-branches, the answer is 1. Consider the previous example where the first two and the last three ranked vertices are d-branch vertices. There are 15 non-d-branch vertices ranked before vertices ranked 18, 19, and 20 and no non-d-branch nodes before nodes ranked 0 and 1. The Ranking Loss value equals $45/(5*15)$ or 0.6.

Average Precision (RM4): It computes for each d-branch vertex v the proportion of d-branch vertices ranked from its rank position upwards and calculates the average of these values. Thus, if all d-branch vertices are ranked in the top-ranked positions, the value of this measure is equal to 1. It is given by Average Precision $= \frac{1}{|Y|}\sum_{v\in Y}\frac{|\{w|w\in Y, r(w)\leq r(v)\}|}{r(v)}$. Considering the previous example, we have for the vertice ranked 1 only itself as d-branch ranked from its position. For node ranked 2, we have 2, for node ranked 18, we have 3, for node ranked 19, we have 4; and for node ranked 20, we have 5. Therefore, Average Precision equals to $((1/1)+(2/2)+(3/18)+(4/19)+(5/20))/5) = 0.52$.

3.3. Results

Two datasets widely used in the context of the minimum d-branch problem, for $d = 2$, were considered for the computational experiments. They were proposed by Carrabs et al. [2013] and Merabet et al. [2018]. From now on, we will refer to the datasets respectively by **Carrabs** and **Merabet**.

The dataset **Carrabs** has 25 instances for each graph size ranging from $n = 20$ to $n = 1000$. The dataset **Merabet** has three groups (called $ias1$, $ias2$, and $ias3$) of 30 instances for each

graph size ranging from $n = 50$ to $n = 800$. Both are essentially composed of sparse random graphs.

The instances dataset files and the optimal solutions for all instances were provided by the authors of Moreno et al. [2018]. The available code for finding optimal solutions is in C++ language, the centralities were computed using Igraph package [Csardi and Nepusz, 2006], and the meta-measures were implemented using C language.

Figure 1 presents the F-score results for each centrality, considering the datasets **Carrabs** (on the top-left) and **Merabet** (ias1, on the top-right, ias2, on the bottom-left and ias3, on the bottom-right). Each 5-colour stacked column brings the average of the F-score values obtained for the set of instances for each value of n of each dataset. The colours of the columns represent the different centralities as indicated in the figure legend.

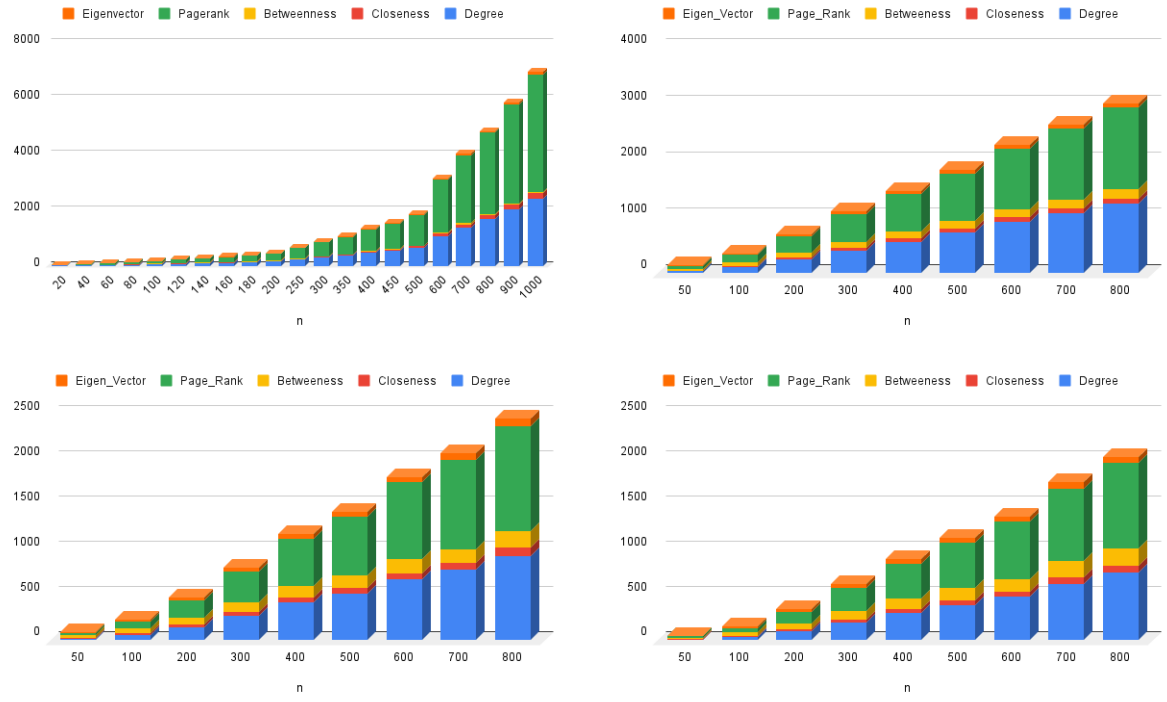

Figure 1: F-score results. Top-left: results for 25 Carrabs instances for each $n = 20$ to 1000. Top-right, bottom-left, and bottom-right: results for 30 instances for each $n = 50$ to 800 of groups ias1, ias2, and ias3.

The F-score results in Figure 1 show a pattern related to the type of centrality for discriminating the d-branch vertices. PageRank presented the larger values of the F-score, indicating that this centrality is the best in detecting the d-branch vertices for all instances of the Carrabs dataset, followed by degree. The only exception was for the group of instances of 20 nodes, for which Betweenness has slightly larger results. Closeness and eigenvector presented the worst values, showing that they are irrelevant in classifying the vertices as d-branch. Similar behaviour was obtained for the F-score computed for Merabet groups of instances. PageRank did not get the best results for instances of 50 nodes of groups $ias1$ and $ias2$. We also highlight that the differences in discrimination strength increase when the instances sizes are larger, as we can observe for all figures for instances with 200 or more vertices.

The higher the value of the F-score, the higher the ratio value (see Section 2.2.1) that defines it. This happens when the numerator of this ratio is greater than the denominator, indicating that the discrimination between interclass elements is greater than intraclass elements. This reinforces the idea that PageRank discriminates well between d-branch and non-d-branch vertices.

Figures 2 and 3 present the results for the ranking measures RM1 to RM4, for each centrality, considering the datasets **Carrabs** and **Merabet**. Each 5-colour stacked column brings the average of the ranking measure values obtained for the set of instances for each n. The colours of the columns represent the different centralities (as indicated in the figure legend).

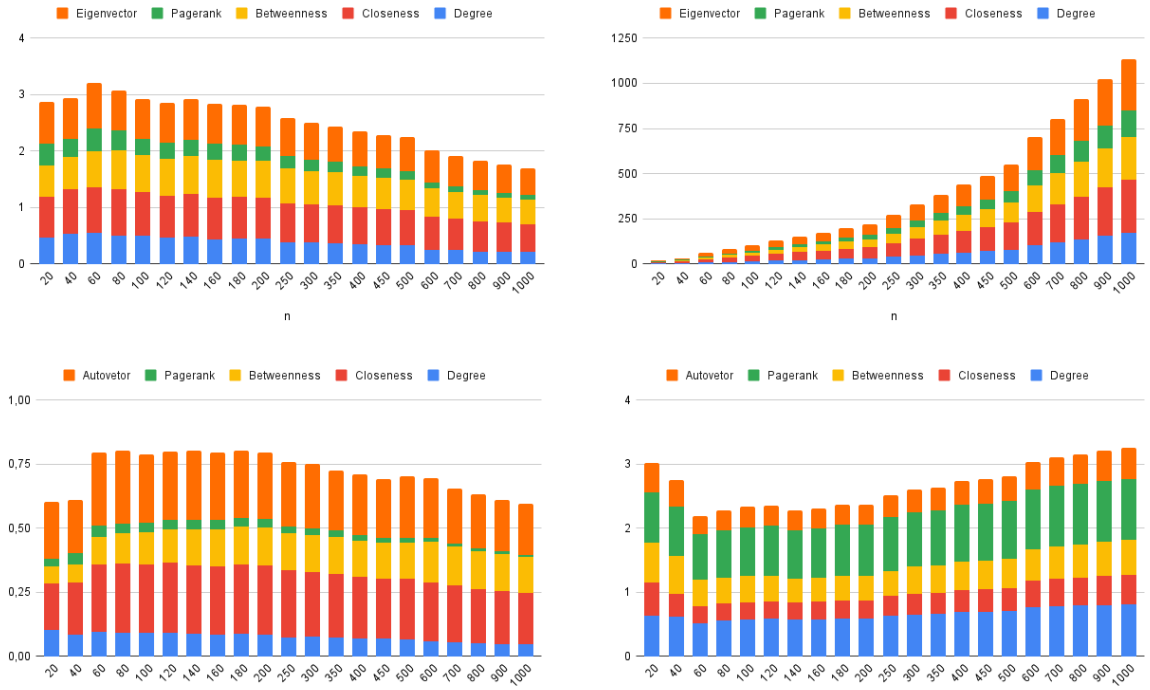

Figure 2: Ranking Measures RM1 (to the left and above), RM2 (to the right and above), RM3 (to the left and below), and RM4 (to the right and below) results for dataset Carrabs.

We observe the same behaviour for the ranking measure for both datasets. As described in Section 3.2.2, small values for RM1, RM2, and RM3 and great values for RM4 denote that the centrality may be used to distinguish the d-branch vertices. The best results are for PageRank, followed by degree, and the worst results alternate between closeness and eigenvector. In the Carrabs dataset, the second better results are achieved by Betweenness, instead of degree, only for RM2 and RM3 and for instances with 20 and 40 vertices. In the case of the Merabet dataset, this also occurs for RM2 and RM3, but only for instances with 50 nodes of groups ias2 and ias3 and RM4 for ias3.

To corroborate the results obtained, Table 1 and Figure 4 bring data from two example instances, each one of each dataset. One has 100 vertices and 114 edges from the Carrabs dataset; another has 300 vertices and 326 edges from the Merabet dataset (ias1). Table 1 shows the values of F-score and ranking measures obtained for each centrality and considering each instance. The rows in the table were sorted in descending F-score order since higher F-score values indicate that centrality tends to discriminate d-branch vertices better. We observe that ranking measures follow

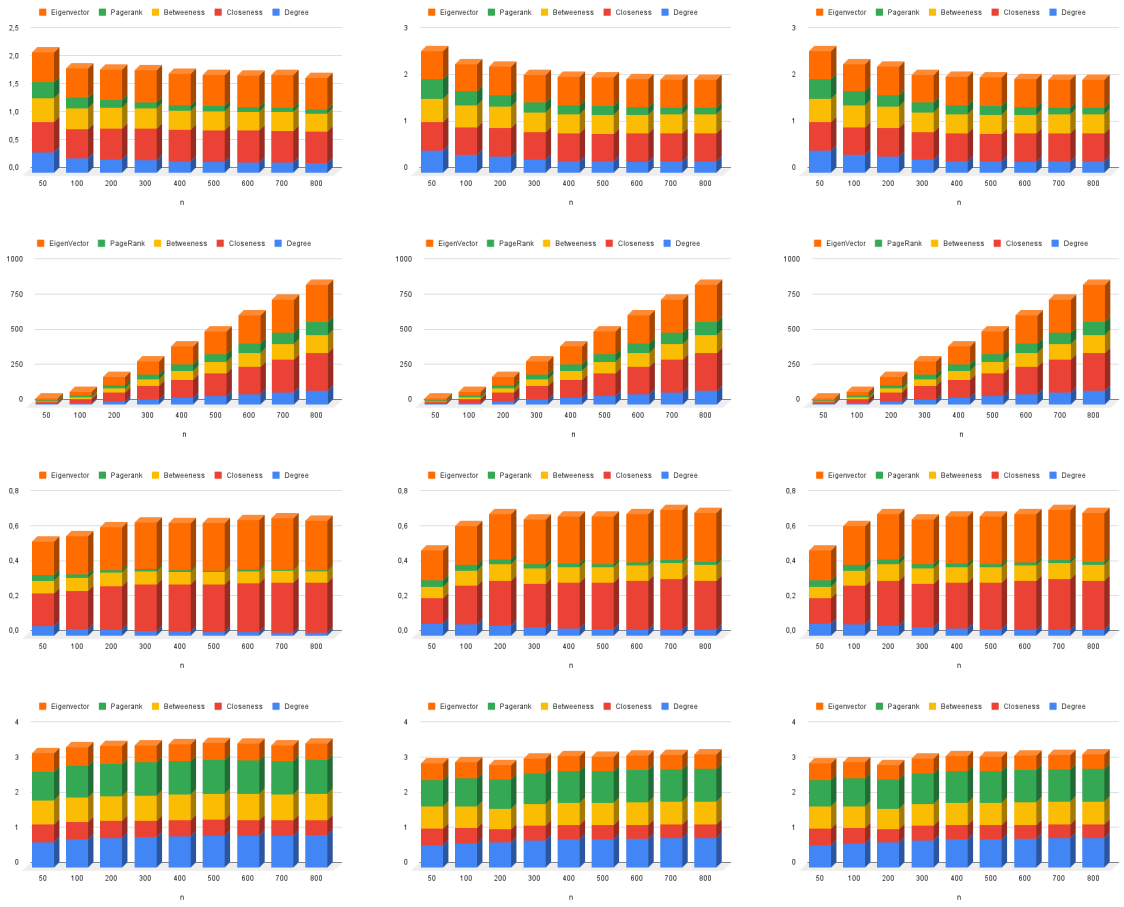

Figure 3: Ranking Measures RM1 to RM4 results (top-down rows) for dataset Merabet. On the left, ias1; in the middle, ias2 and on the right, ias3))

the F-score for both instances, specifically for the best result, PageRank (darker blue row); for the second best result, Degree (lightest blue row); and for the worst result, Eigenvector (red row). The Closeness and Betweenness centralities change positions when we compare the two instances.

Carrabs 100_114						Merabet_300_326					
Centrality	F-score	RM1	RM2	RM3	RM4	Centrality	F-score	RM1	RM2	RM3	RM4
PageRank	274.22	0.08	14.62	0.02	0.94	**PageRank**	570.47	0.06	34.74	0.01	0.97
Degree	167.83	0.19	16.65	0.05	0.82	**Degree**	468.41	0.17	38.48	0.02	0.91
Closeness	20.83	0.50	25.39	0.16	0.51	Betweeness	110.32	0.32	49.91	0.07	0.72
Betweenness	15.94	0.35	21	0.10	0.60	Closeness	58.03	0.48	89.68	0.24	0.53
Eigenvector	15.49	0.54	28.96	0.21	0.49	**Eigenvector**	42.64	0.59	99.94	0.29	0.49

Table 1: F-score and ranking values for Carrabs instance with $n = 100$ and $m = 114$ (on the left) and for Merabet (ias1) instance with $n = 300$ and $m = 326$ (on the right)

Figure 4 brings, for each instance, the ranking of the k d-branch vertices of the optimal solution, considering each centrality. The top row represents the ranking obtained by Eigenvector, followed by Betweenness, Closeness, Degree, and PageRank in the last row. Each row is divided

into n slots, representing the n ranking positions in ascending order from left to right. The orange slots represent the positions of d-branch vertices. The blue column indicates k ranking position, respectively equal to 26 for the 100-node instance and 65 for the 300-node instance. We can observe that PageRank ranks in the best positions in practically all the d-branch vertices. In the case of the 100-node instance, only 2 vertices, and in the case of the 300-node instance, only 4 vertices are outside the first k positions. We also observe that the degree is in the second best ranking and the Eigenvector is in the worst, coinciding with the behavior observed in Table 1.

The evaluation of Figure 4 in conjunction with Table 1 gives strong evidence that the meta-measures were able to indicate the capacity of the centrality measures to identify d-branch vertices.

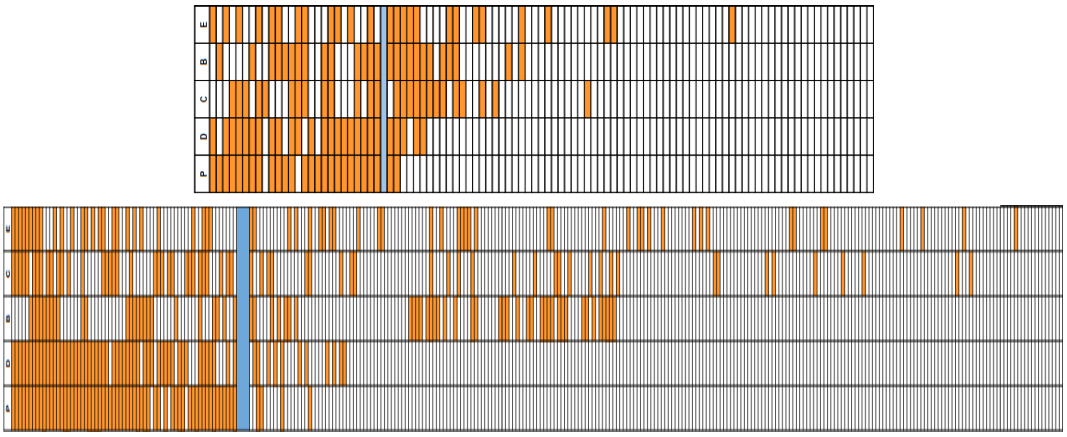

Figure 4: On the top, Carrabs instance with $n = 100$ and $m = 114$ and on the bottom, Merabet (ias1) instance with $n = 300$ and $m = 326$

4. Conclusions

In this work, we investigate if centrality measures on graphs can be used to identify elements belonging to solutions of combinatorial optimization problems modeled by graphs. This investigation was instantiated to the d-branch vertices minimization problem.

Graph centralities are closely related to the graph's topological information. We evaluated five well-known graph centralities to determine if they can identify vertices d-branch in the graph problem using five meta-measures inherited from Data Science.

The results show that the meta-measures can indicate the centralities' capacity to identify d-branches. PageRank and Degree outperform the other three centralities in this skill.

In future work, we aim to evaluate similar centralities for other optimization problems and analyze the use of centralities within heuristics to solve optimization problems.

References

Alozie, G. U., Arulselvan, A., Akartunali, K., and Jr., E. L. P. (2022). A heuristic approach for the distance-based critical node detection problem in complex networks. *Journal of the Operational Research Society*, 73(6):1347–1361.

Carrabs, F., Cerulli, R., Gaudioso, M., and Gentili, M. (2013). Lower and upper bounds for the spanning tree with minimum branch vertices. *Computational Optimization and Applications*, 56: 405–438.

Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, Complex Systems:1695. URL https://igraph.org.

Frinhani, R., Moreira de Carvalho, M., and Soma, N. Y. (2018). A PageRank-based heuristic for the minimization of open stacks problem. *PLoS ONE*, 13(8):e0203076.

Gargano, L., Hell, P., Stacho, L., and Vaccaro, U. (2002). Spanning trees with bounded number of branch vertices. In Widmayer, P., Eidenbenz, S., Triguero, F., Morales, R., Conejo, R., and Hennessy, M., editors, *Automata, Languages and Programming*, p. 355–365, Berlin, Heidelberg. Springer Berlin Heidelberg. ISBN 978-3-540-45465-6.

Herrmann, S., Ochoa, G., and Rothlauf, F. (2018). Pagerank centrality for performance prediction: the impact of the local optima network model. *Journal of Heuristics*, 24(3):243–264. ISSN 1572-9397.

Lozano, M. and Trujillo, H. M. (2019). Optimizing node infiltrations in complex networks by a local search based heuristic. *Computers & Operations Research*, 111:197–213. ISSN 0305-0548. URL https://www.sciencedirect.com/science/article/pii/S0305054819301716.

Martin, C. and Niemeyer, P. (2019). Influence of measurement errors on networks: Estimating the robustness of centrality measures. *Network Science*, 7(2):180–195.

Merabet, M., Desai, J., and Molnár, M. (2018). A generalization of the minimum branch vertices spanning tree problem. In *International Symposium in Combinatorial Optimization*.

Moreno, J., Frota, Y., and Martins, S. (2018). An exact and heuristic approach for the d-minimum branch vertices problem. *Computational Optimization and Applications*, 71(3):829–855. ISSN 1573-2894. URL https://doi.org/10.1007/s10589-018-0027-x.

Pereira, R. B., Plastino, A., Zadrozny, B., and Merschmann, L. H. (2018). Correlation analysis of performance measures for multi-label classification. *Information Processing & Management*, 54 (3):359–369. ISSN 0306-4573. URL https://www.sciencedirect.com/science/article/pii/S0306457318300165.

Sharma, P., Bhattacharyya, D. K., and Kalita, J. K. (2016). Centrality analysis in ppi networks. In *2016 International Conference on Accessibility to Digital World (ICADW)*, p. 135–140.

Zaki, M. J. and Wagner Meira, J. (2020). *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*. Cambridge University Press, Second edition. ISBN 978-1108473989.

Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., and Liu, H. (2010). Advancing feature selection research. *ASU Feature Selection Repository Arizona State University*, p. 1–28.