UM ALGORITMO BASEADO EM COBERTURA DE VÉRTICES PARA REMOÇÃO DE AMBIGUIDADES EM REDES DE COLABORAÇÃO CIENTÍFICA

Hugo Henrique de Melo Kling

Projeto de Graduação apresentado ao Curso de Engenharia de Computação e Informação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientadores: Daniel Ratton Figueiredo
Janaina Sant'Anna Gomide

Rio de Janeiro
Setembro de 2016

UM ALGORITMO BASEADO EM COBERTURA DE VÉRTICES PARA REMOÇÃO DE AMBIGUIDADES EM REDES DE COLABORAÇÃO CIENTÍFICA

Hugo Henrique de Melo Kling

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA DE COMPUTAÇÃO E INFORMAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO DE COMPUTAÇÃO E INFORMAÇÃO.

Examinado por:

Prof. Daniel Ratton Figueiredo, Ph.D.

Profa. Janaina Sant'Anna Gomide, M.Sc.

Profa. Marta Lima de Queirós Mattoso, D.Sc.

Prof. Ricardo Guerra Marroquim, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2016

Henrique de Melo Kling, Hugo

Um Algoritmo Baseado em Cobertura de Vértices para Remoção de Ambiguidades em Redes de Colaboração Científica/Hugo Henrique de Melo Kling. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2016.

X, 27 p. 29, 7cm.

Orientadores: Daniel Ratton Figueiredo
Janaina Sant'Anna Gomide

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia de Computação e Informação, 2016.

Bibliography: p. 26 – 27.

I. Ratton Figueiredo, Daniel *et al.* II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia de Computação e Informação. III. Título.

A Deus

Agradecimentos

Primeiramente, agradeço aos meus pais e a meu irmão por todo o tempo que estiveram comigo. Pelas situações que passamos juntos, e pela força de vontade que sempre nos move adiante, a despeito das adversidades da vida. Em especial ao meu falecido pai, que tanto influenciou no meu jeito de ser – e que espero poder reencontrar quando eu partir.

Agradeço aos meus familiares e meus amigos, que mesmo à distancia sempre torceram por mim. À minha namorada, que por tanto tempo esteve ao meu lado e sempre me apoiou. A todos os meus colegas de curso, aos quais possuo grande apreço e que me ajudaram tanto durante essa jornada.

Aos meus professores, que mostraram o real valor dessa nobre profissão – sem os quais não chegaria onde estou hoje. Em especial aos meus orientadores, que sempre me ajudaram e me motivaram na jornada acadêmica.

"All have their worth and each contributes to the worth of the others."
— J.R.R. Tolkien

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação e Informação.

UM ALGORITMO BASEADO EM COBERTURA DE VÉRTICES PARA REMOÇÃO DE AMBIGUIDADES EM REDES DE COLABORAÇÃO CIENTÍFICA

Hugo Henrique de Melo Kling

Setembro/2016

Orientadores: Daniel Ratton Figueiredo
Janaina Sant'Anna Gomide

Curso: Engenharia de Computação e Informação

Redes vem sendo cada vez mais utilizadas para representar diversos tipos de estruturas, tais como redes de informação (hiperlinks na web), redes sociais (amizades no Facebook) e redes biológicas (proteínas na célula). Em muitos cenários, os vértices da rede possuem rótulos que servem como identificadores dos objetos que representam. Neste contexto, surge o problema de ambiguidade estrutural, que consiste em determinar vértices equivalentes na rede - nós com identificadores diferentes que representam o mesmo objeto, ou nós com identificadores iguais que representam objetos diferentes.

Este trabalho tem como objetivo propor e avaliar um algoritmo para identificação de ambiguidades de nomes no contexto de redes de colaboração científica, no caso onde um mesmo indivíduo é representado na rede por mais de um vértice - ou seja, rótulos diferentes para o mesmo objeto, como por exemplo, "Bill Gates" e "William Henry Gates". Em particular, o trabalho tem como foco redes de colaboração induzidas por publicações científicas de um único autor – denominadas Egonets – onde o mesmo possui mais de um rótulo (nome) distinto em suas publicações.
Palavras-chave: Redes de Colaboração Científica, Ambiguidade, Egonets, Cobertura de Vértices.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

A VERTEX COVER BASED ALGORITHM FOR AMBIGUITY REMOVAL IN SCIENTIFIC COLLABORATION NETWORKS

Hugo Henrique de Melo Kling

September/2016

Advisors: Daniel Ratton Figueiredo
Janaina Sant'Anna Gomide

Course: Computer and Information Engineering

Networks are being increasingly used to represent various types of structures, such as information networks (hyperlinks on the web), social networks (friends on Facebook) and biological networks (proteins in the cell). In many scenarios, the network vertices have labels that serve as identifiers of the objects they represent. In this context, the structural ambiguity problem arises of determining equivalent vertices in the network - nodes with different identifiers that represent the same object, or nodes with the same identifiers that represent different objects.

This work proposes and evaluate an algorithm for name ambiguities identification in the context of scientific collaboration networks, in the case where one individual is represented on the network by more than one vertex - that is, different labels for the same object (for example, "Bill Gates" and "Henry William Gates"). In particular, this work focuses on collaborative networks induced by scientific publications of a single author - called Egonets - that have more than one distinct label (name) in their publications.
Keywords: Scientific Collaboration Networks, Entity Ambiguity, Egonets, Vertex Cover.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Ambiguity Problem

Nowadays, with increasingly advancements in computer technology in several different areas – such as storage capacity, processing power and algorithm design – storing and processing massive amounts of data is now possible. For instance, a company seeking to improve its management of data may decide to digitalize all of its paper documents, or a hospital that chooses to Store patient data digitally instead of using traditional paper records.

In this context, a notorious problem arises from the need of analysing great volumes of data: name ambiguity. In the hospital case, a patient may be registered with a slight error in his or her name. This error can possibly go on unnoticed for several years, and if that patient ever returns, it is also possible that he or she gets registered again with the correct name in the system (instead of correcting the old profile). As a consequence, there will be two profiles in the database that represent the same person in real life – a perfect example of the name ambiguity problem. A similar problem can occur if she registers in two or more hospitals that decide to share data. It is important to notice that there are two underlying and fundamentally different problems related to name ambiguity: in the first one, multiple distinct entities may have or be associated to one single name or label; the second one, multiple names or labels are related to one single entity.

Several other examples of this problem can be given: a city may appear in web pages with multiple names (Rio de Janeiro, for instance, that some call it just Rio or "Cidade Maravilhosa"); a researcher can appear in bibliographical databases with different names. These examples show the great relevance of name ambiguity in data analysis, and there has been many attempts to design systems over the past years.

1.2 Objective and Organization

This final course project (TCC) is part of the ongoing research work of Janaina Gomes, a doctoral student at PESC/COPPE under the supervision of prof. Daniel R. Figueiredo. The present work aims to address the entity ambiguity problem, proposing and evaluating an algorithm based on network features for identifying name ambiguities in scientific collaboration networks, focusing on the second type of ambiguity: in which a single individual is represented in the network by multiple names.

In order to not assume any previous knowledge by the reader, this work also introduces some basic concepts needed to fully understand the problem and the proposed algorithm. After this brief introduction, the methodology adopted will be described, as well as important decisions that were taken in the design of the algorithm. Afterwards, the evaluation process and results are discussed, taking into consideration the characteristics of the studied network and other proposed strategies for this problem.

1.3 Related Work

The ambiguity problem in the context of bibliographic records is considered to not be fully solved yet. Several different solutions have been proposed and reported [8] [6], that include: manual inspection [5], feature-based heuristics using context (such as name and institution) [19], probabilistic classifiers [14], machine learning

techniques [11, 15, 20] and also algorithms based on network features [2, 10, 18]. There is a significant scientific interest about this subject, as the name ambiguity problem directly impacts the analysis of large-scale co-authorship networks [13].

In this work, name ambiguity in collaboration networks is studied by creating collaboration networks from real data, collected from a publicly available database and processed in order to generate egonets. Then, an algorithm based on network features, designed to detect name ambiguities in a researcher's set of publications, is proposed and evaluated.

Chapter 2

Basic Concepts

2.1 Graphs and Network Science

Network Science is an interdisciplinary field [17] that basically studies sets of elements and the relationships between them. There are an uncountable number of different sets, such as sets of people, words, proteins, and so on. Relationships can be seen as a property that "connects" two elements, like friendship and collaboration, for example (in the case of a set that contains people).

The mathematical tool used to represent networks is a graph, that can be defined as an ordered pair of sets $G = (V, E)$, in which V represents the set of vertices and E the set of edges – in network science, they are called nodes and links, respectively. Each element of these sets can also present properties such as labels and values (edge weights, for instance).

An edge, for generic purposes, can be seen as an ordered or unordered subset of V. Though, a more strict definition will be used in this work in order to simplify the problem analysis: the definition of an edge will be an unordered pair of elements (x, y) so that $x, y \in V$, $x \neq y$. Note that this definition, in the context of general network study, can be insufficient to represent some types of relationships (e.g. motherhood, that intrinsically is not a symmetric relation). In this work, however, edges will represent co-authorship – a relationship that can be considered symmetric. With these definitions in mind, there are still some important concepts concerning

graph theory that need to be elucidated for a complete understanding of the proposed algorithm.

The first one is the concept of neighbourhood: a vertex x is said to be neighbour of another vertex y if $(x, y) \in E$ – and, since this is an unordered relation by previous definition, y is also neighbour of x. Therefore, The neighbourhood u of vertex x is the subset $u \subset V$ that contains all neighbours of x. With this definition, we can as well define degree of a vertex $d(x)$ as being the number of neighbours of x.

Then, we can define the path between two vertices x and y: a sequence p of edges that connects x and y. If such sequence exists, then we can say that there is a path between these vertices. For unordered edges (undirected graph), there can only exist a path between x and y if and only if there is also a path between y and x, or $p(x, y) \iff p(y, x)$. It is also useful to define the shortest path between two vertices, that is the path with minimum length, known as distance: on a unweighted graph, it is equivalent to the sequence of edges with the least number of elements, and when weighted edges are used, it is the path with the least total sum of weights. Another useful concept is the notion of connected component, that can be defined as a subset $C, C \subset V$ that $\forall x, y \in C, \forall z \notin C, \exists p(x, y)$ and $\nexists p(x, z)$. Basically, we have a path between any two elements within the connected component.

2.2 Collaboration Networks

A collaboration network consists in a group of entities (vertices) linked by some type of collaboration relationship (edges). For instance, in scientific collaboration networks these entities represent researchers and the edges, publication co-authorships. Thus, a link is created between two researchers if there is a publication that both appear as authors.

A simple process may be applied in order to create a scientific collaboration network from set of publication records: for each publication, increment the network by adding its authors in V if they were not there previously, and insert into E edges containing each possible pair of authors in that publication, if these relationships

did not already exist.

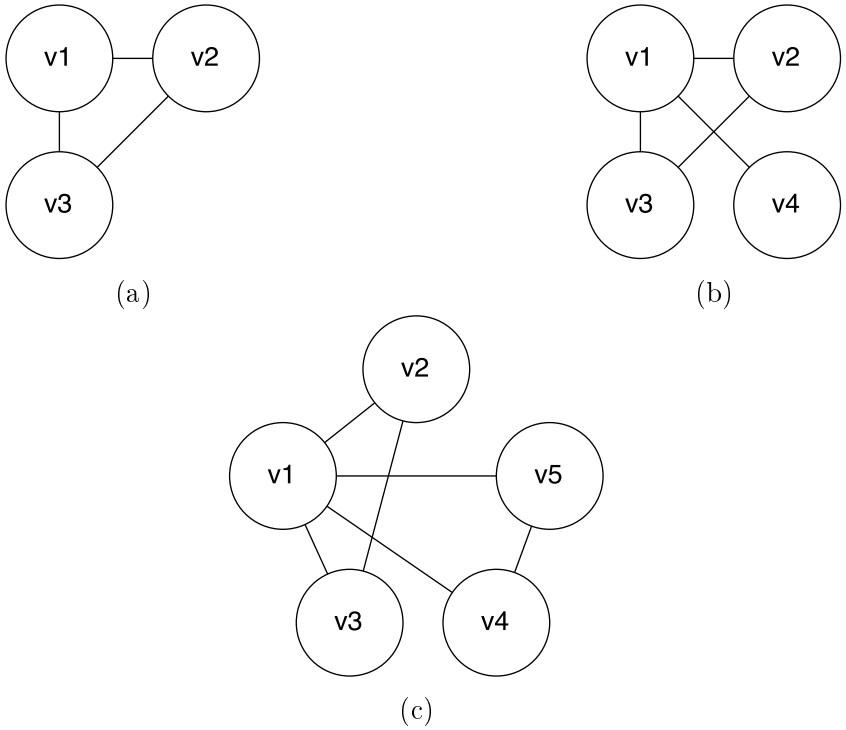


Figure 2.1: Example of a collaboration network creation process.

This process is illustrated in Fig. 2.1 using as example a set of three publications $p1 = \{v1, v2, v3\}$, $p2 = \{v1, v4\}$ and $p3 = \{v1, v4, v5\}$. With the first publication, vertices $v1, v2$ and $v3$ were added in the network among all their relationships (edges $(v1, v2), (v1, v3)$ and $(v2, v3)$). Then, with publication $p2$ the vertex $v4$ was added – since $v1$ was already present in the network, there was no need to insert it again – along its relationship with vertex $v1$. Finally, vertex $v5$ and edges $(v1, v5), (v4, v5)$ were included in the graph as result of the last publication $p3$ – and the resulting collaboration network created from these three publications in Fig. 2.1c.

2.3 Egocentric Networks

According to [3], an ego-network (egonet) is the neighbourhood of a focal vertex, called ego, together with the set of edges among members of the ego network. This is a fundamental concept used in this work, given the way collaboration networks will be generated from a set of scientific publications of a given person. For instance,

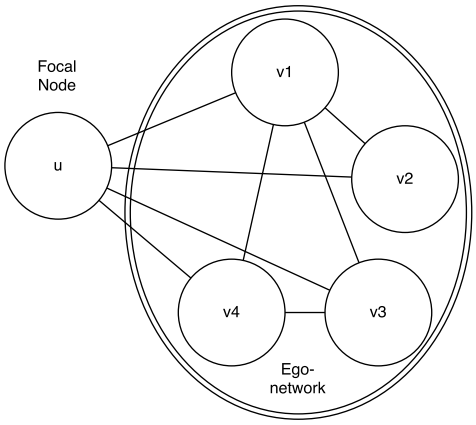


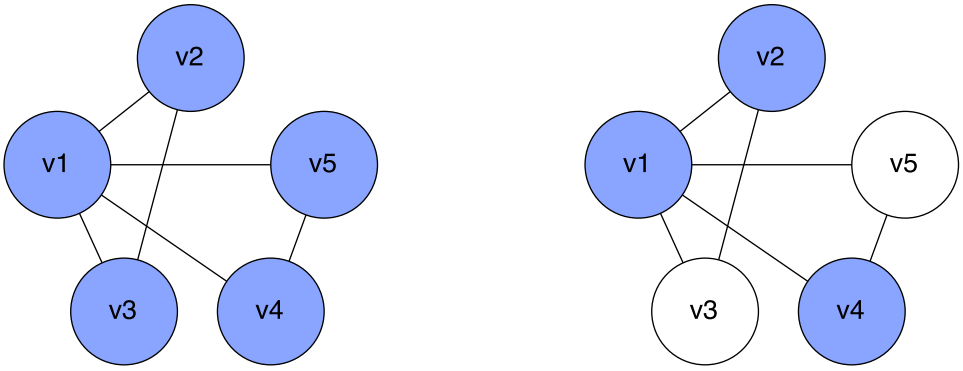
Figure 2.2: Ego-network example.

the collaboration network induced by all publications of a researcher is essentially an ego-centric network – thus, the process of generating a collaboration network from a large database of publications is tied with the concept of egonets.

In example Fig. 2.2 we can see the egocentric network of vertex v, which consists of $V = \{v1, v2, v3, v4\}, E = \{(v1, v2), (v1, v3), (v1, v4), (v3, v4)\}$.

2.4 Vertex Cover

A vertex cover of a graph $G = (V, E)$ can be defined as a subset $V' \subset V$ such that $\forall e \in E, \exists v \in V', v \in e$. This means that every edge in the graph is incident to at least one vertex in the cover. Though finding a vertex cover from graph G is a trivial problem, finding the minimum vertex cover – which is a vertex cover V' with the minimum number of elements – is a NP-complete problem [12].



(a) Trivial vertex cover.

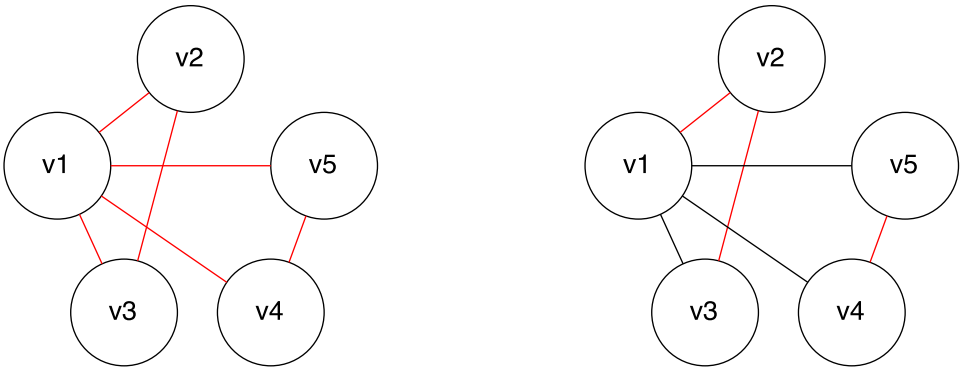
(b) Minimum vertex cover.

Figure 2.3: Vertex cover example.

It is illustrated in Fig. 2.3 a simple example of vertex cover. As shown in 2.3a, we can see the trivial case of a vertex cover in which $V' = V$. In 2.3b, one possible minimum vertex cover of the given graph.

2.5 Edge Cover

An edge cover of a graph $G = (V, E)$ can be defined as a subset $E' \subset E$ such that $\forall v \in V, \exists e \in E', v \in e$. This means that, for every vertex in the graph, there is an edge in the cover that is incident to it. Unlike the vertex cover problem, finding the minimum edge cover – that is an edge cover with minimum number of elements – can be achieved in polynomial time [9].



(a) Trivial edge cover. (b) Minimum edge cover.

Figure 2.4: Edge cover example.

It is illustrated in Fig. 2.4 a simple example of edge cover. As shown in 2.4a, we can see the trivial case of a edge cover in which $E' = E$. In 2.4b, one possible minimum edge cover of the given graph.

2.6 Clique

A clique q of a graph $G = (V, E)$ is a particular subset of vertices that is also a complete graph – this means that $\forall x, y \in q, \exists (x, y) \in E$. Finding the maximum clique of a graph, which is the clique with the largest amount of elements, has been proven to be a NP-complete problem [12]. The concept of a clique has long been

investigated [16], and its definition is particularly useful in the context of this work, as each publication induces a clique in the scientific collaboration network.

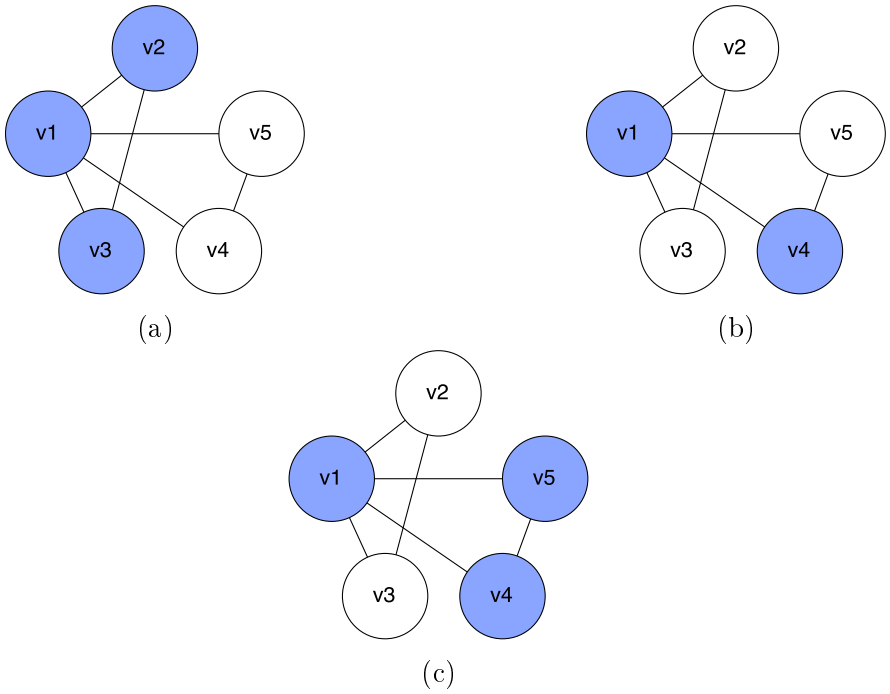


Figure 2.5: Cliques induced by publications.

As we can see in Fig. 2.5, three different cliques were induced by the publications listed as example in Section 2.2. The clique in 2.5a was induced by publication $p1$; $p2$ originated 2.5b and the last publication $p3$ induced 2.5c. The same process occurs when creating collaboration networks from a list of real publications.

Chapter 3

Methodology

3.1 Database Choice

The objective of this work is to address the name ambiguity problem in scientific collaboration networks from the point of view of egonets. But, before any attempt in designing a solution, a real network must be obtained in order to elaborate ideas and test propositions.

Theoretical conclusions about networks can be achieved by studying artificially created networks, originated from mathematical models [7]. However, this work aims at solving a relatively common problem (entity ambiguity) using real world data and, therefore, we obtained and constructed a network based on a large available scientific publications repository.

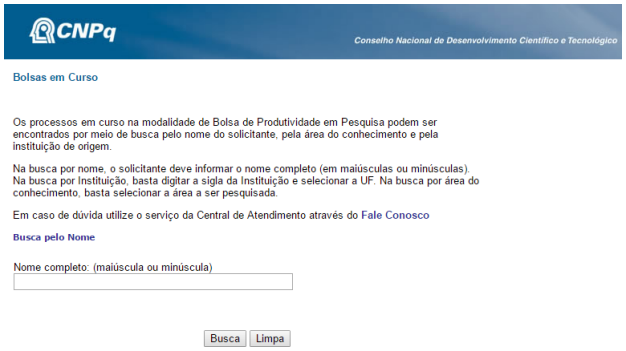
Google Scholar's database does present these characteristics, and hence was chosen to be the data source for this work. It maintains a profile for each researcher, which lists all profile owner's publications – that were collected from university repositories, publishers and scholarly websites – along with other information, such as full name and affiliation. However, a profile can only become publicly accessible after approval by its owner.

Figure 3.1: Google Scholar page. "Standing on the shoulders of giants" famous expression at the bottom.

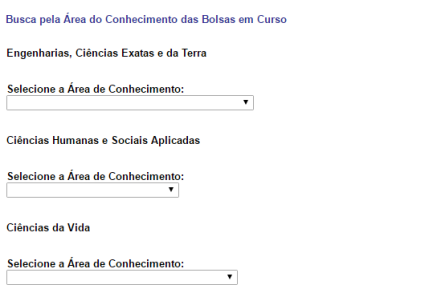
3.2 Data Extraction

Since the amount of data in this database is enormous, we decided to work with a subset of publications since we assumed that similar results would probably still be attainable in a smaller scale network.

Thus, in this work we decided to use a subset of individuals containing all CNPq's current researchers that receive scientific funding (research fellowship). This process was done by manually gathering all names contained in each knowledge area available in CNPq's website in the year of 2016.



(a) Search by name.



(b) List of researchers found in the selected area.

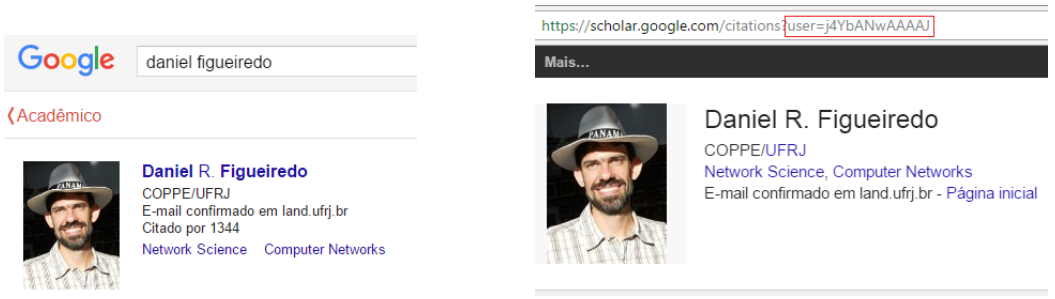
Figure 3.2: CNPQ search tool.

In Fig. 3.2 we can see both types of author search in the website. The method used in this work was manually gathering all authors names listed in each area available (Fig. 3.2b), dividing them among three different groups (Life Sciences, Human Sciences and Exact Sciences).

Table 3.1: Number of researchers by group.

Group	Number
Life Sciences	5759
Human Sciences	3213
Exact Sciences	4501
Total	13473

In Table 3.1, the number of researchers found by group (Life, Human and Exact Sciences). Even after this initial filter, the total number of researchers (13473) was still considered high for this study – then, this number was further reduced after selecting only the Exact Sciences group.



(a) Search query example.

(b) A researcher's profile page, and the corresponding id at the top.

Figure 3.3: Google Scholar identification process.

After these filters were applied, the selected names where then submitted to queries in Google Scholar's website as seen in Fig. 3.3 in order to find each corresponding identification code – which leads to the researcher's profile page. The web page query process was taking significant time to be concluded, and hence we decided to further diminish the number researchers in our study to about 1000 individuals. Then, some individuals did not have a profile in Google Scholar and were

thereby excluded of the process – in the end, 638 individuals among these presented a Google Scholar profile that could be detected. The next step, after gathering all possible researchers identifications, the website was once again queried for their publication pages.

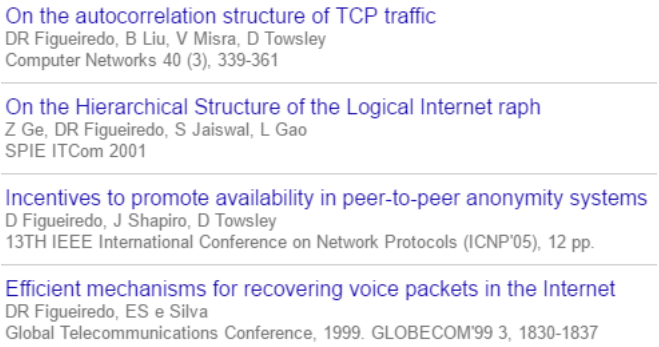


Figure 3.4: List of publications sample by a given author.

Since there was no way of knowing beforehand the exact number of publications of a particular researcher, next pages are requested as long as they presented publications. An example of some publications found in one of these pages can be seen in Fig. 3.4. Once an empty page was returned, e.g. Fig. 3.5, further page requests for that particular profile are ceased, and the process continues for another author.

Figure 3.5: Example of empty publications web page.

After all profile pages were gathered, we started the process of generating collaboration networks based on the author's publications – one network for each profile

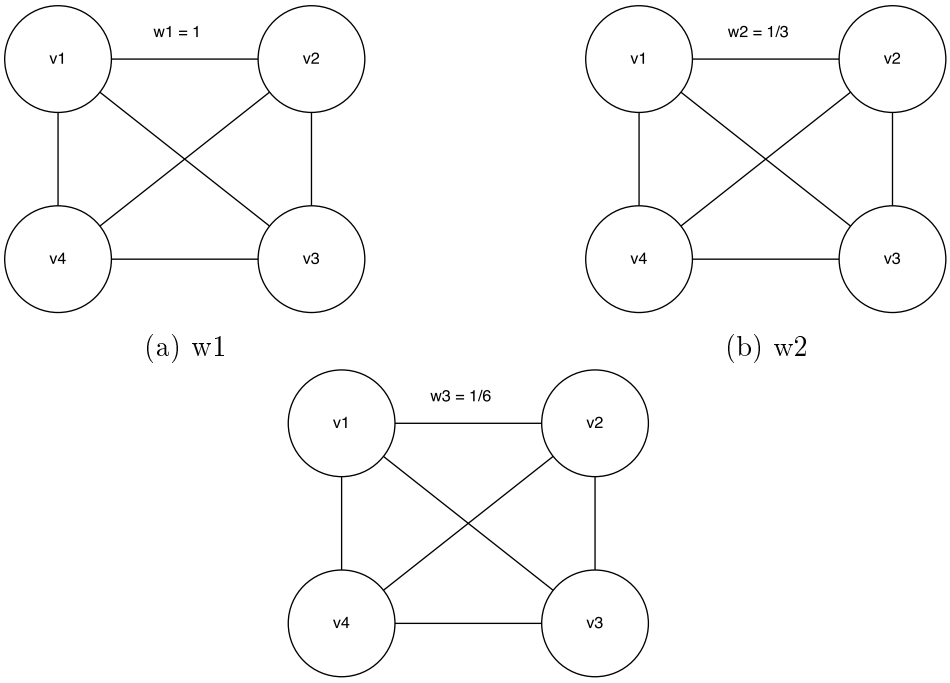
page, the egonet of the researcher. The algorithm to generate these networks starts as following: initialize the local egonet as a graph $G = (V, E), V, E = \{\}$. Then, for each publication p in the profile, do: for every author name $v1 \in p$, if $v1 \notin V$, insert $v1$ into V. Then, for every other author $v2 \in p$, if $(v1, v2) \notin E$, insert $(v1, v2)$ into E. Note that generates a clique of size n, that is the number of authors in publication p.

A few useful metrics were also stored for further algorithm development, such as three different weights for the edges: $w1$, that sums up all the times the edge appeared in publications; $w2$, that accumulates the inverse of the number of authors minus 1 for each publication containing that particular edge; and $w3$, that sums the inverse of $\binom{n}{2}$, where n is the number of authors in each publication containing that edge. A matrix was also created in order to store the information regarding which labels appeared in the exact same publications.

It is important to notice that these metrics can be useful for gathering important information about the network: for instance, the sum of $w1$ for a particular edge corresponds to the number of times two authors collaborated together; the sum of $w2$ for all edges that are incident to a particular vertex corresponds to the number of publications of that author; and finally, the sum of $w3$ for all edges belonging to a particular connected component corresponds to the number of publications that generated that component.

To exemplify the collaboration network creation process with real data, we present a egonet of John Von Neuman based on a few publications gathered from his Google Scholar profile page. In Table 3.2 there is a list of publications that will be considered in this example. For each co-author name an identifier is given, Table 3.3a, and the edges weights are calculated, Table 3.3b. The John Von Neumans egonet are shown in Figure 3.7a.

The corresponding egonets created for each profile where then imported by the Python framework *Networkx* [1]. This tool was also used during this work in the process of analysing and manipulating the networks, and evaluating the designed



(a) w1 (b) w2

(c) w3

Figure 3.6: Example of the different metrics induced by one publication.

algorithm with the collected data.

3.3 Algorithm Development

After setting up all the environment needed to work with the extracted collaboration networks, some thoughts were put into the problem itself before any further progress. This stage was crucial for determining strategies that were tested in order to design an effective algorithm to remove ambiguities in these networks.

The first point that needs to be brought into attention is the fact that those networks were generated in a very particular way: by iterating over a list of publications, the profile owner presumably will only appear in each publication a single time. It can not be assumed that this appearance will always have the same name – though intuitively there is a name that mostly represents a certain researcher. For instance, in Figure 3.4, the name "DR Figueiredo" appears in three publications, whereas "D Figueiredo" appears only once.

A consequence of the first observation is that it is very unlikely that two vertices

Table 3.2: Some of John Vohn Neumanns publications listed in his Google Scholar profile.

Authors	Title
AW Burks, HH Goldstine, **J Von Neumann**	Preliminary discussion of the logical design of an electronic computing instrument
S Chandrasekhar, **J Von Neumann**	The Statistics of the Gravitational Field Arising from a Random Distribution of Stars
BI Hart, **John von Neumann**	Tabulation of the probabilities for the ratio of the mean square successive difference
BO Koopman, **J V Neumann**	Dynamical systems of continuous spectra
HH Goldstine, **J V Neumann**	Blast wave calculation
J Von Neumann, RH Kent, HR Bellinson, BI Hart	The mean square successive difference
D Hilbert, **J Neumann**, L Nordheim	Uber die grundlagen der quantenmechanik
R Zeller, **J Neumann**	Calibration-test member for a coordinate-measuring instrument

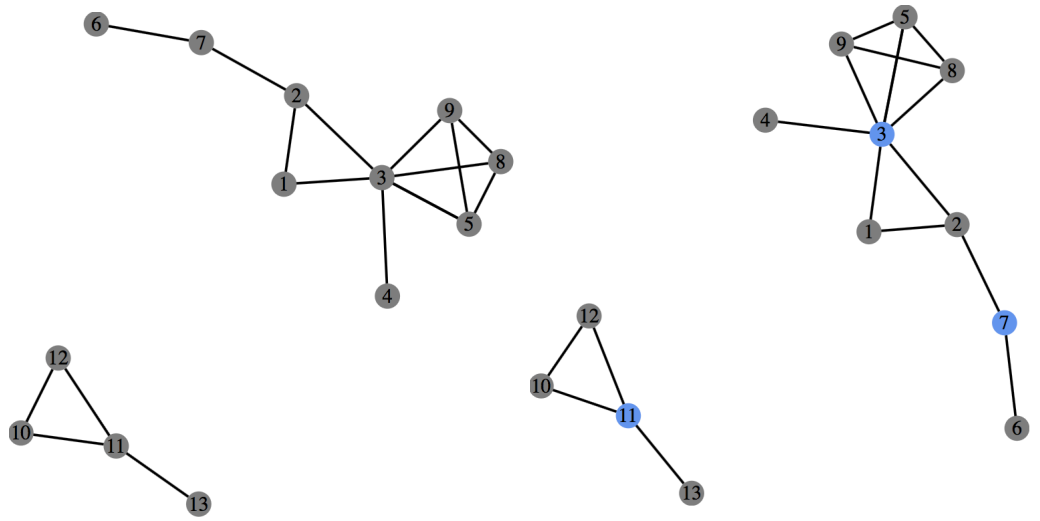


Figure 3.7: Exemple of John Von Neumans egonet. All nodes in (a); in (b), the ambiguous nodes in blue.

Table 3.3: In (a) the identifiers of the co-authors from John Von Neumanns publications and in (b) the weights of each edge.

(a)

Name	*id*
AW Burks	1
HH Goldstine	2
J Von Neumann	3
S Chandrasekhar	4
BI Hart	5
BO Koopman	6
J V Neumann	7
RH Kent	8
HR Bellinson	9
D Hilbert	10
J Neumann	11
L Nordheim	12
R Zeller	13

(b)

$Edge(id1, id2)$	$Weights[w1, w2, w3]$
(1,2)	[1,0.5,0.33]
(1,3)	[1,0.5,0.33]
(2,3)	[1,0.5,0.33]
(4,3)	[1,1.0,1.0]
(5,3)	[1,1.0,1.0]
(6,7)	[1,1.0,1.0]
(2,7)	[1,1.0,1.0]
(3,8)	[1,0.33,0.17]
(3,9)	[1,0.33,0.17]
(3,5)	[1,0.33,0.17]
(8,9)	[1,0.33,0.17]
(8,5)	[1,0.33,0.17]
(9,5)	[1,0.33,0.17]
(10,11)	[1,0.5,0.33]
(10,12)	[1,0.5,0.33]
(11,12)	[1,0.5,0.33]
(13,11)	[1,1.0,1.0]

that represent the same author are neighbours in the collaboration network. Otherwise, there would exist a publication where the same person appears more than once – if the database presents any sort of maintenance and error checking, this should be considered a rare event.

With these two facts in mind, the idea of using a vertex cover based approach in the design of the proposed algorithm became more evident. The idea behind this choice is that, in the end of the algorithm, the cover would contain all vertices that in fact represent the owner of the profile. Although solving the minimum vertex cover problem is NP-complete, this work adopts a very simple greedy algorithm as a workaround for this limitation: pick the vertex with the largest number of neighbours and add it to the cover, removing it and its neighbourhood from the graph; repeat it until there are no vertices left in the graph. This is a slight different version of vertex cover definition as seen in Section 2.4: for $u \in V', \forall v \notin V', \exists (v, u) \in E$. This basically states that every vertex that is not included in the cover is neighbour of at least one vertex that is in the cover. In fact, this approach worked well given the

way collaboration networks were generated in this work.

This initial version of the algorithm is indeed very simple, but it has a fundamental limitation: it can not solve ties in the process of choosing the vertex with largest degree – as a matter of fact, in many scenarios more than one vertex has the largest degree.

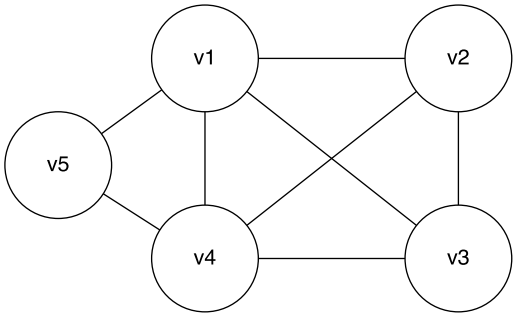


Figure 3.8: Example of a tie in the algorithm (v1 and v4).

Then, other criteria were introduced in order to refine the algorithm, such as: greater number of publications (given by the sum of $w2$ for all incident edges on a vertex), greater number of common neighbours in relation to other vertices in the cover, and the least sum of $w3$ for all incident edges – this criterion is only used after the two previous ones, and it tries to capture the idea of prioritizing vertices that normally appear in publications with a high number of co-authors.

But still, even after these criteria were implemented, there were cases in which the algorithm could not complete its execution. These were further investigated, and the results showed that there were, in several cases, connected components that were being generated by a single publication (or multiple publications with the same group of authors); and also pairs of vertices that had the same values for metrics used in the untying process. What these cases all had in common is the fact that, if two vertices appear in the exact same publications, there is no way to distinguish them by using only network properties – they cause a symmetry in the network. Hence, the matrix described in Section 3.2 was used as a final stopping criterion for the algorithm.

The previous example presented in Section 2.2 perfectly illustrates this problem. Since both vertices $v2$ and $v3$ appear in the same publications together ($p2$), we can

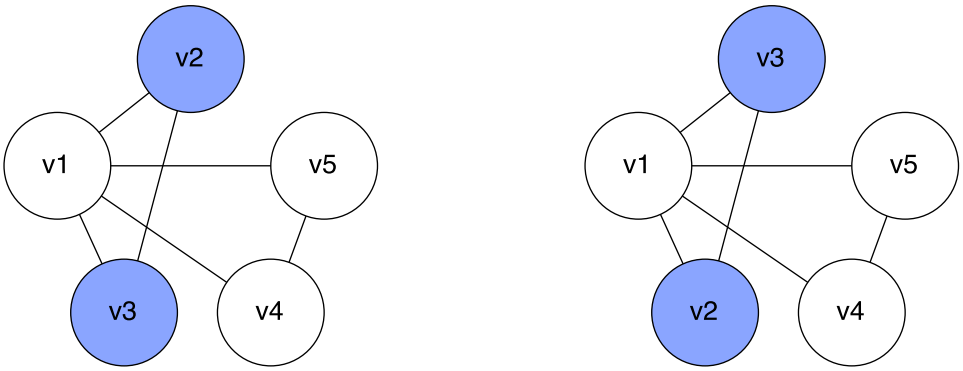


Figure 3.9: Vertex symmetry example.

not distinguish one from the other in terms of network properties (for instance, the degree and all three weights, $w1$,$w2$ and $w3$, have the same values for both vertices). Thus, we needed to store the list of vertices that had always appeared together in publications in order to provide the algorithm with a reliable stop sign.

Table 3.4: Storage of vertex symmetry information.

	v1	**v2**	**v3**	**v4**	**v5**
v1	0	0	0	0	0
v2	0	0	1	0	0
v3	0	1	0	0	0
v4	0	0	0	0	0
v5	0	0	0	0	0

We can see from Table 3.4 that it is indeed a symmetric matrix – the value zero means that there is no symmetry between vertices $[i, j]$, and the value one indicate that there is symmetry between $[i, j]$. As matrices grow quadratically in memory, a memory-light representation (a list of adjacencies) was used in this work with the purpose to store this information.

Chapter 4

Results

4.1 Analysis Strategy

In order to analyze the algorithm's performance, it is necessary to compare the vertex cover of each network (for each profile) with the actual set of names used by the respective author. Since this set was not known beforehand for each author, we had to generate these names heuristically.

A specific feature of the Google Scholar dataset was used in this process: given the author's full name, there is a finite set of possible names that Google uses to represent that specific author. And the elements of this set can be constructed using some simple rules, such as (taking as example the full name "Daniel Ratton Figueiredo"): there must be at least one unabbreviated surname in the final name ("Daniel RF" will not appear in the database); a first name and a surname can not be abbreviated if at least one previous name is not abbreviated ("D Ratton F", another example of impossibility); a first name or a surname can be discarded, if this does not break any previous rules. For instance, the list of all "possible" final names for "Daniel Ratton Figueiredo": "D Ratton Figueiredo", "D Ratton", "DR Figueiredo", "D Figueiredo" and "R Figueiredo".

Using these sets for each profile, the evaluation proceeded in comparing each name found by the algorithm (vertex cover) with all names generated for the comparison, using the rules previously mentioned.

4.2 Name Matching Problem

Since the name matching problem is widely studied, and several proposals [4] have been suggested for solving this issue, we decided to follow a more simple approach that uses a string distance algorithm (Levenshtein Distance) for name comparisons. Given two strings $s1, s2$, we can define the Leveshtein Distance $Lev(s1, s2)$ as being the least number of changes in either $s1$ or $s2$ as to make them identical. In this work, nevertheless, we used a slightly modified version of the original metric: consider len to be the length of the longest string among $s1$ and $s2$. Then, we define $Lev'(s1, s2)$ as being $\frac{len - Lev(s1,s2)}{len}, len \neq 0$.

This decision was taken in order to normalize the obtained values in the range [0,1], and to give a high score for strings that are alike (achieving value 1 when both strings are the identical). As it stands, these values are also more easily understandable when evaluating the results of the algorithm.

Thus, for each name $n1$ found by the algorithm in a egonet, we applied $Lev'(n1, n2)$ for each name $n2$ generated for that specific author. Then, the best matching result is stored for each $n1$ – and it represents the matching value for each of these names.

4.3 Matching Results

Two approaches were taken in order to evaluate the final results: the first one, we look into all the names found by the algorithm individually and their matching values. The second, we analyze the fraction of names, for each cover, that presents a matching value equal or greater than a threshold T.

As an example of the second approach, let $m(v) = \{1.0, 0.95, 0.7, 0.5\}$ be the cover name matching of a given vertex v. In this case, the fraction of names that have a matching value equal to 1 ($T = 1$) is $\frac{1}{4}$; analogously, the fraction of names that have a matching value greater or equal to 0.9 ($T >= 0.9$) is $\frac{2}{4}$, and if the threshold is set to 0.6, the corresponding fraction is $\frac{3}{4}$.

The values presented in Table 4.1 show that the average matching score for all names (93%) as well as the average fraction of names (87%), for each cover, that have a perfect matching score ($T = 1$) are very promising results.

A possible cause for high values in standard deviations is the fact that, if a name is put in the cover but does not actually represent the author, usually it presents a very low matching value – this happens because, in most cases, names of truly different authors tend not to be similar.

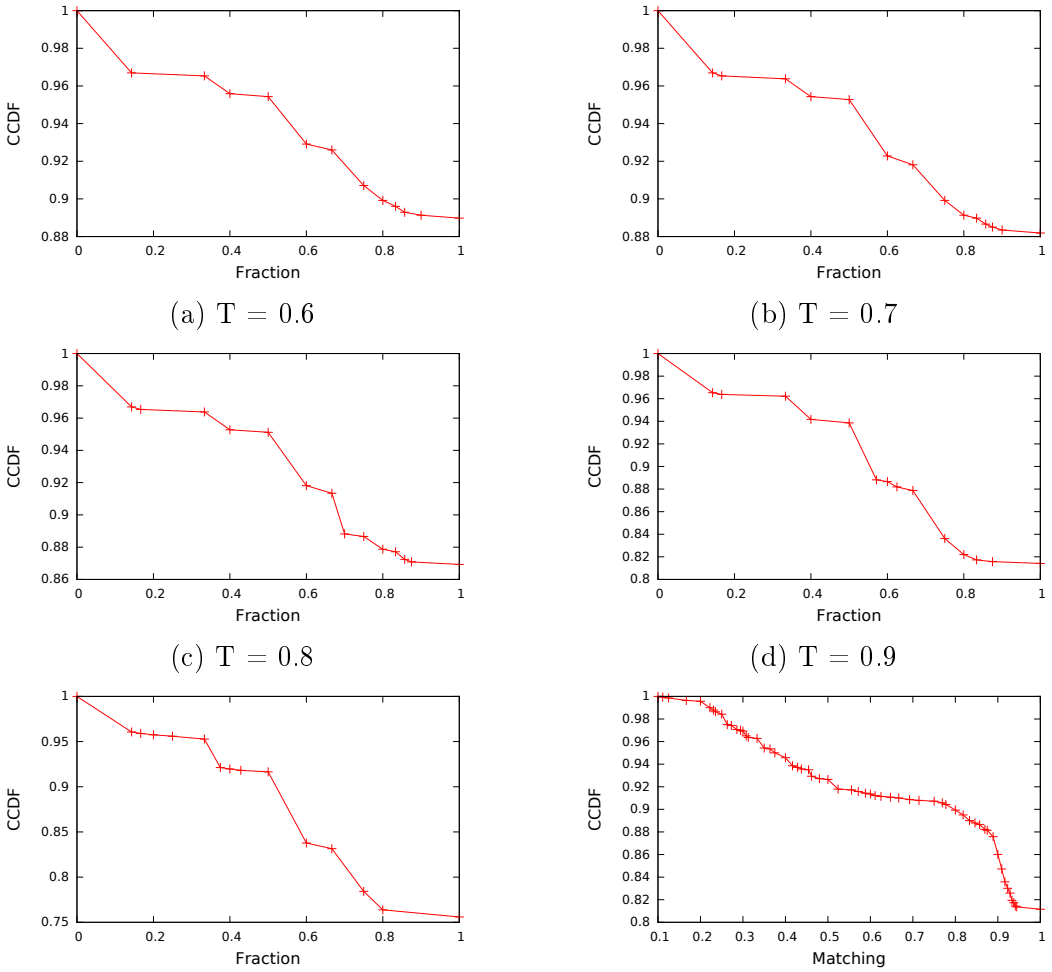
Table 4.1: Results statistics

	Mean	Stdev
Total	0.93	0.19
T = 1.0	0.87	0.26
T = 0.9	0.90	0.23
T = 0.8	0.92	0.22
T = 0.7	0.93	0.21
T = 0.6	0.94	0.21

An interesting result presented in Fig. 4.1e is that 75.5% of covers presented a fraction of 1 when the threshold was set to 1: this means that all names contained in these covers presented the best matching value of 1 to a name generated as shown in Section 4.1 – which suggests a satisfactory performance of the designed algorithm.

Note, as well, that when the threshold is set to a less strict matching result (such as 0.6) we can see that the percentage of covers that have a fraction of 1 goes up to 88.9% (Fig. 4.1a). Though a matching value of 0.6 may be considered low in terms of string matching (for instance, the strings “gap” and “tap” have a matching value of $\frac{2}{3}$, but they represent very different ideas), a higher threshold can be used for evaluating the algorithm.

In Fig. 4.2 we see the distribution of cover sizes. It has a mean value of 2.2 and a standard deviation of 1.2, which indicates that most ambiguities detected by the algorithm presents few different names (91.4% of covers have three or less vertices)



(a) T = 0.6 (b) T = 0.7

(c) T = 0.8 (d) T = 0.9

(e) T = 1.0 (f) Total

Figure 4.1: Graphical results obtained.

– even though the greatest cover presented 10 names in its composition. This result can indicate a certain author name usage pattern in Google Scholar database, since most covers showed exactly 2 names as shown in Table 4.2.

Table 4.2: Most frequent cover sizes.

Size	**% of total**
1	22.7%
2	51.4%
3	17.1%
4	4.7%
5 or more	4.1%

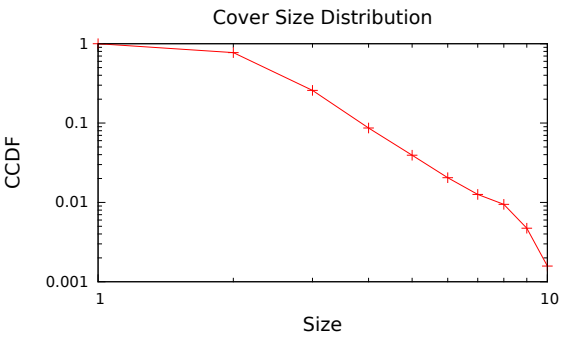


Figure 4.2: CCDF of presented cover sizes.

Chapter 5

Conclusion

Name ambiguity problem has been proven to be crucial in the context of scientific collaboration networks analysis. Several methods have been proposed in the literature in order to address this problem, ranging from manual inspection to machine learning tools and network feature based algorithms.

The presented work addressed this problem by designing and evaluating an algorithm, based on vertex cover, using networks that were generated from real world data – gathered from Google Scholar database.

The results obtained by this work were considered to be satisfactory, and show that the proposed algorithm can be useful in problems relating to correctly identifying author's names among bibliographical data – it also presented a high success rate in solving these types of name ambiguity.

This problem can also arise in several other scientific collaboration databases, such as the Lattes database for instance – increasing the relevance and possible applicability of the proposed solution.

Though the problem of ambiguity when multiple entities have the same identifier was not subject of this work, the problem focused here is still a major concern in bibliographical research. It is particularly relevant in the Brazilian context, where people usually have several surnames.

5.1 Future Work

Subsequent investigation of the proposed algorithm performance under a different database – such as Lattes – is being taken into consideration. Also, studying this algorithm in different types of networks (not only those induced by scientific collaboration) in another milestone for the presented work, as a way of broadening its applicability into other fields inside Network Science.

Bibliography

[1] 2016. "High-productivity software for complex networks". Disponível em: <https://networkx.github.io/>.

[2] AMANCIO, D. R., OLIVEIRA JR, O. N., DA F. COSTA, L., 2015, "Topological-collaborative approach for disambiguating authors' names in collaborative networks", *Scientometrics*, v. 102, n. 1, pp. 465–485. ISSN: 1588-2861.

[3] BORGATTI, S. P., MEHRA, A., BRASS, D. J., et al., 2009, "Network analysis in the social sciences", *science*, v. 323, n. 5916, pp. 892–895.

[4] CHRISTEN, P., 2006, "A comparison of personal name matching: Techniques and practical issues". In: *Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06)*, pp. 290–294. IEEE.

[5] ELLIOT, S., 2010, "Survey of author name disambiguation: 2004 to 2010", *Library Philosophy and Practice*, v. 473.

[6] ELLIOT, S., 2010, "Survey of author name disambiguation: 2004 to 2010", .

[7] ERDDS, P., R&WI, A., 1959, "On random graphs I", *Publ. Math. Debrecen*, v. 6, pp. 290–297.

[8] FERREIRA, A. A., GONÇALVES, M. A., LAENDER, A. H., 2012, "A brief survey of automatic methods for author name disambiguation", *Acm Sigmod Record*, v. 41, n. 2, pp. 15–26.

[9] GAREY, M. R., JOHNSON, D. S., 1990, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA, W. H. Freeman & Co. ISBN: 0716710455.

[10] GOMIDE, J., KLING, H., FIGUEIREDO, D., 2015, "A Model for Ambiguation and an Algorithm for Disambiguation in Social Networks". In: *Complex Networks VI*, Studies in Comp. Intelligence, Springer, pp. 37–44. ISBN: 978-3-319-16111-2.

[11] HUANG, J., ERTEKIN, S., GILES, C. L., 2006, "Fast author name disambiguation in CiteSeer", *ISI Tech. Report*.

[12] KARP, R. M., 1972, "Reducibility among combinatorial problems". In: *Complexity of computer computations*, Springer, pp. 85–103.

[13] KIM, J., DIESNER, J., 2015, "Distortive effects of initial-based name disambiguation on measurements of large-scale coauthorship networks", *J. of the Assoc. for Inform. Science and Tech.*

[14] LI, G.-C., LAI, R., DAMOUR, A., et al., 2014, "Disambiguation and co-authorship networks of the U.S. patent inventor database", *Research Policy*, v. 43, n. 6, pp. 941 – 955. ISSN: 0048-7333. doi: http://dx.doi.org/10.1016/j.respol.2014.01.012. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0048733314000225>.

[15] LIU, W., ISLAMAJ DOĞAN, R., KIM, S., et al., 2014, "Author name disambiguation for PubMed", *J. of the Assoc. for Inf. Science and Tech.*, v. 65, n. 4, pp. 765–81.

[16] LUCE, R. D., PERRY, A. D., 1949, "A method of matrix analysis of group structure", *Psychometrika*, v. 14, n. 2, pp. 95–116.

[17] NEWMAN, M., 2010, "Networks: an introduction". pp. 185–193, Oxford University Press.

[18] SHIN, D., KIM, T., CHOI, J., et al., 2014, "Author name disambiguation using a graph model with node splitting and merging based on bibliographic information", *Scientometrics*, v. 100, n. 1, pp. 15–50. ISSN: 1588-2861.

[19] TORVIK, V. I., SMALHEISER, N. R., 2009, "Author name disambiguation in MEDLINE", *ACM Transactions on Knowledge Discovery from Data*, v. 3, n. 3, pp. 11.

[20] WANG, J., BERZINS, K., HICKS, D., et al., 2012, "A boosted-trees method for name disambiguation", *Scientometrics*, v. 93, n. 2, pp. 391–411. ISSN: 1588-2861.