UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

Vitor Elisiário do Carmo

FastCELF++: Uma heurística de baixo custo computacional para maximização da influência em redes

São João del-Rei

2021

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

Vitor Elisiário do Carmo

FastCELF++: Uma heurística de baixo custo computacional para maximização da influência em redes

Dissertação apresentada como requisito para obtenção do título de mestre em Ciências no Curso de Mestrado do Programa de Pós Graduação em Ciência da Computação da UFSJ.

Orientador: Carolina Ribeiro Xavier

Coorientador: Vinícius da Fonseca Vieira

Universidade Federal de São João del-Rei – UFSJ

Mestrado em Ciência da Computação

São João del-Rei

2021

Vitor Elisiário do Carmo

FastCELF++: Uma heurística de baixo custo computacional para maximização da influência em redes

Dissertação apresentada como requisito para obtenção do título de mestre em Ciências no Curso de Mestrado do Programa de Pós Graduação em Ciência da Computação da UFSJ.

Trabalho aprovado. São João del-Rei , 6 de agosto de 2021:

Carolina Ribeiro Xavier
Orientadora

Vinícius da Fonseca Vieira
Coorientador

Fernanda Sumika Hojo de Souza
Avaliadora interna

Iago Augusto de Carvalho
Avaliador externo

São João del-Rei
2021

Dedico este trabalho aos meus pais Maria Aparecida e Daniel.
Ao meu tio Miguel (in memorian).

Agradecimentos

Agradeço primeiramente a Deus, por me proporcionar saúde, vida e determinação para superar os desafios que me foram apresentados.

Agradeço aos meus pais pelo apoio incondicional em todos os momentos, pelo incentivo, educação e exemplo de vida.

Aos meus grandes amigos Marcelo, Leonardo, Emanuele, Utã, Pedro que sempre estiveram do meu lado e torceram para o meu sucesso. E também ao meu afilhado Dominic.

Aos meus orientadores Carolina Xavier e Vinícius Vieira, não tenho palavras para agradecer todo apoio, disponibilidade e incentivo que recebi de vocês, sem contar a paciência que tiveram comigo. A conclusão deste trabalho só foi possível graças a vocês.

Aos professores da Universidade Federal de São João del-Rei e aos professores integrantes da excelente banca examinadora Fernanda Sumika e Iago Augusto pela disponibilidade em analisar e sugerir melhorias deste trabalho.

Por fim, aos colegas que fiz no mestrado e alguns da graduação que tive oportunidade de reencontrar nessa etapa.

Resumo

As redes sociais podem refletir os relacionamentos e interações entre indivíduos e têm desempenhado um papel muito importante no estudo da difusão de informação, em que a comunicação de ideias e compartilhamento de opiniões acontecem a todo momento. São diversos os exemplos de como as redes sociais podem afetar o comportamento dos indivíduos, tais como o *marketing* viral, a difusão de memes e a propagação de *fake news*. Essa dinâmica de difusão de informação tem motivado o estudo de diversas abordagens para identificar os principais influentes em uma rede. O Problema de Maximização de Influência consiste em encontrar um subconjunto S, chamado de conjunto de sementes, que seja capaz de propagar uma informação para o maior número possível de indivíduos na rede, tendo S como os influenciadores iniciais. Por ser um problema NP-difícil, é inviável encontrar o subconjunto S que garanta a difusão mais abrangente. Este trabalho apresenta uma modificação de dois algoritmos bem conhecidos para a seleção de conjunto de sementes, CELF e CELF++, substituindo as simulações de Monte Carlo por funções para calcular as estimativas de difusão (metamodelos). Os resultados mostram que: (1) os metamodelos podem ser usados para estimar qualitativamente a difusão de conjuntos de sementes; (2) o uso de métodos já conhecidos na literatura em conjunto com metamodelos é capaz de identificar, ordens de grandeza mais rápido, indivíduos mais influentes e, em alguns casos, até superar o resultado desses métodos em propagação.

Palavras-chaves: Maximização de influência, redes sociais, teoria de grafos, CELF, CELF++, otimização

Abstract

Social networks reflect the relationships and interactions between individuals and have played a very important role in the spread of information, in which the communication of ideas and dividing of opinions happen all the time. There are a variety of examples of how social networks can affect the behavior of individuals, like viral marketing, the spread of memes and the propagation of fake news. This dynamic of information diffusion has motivated the research of several approaches to identify the major influencers in a network. The Influence Maximization Problem consists of finding a subset S, called seed set, which is capable of propagating information to the largest possible number of individuals in the network, having S as the initial influencers. As it is an NP-hard problem, it is not feasible to find the S subset that guarantees the widest diffusion. This work introduces a modification of two well-known algorithms for selecting of a set of seeds, CELF and CELF++, replacing Monte Carlo simulations by functions to calculate the diffusion estimations (metamodels). The results show that: (1) metamodels can be used to qualitatively estimate the diffusion of seed sets; (2) the adoption of well-known methods in the literature together with metamodels is able to identify, orders of magnitude faster, more influential individuals and, in some cases, even outperform the result of these methods in propagation.

Key-words: Influence maximization, social networks, CELF, CELF++, graph theory, optimization

Lista de ilustrações

Figura 1 – *Betweenness* 23

Figura 2 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *CA-GrQc* 45
Figura 3 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *CA-HepTh* 46
Figura 4 – Correlações dos conjuntos de sementes nas redes *CA-GrQc* e *CA-HepTh* 47
Figura 5 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *fb-pages-company* 47
Figura 6 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *fb-pages-sport* 48
Figura 7 – Correlações dos conjuntos de sementes nas redes *fb-pages-company* e *fb-pages-sport* 48
Figura 8 – Tempo de execução dos algoritmos 50
Figura 9 – Difusão e similaridades de sementes na rede *CA-GrQc* 51
Figura 10 – Difusão e similaridades de sementes na rede *soc-hamsterster* 51
Figura 11 – Difusão nas redes *CA-HepTh* e *fb-pages-company* 52
Figura 12 – Difusão e similaridades de sementes na rede *soc-epinions* 52
Figura 13 – Difusão e similaridades de sementes na rede *slashdot0902* 53

Figura 14 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *netscience* 60
Figura 15 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *email* 60
Figura 16 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *soc-hamsterster* 61
Figura 17 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *CA-GrQc* 61
Figura 18 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *CA-HepTh* 61
Figura 19 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *fb-pages-company* 62
Figura 20 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *fb-pages-sport* 62
Figura 21 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *CA-AstroPh* 62
Figura 22 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *CA-CondMat* 63
Figura 23 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *soc-epinions* 63
Figura 24 – Comparativo γ_{EDV}, γ_{LIE} e σ na rede *slashdot0902* 63

Figura 25 – Difusão e similaridades de sementes na rede *netscience* 64
Figura 26 – Difusão e similaridades de sementes na rede *email* 65
Figura 27 – Difusão e similaridades de sementes na rede *soc-hamsterster* 65
Figura 28 – Difusão e similaridades de sementes na rede *CA-GrQc* 66
Figura 29 – Difusão e similaridades de sementes na rede *CA-HepTh* 66
Figura 30 – Difusão e similaridades de sementes na rede *fb-pages-company* 67
Figura 31 – Difusão e similaridades de sementes na rede *fb-pages-sport* 67
Figura 32 – Difusão e similaridades de sementes na rede *CA-AstroPh* 68
Figura 33 – Difusão e similaridades de sementes na rede *CA-CondMat* 68

Figura 34 – Difusão e similaridades de sementes na rede *soc-epinions* 69
Figura 35 – Difusão e similaridades de sementes na rede *slashdot0902* 69

Figura 36 – Difusão e similaridades de sementes na rede *netscience* 70
Figura 37 – Difusão e similaridades de sementes na rede *email* 71
Figura 38 – Difusão e similaridades de sementes na rede *soc-hamsterster* 71
Figura 39 – Difusão e similaridades de sementes na rede *CA-GrQc* 72
Figura 40 – Difusão e similaridades de sementes na rede *CA-HepTh* 72
Figura 41 – Difusão e similaridades de sementes na rede *fb-pages-company* 73
Figura 42 – Difusão e similaridades de sementes na rede *fb-pages-sport* 73
Figura 43 – Difusão e similaridades de sementes na rede *CA-AstroPh* 74
Figura 44 – Difusão e similaridades de sementes na rede *CA-CondMat* 74
Figura 45 – Difusão e similaridades de sementes na rede *soc-epinions* 75
Figura 46 – Difusão e similaridades de sementes na rede *slashdot0902* 75

Lista de tabelas

Tabela 1 – Redes utilizadas 38
Tabela 2 – Complexidade dos algoritmos 43
Tabela 3 – Complexidade dos algoritmos com os metamodelos 43

Lista de abreviaturas e siglas

CELF	Cost-Effective Lazy Forward
EDV	Expected Diffusion Value
GK	Algoritmo guloso proposto por Kemp
IC	Independent Cascade Model
IMP	Influence Maximization Problem, Problema de Maximização de Influência
LIE	Local Influence Estimation
LT	Linear Threshold Model
PCA	Principal Components Analysis
RBO	Rank-Biased Overlap
SA	Simulated Annealing

Sumário

1 Introdução 14
1.1 Justificativa 16
1.2 Objetivos 17
1.2.1 Geral 17
1.2.2 Objetivos específicos 17
1.3 Organização do Texto 17

2 Conceitos 18
2.1 Redes Complexas 18
2.1.1 Medidas de Centralidade 18
2.1.2 Centralidade de grau 19
2.1.3 Centralidade de autovetor 20
2.1.4 Centralidade de *closeness* 21
2.1.5 Centralidade de *betweenness* 21
2.1.6 Centralidade de PageRank 23
2.2 Redução de Dimensionalidade 24
2.2.1 PCA 24
2.3 Maximização de Influência 25
2.3.1 Propriedades da função de influência 25
2.4 Modelos de difusão 26
2.4.1 Independent Cascade 26
2.4.2 Linear Threshold 26
2.4.3 O cálculo da estimativa de propagação 27
2.5 Metamodelos 27
2.5.1 Expected Diffusion Value (EDV) 28
2.5.2 Local Influence Estimation (LIE) 28
2.6 Rank-Biased Overlap (RBO) 29

3 Trabalhos Relacionados 32
3.1 Métodos baseados em simulações de Monte Carlo 32
3.1.1 Algoritmo CELF 33
3.1.2 Algoritmo CELF++ 34
3.2 Métodos heurísticos 34

4 Metodologia 37
4.1 Conjunto de redes avaliadas 38

4.2 Execução dos experimentos . . . 39
4.3 Algoritmos propostos . . . 40
4.3.1 Cálculo das estimativas de difusão com os metamodelos . . . 40
4.3.2 FastCELF . . . 40
4.3.3 FASTCELF++ . . . 40
4.3.4 Análise de complexidade . . . 41

5 Resultados e discussões . . . 44
5.1 Configuração do ambiente de testes . . . 44
5.2 Validação dos metamodelos . . . 45
5.3 Eficiências das Heurísticas . . . 49
5.3.1 Tempo de execução . . . 49
5.3.2 Propagação de influência . . . 50

6 Considerações finais . . . 54
6.1 Trabalhos futuros . . . 54

Referências . . . 55

Apêndices 59

APÊNDICE A Comparativo das estimativas de difusão dos metamodelos e σ 60

APÊNDICE B Propagação de influência e correlação entre conjuntos de sementes - centralidades . . . 64

APÊNDICE C Propagação de influência e correlação entre conjuntos de sementes - heurísticas . . . 70

1 Introdução

As redes sociais podem refletir os relacionamentos e interações entre indivíduos e têm desempenhado um papel muito importante no estudo da difusão de informação, em que a comunicação de ideias e compartilhamento de opiniões acontecem a todo momento. São diversos os exemplos de como as redes sociais podem afetar o comportamento dos indivíduos, tais como o *marketing* viral (MA, 2008), a difusão de *memes* e a propagação de *fake news* (CAMPAN, 2017). Além disso, com um simples *tweet*, indivíduos influentes podem causar grande movimentação no mercado de ações e *criptomoedas* (BAMBROUGH, 2021), e quando tais indivíduos possuem cargo de destaque no meio político, seus comportamentos nas redes repercutem instantaneamente e podem afetar as relações entre países. As redes sociais são ferramentas extremamente poderosas, capazes de definir o rumo de eleições nacionais, já que a maioria da população tem acesso às diversas redes sociais existentes.

A dinâmica de difusão de informação em redes tem motivado o estudo de diversas abordagens para identificar os principais influenciadores em uma rede (capazes de propagar uma informação para o maior número possível de indivíduos), uma vez que a difusão de tais informações pode afetar toda a sociedade e trazer grandes consequências, tanto boas quanto ruins. Considerando o *marketing* viral, por exemplo, uma empresa paga para que pessoas divulguem seus produtos e serviços com o intuito de atingir um público que muitas vezes não conhece sua marca. A empresa quer obter o maior lucro possível a partir de uma campanha de publicidade e a escolha de quem vai fazer sua divulgação vai impactar diretamente no sucesso dessa campanha. Idealmente os selecionados para realizar a divulgação são pessoas altamente influentes. Por vezes a empresa pode distribuir amostras de seus produtos para uma parcela limitada de usuários, por questões econômicas. Alguns desses tendem a gostar do produto e vão influenciar seus vizinhos, dando início a um processo em cascata e, no fim das contas, uma grande fração dos usuários testará o produto, levando a uma melhoria significativa na receita obtida. Consequentemente esse processo só será bem-sucedido se as amostras forem distribuídas a usuários altamente influentes e, portanto, o problema aqui se resume em selecionar indivíduos influentes da rede.

Esse problema é conhecido como Problema de Maximização de Influência e possui aplicações em diversos domínios, além dos supracitados: recomendações personalizadas (SONG, 2006), classificação de *feeds* (IENCO, 2010), seleção de perfis influentes no *Twitter* (BAKSHY, 2011), direcionamento de anúncios em tempo real (LI, 2015).

O Problema de Maximização de Influência (denotado por IMP) consiste em en-

contrar um subconjunto S, chamado de conjunto de sementes, de no máximo k elementos, de modo que a propagação máxima (esperada) seja alcançada por meio de um modelo de difusão, tendo S como os influenciadores iniciais em uma rede. Existem dois modelos representativos popularmente adotados para estudar o processo de difusão: o *Independent Cascade Model*, que captura o comportamento independente de agentes na rede e o *Linear Threshold Model*, que captura o comportamento coletivo. Em ambos os modelos, a informação é difundida em intervalos de tempo discretos e o processo continua por várias rodadas.

Já foi demonstrado que o IMP é um problema de otimização NP-difícil (KEMPE, 2003), portanto, devido à sua complexidade, é inviável encontrar o subconjunto S que garanta a difusão mais abrangente. A abordagem mais comum para este problema, portanto, é a utilização de algoritmos aproximados, sendo a estratégia gulosa proposta por (KEMPE, 2003) (denotada por GK) capaz de alcançar uma aproximação de 63%. Porém esse método possui uma grande limitação na eficiência computacional, dado que a propagação de influência é estimada por simulações de Monte Carlo, que se repetem a cada iteração. Como resultado, encontrar um pequeno conjunto de sementes em uma rede moderadamente grande (20000 vértices, por exemplo), pode levar dias para concluir.

Em um dos mais notáveis trabalhos sobre o tema, (LESKOVEC, 2007) propõe o *Cost-Effective Lazy Forward* (CELF), cerca de 700 vezes mais rápido do que o GK, por explorar a propriedade de submodularidade da função de propagação para os modelos de difusão. Esse algoritmo seleciona um nó com o maior ganho marginal em cada iteração. O ganho marginal de um nó selecionado em cada iteração deve ser maior que os anteriores. Com isso, esse algoritmo consegue reduzir o número de chamadas na avaliação da propagação de influência.

(GOYAL, 2011) propõe o CELF++, que tenta melhorar a estratégia do CELF realizando o cálculo da propagação de influência para duas etapas do algoritmo de maneira simultânea, mas em tempo de execução excepcionalmente alto, embora nos experimentos dos autores ele tenha apresentado ganhos em tempo de execução entre 35 a 55% sobre o CELF.

Além das estratégias gulosas, é possível encontrar na literatura diversas abordagens que tentam encontrar boas soluções para o IMP através de métodos de otimização combinatória. Tais métodos são chamados de meta-heurísticas e muitas vezes fazem uso de uma *função objetivo* de minimização ou maximização para obter a melhor combinação possível de elementos pra solucionar um problema.

(JIANG, 2011) propõe a primeira abordagem baseada no *Simulated Annealing* para o IMP, com o diferencial de substituir as simulações de Monte Carlo por uma função objetivo bastante eficiente, chamada de EDV (abreviação para *Expected Diffusion Value*). O EDV obtém uma estimativa de difusão realizando um cálculo simples em que considera a

probabilidade de difusão e o número de vizinhos a um salto de distância dos nós avaliados. Seguindo a mesma linha, Gong et al. (GONG, 2016) utilizam um algoritmo de Enxame de Partículas Discreto combinado com uma função objetivo chamada de LIE (*Local Influence Estimation*), que estende o EDV para considerar também a vizinhança de dois saltos de distância e isso o torna mais sensível quanto à estimativa obtida. Visando simplificar a leitura, o termo "metamodelo" será adotado daqui em diante para referenciar alguma função de estimativa de difusão, como o EDV e o LIE.

Inspirado nos dois algoritmos supracitados, CELF e CELF++, este trabalho apresenta uma modificação dos mesmos, utilizando-se dos metamodelos EDV e LIE, normalmente utilizados como função objetivo em heurísticas de otimização, para substituir as simulações de Monte Carlo, visando tanto reduzir o custo computacional dos algoritmos clássicos como superá-los em propagação de influência. Buscou-se sintetizar e, com um foco especial, mostrar que: (1) os metamodelos podem ser usados para estimar qualitativamente a difusão de conjuntos de sementes; (2) o uso de métodos já conhecidos na literatura em conjunto com metamodelos é capaz de identificar indivíduos mais influentes em um tempo ordens de grandeza menor e, em alguns casos, até superar os resultados desses métodos em propagação.

1.1 Justificativa

O campo de estudo de processos de difusão em redes complexas ainda encontra-se em sua fase inicial, se comparado com outras áreas e com muitas questões a serem exploradas. A estrutura topológica da rede pode exercer um importante papel na dinâmica social pois ela pode afetar como a informação se propaga, porém ainda não está claro como essa estrutura afeta na dinâmica da difusão. Conhecer o funcionamento dessa estrutura é fundamental e o IMP é apenas um dos temas dentre vários de grande importância nos estudos de redes complexas, portanto isso pode conduzir a bons resultados para o problema apresentado neste trabalho, além de servir de inspiração para outras pessoas que possuam interesse nessa área.

Embora o IMP seja bastante relevante, são poucos os trabalhos encontrados na literatura que possuem uma abordagem parecida à que está sendo proposta neste trabalho. O desempenho dos modelos de difusão está diretamente relacionado à seleção de indivíduos influentes na rede, portanto a criação de métodos inteligentes ou o aperfeiçoamento das estratégias já existentes para a solução do IMP torna-se importante. Além disso, este trabalho fornece uma alternativa de baixo custo computacional e capaz de alcançar bons resultados.

1.2 Objetivos

1.2.1 Geral

Este trabalho tem como objetivo geral propor dois algoritmos inspirados no CELF e CELF++ para o problema de maximização de influência, nomeados de FastCELF e FastCELF++. A parte mais onerosa dos métodos originais são as simulações de Monte Carlo para calcular a estimativa de difusão sob um modelo. A substituição dessas simulações por funções que calculam uma estimativa de difusão reduz o tempo de execução desses algoritmos drasticamente. As funções para calcular a estimativa de difusão, aqui denominadas de metamodelos, foram o EDV e o LIE. Busca-se responder a seguinte pergunta: os metamodelos EDV e LIE são elegíveis para substituir o cálculo da estimativa propagação de influência nos algoritmos CELF e CELF++?

1.2.2 Objetivos específicos

O objetivo principal deste trabalho de subdivide da seguinte maneira:

- Selecionar um conjunto diversificado de redes, de tamanhos e contextos variados;
- Avaliar qualitativamente os metamodelos utilizando medidas de centralidade, afim de mostrar a viabilidade da substituição das simulações de Monte Carlo por metamodelos;
- Analisar a similaridade dos conjuntos de sementes obtidos através de diversos métodos. Uma vez que os conjuntos de sementes dos métodos propostos possuem alta correlação com os conjuntos dos algoritmos clássicos, pode-se considerar que as soluções obtidas são válidas;
- Analisar os resultados dos experimentos, confrontando os dados de desempenho em tempo de execução e eficiência dos algoritmos.

1.3 Organização do Texto

O restante deste trabalho está organizado da seguinte forma: o Capítulo 2 apresenta os conceitos teóricos, necessários para o bom entendimento da investigação proposta. O Capítulo 3 comenta sobre os trabalhos relacionados com o tema deste trabalho. O Capítulo 4 apresenta a metodologia a ser adotada. Os resultados atingidos e discussões sobre os mesmos se encontram no Capítulo 5 e por fim, no Capítulo 6 são discutidas as considerações finais.

2 Conceitos

2.1 Redes Complexas

O estudo de redes é um tema que abrange diversas áreas de conhecimento, tais como a biologia, sociologia, física, ciência da computação, matemática e várias outras áreas (SCHMITH, 2005; ANDERSON, 2012; FREEMAN, 1979; GIRVAN; NEWMAN, 2002; KNUTH, 1993). Esse campo tem se beneficiado enormemente de uma vasta gama de perspectivas trazidas por pesquisadores dessas diferentes disciplinas. Uma rede, também chamada de grafo, refere-se a uma estrutura composta por um conjunto de vértices (nós) que são interligados por meio de arestas. Esse tipo de modelagem pode ser usada para capturar as interações entre um grande número de indivíduos, além de permitir uma representação gráfica intuitiva.

Uma rede pode ser definida como $G = (V, E)$, sendo $V = \{v_1, v_2, v_3, ..., v_{\mathcal{N}}\}$ o conjunto de $\mathcal{N}$ vértices e $E = \{e_1, e_2, e_3, ..., e_{\mathcal{M}}\}$ o conjunto de $\mathcal{M}$ arestas. Computacionalmente, uma rede pode ser representada através de uma matriz de adjacência A de tamanho $\mathcal{N} \times \mathcal{N}$. Se existe uma aresta que conectam dois vértices i e j então a entrada $A_{ij} = 1$, caso contrário $A_{ij} = 0$. Caso a rede seja não direcionada $A_{ij} = A_{ji}$. Dois vértices são chamados vizinhos se existe uma aresta que os conectam.

2.1.1 Medidas de Centralidade

Medidas de centralidade em redes se referem a quão importante ou central um vértice é de acordo com uma propriedade. Tais medidas buscam identificar o quanto um vértice é importante em um determinado escopo. Por exemplo, pode-se considerar em um contexto específico que um vértice é importante examinando seu número de conexões (centralidade de grau), o número de caminhos mínimos que passam por ele (centralidade de *betweenness*) ou o quão próximo ele está dos demais vértices da rede (centralidade de *closeness*).

O conceito de centralidade é muito aplicado no contexto das redes sociais. As análises em geral focam no relacionamento interpessoal e na importância desempenhada por determinadas pessoas em uma rede, dada a função desenvolvida ou dado o número de conexões que possuem. Redes reais geralmente apresentam grande complexidade e esta complexidade faz com que alguns nós da rede e algumas de suas relações sejam mais importantes que as demais.

Várias medidas baseadas na estrutura da rede foram então propostas, buscando uma melhor caracterização das redes sociais. Utilizando-se uma determinada caracte-

rística, como por exemplo, o número de amigos que um indivíduo possui ou ainda a quantidade de pessoas importantes que uma pessoas conhece, é possível determinar a importância de um indivíduo para o grupo analisado. Além disso, surgiram diferentes ideias de como um nó ou uma ligação podem ser considerados importantes para uma rede, dando origem aos mais variados tipos de medidas. Tais medidas posteriormente ficaram conhecidas na literatura como medidas de centralidade e passaram a ser utilizadas em diferentes tipos de redes, por exemplo, redes de comunicação, biológicas, metabólicas, ou qualquer outro sistema que possa ser modelado como uma rede complexa.

Nas seções seguintes serão descritas as medidas de centralidade utilizadas neste trabalho e também quais características elas levam em conta e como calculá-las.

2.1.2 Centralidade de grau

A centralidade de grau ou *degree centrality* é talvez a mais simples, intuitiva e fácil de calcular entre todas as medidas de centralidade. Essa métrica avalia a importância de um nó analisando a quantidade de nós a que ele é ligado, ou seja, quanto maior o número de nós ligados a este, maior a importância do nó para a rede e, portanto, maior o valor atribuído para este vértice pela centralidade. O grau de um nó pode ser calculado utilizando-se a seguinte expressão:

$$k_i = \sum_{j=1}^{\mathcal{N}} A_{ij} \tag{2.1}$$

onde k_i é o grau do nó i, A_{ij} são os elementos da matriz de adjacências da rede complexa e $\mathcal{N}$ é o número de vértices na rede complexa. Em alguns casos (e também neste trabalho) pode-se utilizar uma variação da centralidade de grau, onde divide-se o valor de k_i pelo maior grau possível, garantindo-se assim que o valor da centralidade de cada nó esteja entre 0 e 1. Na Equação (2.2) é apresentada a fórmula para o cálculo da medida de centralidade de grau, quando normalizada pelo grau máximo.

$$k_i = \frac{\sum_{j=1}^{\mathcal{N}} A_{ij}}{\mathcal{N} - 1} \tag{2.2}$$

É importante ressaltar que na Equação (2.2) assume-se que a rede seja um grafo simples, ou seja, pode-se considerar que na rede não existam ligações de um nó com ele mesmo (*loops*) ou ligações paralelas (o que implicaria que o grau máximo de cada nó pudesse ser maior que $\mathcal{N} - 1$). Caso tais ligações existam, a expressão da Equação (2.2) continua válida, entretanto seus valores não estarão mais limitados entre 0 e 1. Outro fato que pode ser destacado é que para redes direcionadas deve-se levar em consideração ainda a existência de duas centralidades de grau, as de grau de entrada (*in degree*) e de saída (*out degree*), sendo que a primeira delas leva em conta quantas ligações chegam em um nó i e a última, quantas ligações o nó i possui para outros nós. Avaliar qual dessas duas métricas

deve ser levada em consideração para as redes direcionadas depende intrinsecamente do que a rede representa e quais características estão sendo analisadas. (NEWMAN, 2009) faz uma revisão sobre a centralidade de grau.

2.1.3 Centralidade de autovetor

Proposta por (BONACICH, 1987), a centralidade de autovetor ou *eigenvector centrality* estende o conceito da centralidade de grau da seguinte maneira: um nó é importante para a rede se ele possuir muitas ligações com outros nós, e/ou se tiver algumas conexões com nós que são altamente conectados na rede. Desta maneira, a centralidade de autovetor leva em consideração não apenas as conexões que o nó i possui como também quantas os seus vizinhos possuem. Pensando em uma rede social, a ideia seria que uma pessoa é importante se possui muitos amigos ou se conhece algumas pessoas com muitos contatos, ou ainda se estiver em um ponto intermediário entre as duas situações.

Imagine que no início todos os vértices possuam uma centralidade $x_i = 1$. Dessa maneira pode-se calcular as centralidades x'_i de todos os vértices como sendo o somatório das centralidades de todos os seus vizinhos ou seja:

$$x'_i = \sum_{j=1} A_{ij} x_j \tag{2.3}$$

onde A_{ij} são os elementos da matriz de adjacências representando a rede estudada. É possível notar ainda que a expressão (2.3) também pode ser escrita utilizando-se notação matricial da seguinte maneira $x' = Ax$, onde x é o vetor com os elementos x_i. Repetindo-se o processo da Equação (2.3) t vezes, obtém-se:

$$x(t) = A^t x(0) \tag{2.4}$$

onde $x(t)$ é o vetor com as centralidades para todos os nós após t iterações e $x(0)$ é o valor inicial atribuído a cada nó. Pode-se escrever $x(0)$ como uma combinação linear dos autovetores v_i da matriz de adjacências, de forma que:

$$x(0) = \sum_i c_i v_i \tag{2.5}$$

substituindo-se a Equação (2.5) na (2.4), obtém-se:

$$x(t) = A^t \sum_i c_i v_i = \sum_i c_i \kappa_i^t v_i = \kappa_1^t \sum_i c_i \left[\frac{\kappa_i}{\kappa_1}\right]^t v_i \tag{2.6}$$

onde os κ_i são os autovalores da matriz de adjacências A e κ_1 é o maior autovalor. Para um número de iterações grande o suficiente, os valores de $x(t)$ entrarão em um situação estacionária onde todos os valores de suas componentes não irão mais variar, logo utilizando-se a Equação (2.6) nota-se que no limite de $t \longrightarrow \infty$ obtém-se $x(t) \longrightarrow c_1 \kappa_1 v_1$.

Portanto, pode-se dizer que o valor da centralidade de autovetor no caso em que os valores da centralidade parem de variar pode ser escrita como:

$$Ax = \kappa_1 x \tag{2.7}$$

que é a centralidade de autovetor proposta por (BONACICH, 1987). Da Equação (2.7) pode-se notar que a centralidade do nó i depende de todos os seus vizinhos:

$$x_i = \kappa_1^{-1} \sum_j A_{ij} x_j \tag{2.8}$$

Demonstrando que a centralidade de autovetor leva em consideração o valor da centralidade de todos os vizinhos conectados com o nó i como discutido no início desta seção.

2.1.4 Centralidade de *closeness*

A centralidade de *closeness*, também conhecida pelo termo em português como centralidade de proximidade, foi definida por (FREEMAN, 1979) e é um exemplo de centralidade que utiliza informações sobre a distância entre os vértices. Ela mede o quanto cada vértice está perto dos demais e essa medida é dada pela distância geodésica de um vértice para todos os outros da rede. Uma revisão detalhada a respeito da centralidade de *closeness* pode ser encontrada na referência (NEWMAN, 2009). O objetivo dessa medida é avaliar o quanto um determinado nó está distante dos demais. Assim, os nós que possuírem uma menor distância média comparados com os demais, receberão um valor alto para a centralidade de *closeness*. Além disso, tais nós devem ser considerados importantes em uma rede complexa devido à sua influência, pois as informações presentes neles atingem os demais elementos da rede em um tempo menor do que os outros. Essa centralidade pode ser calculada da seguinte maneira:

$$C_i = \frac{1}{l_i}, \text{ onde } l_i = \frac{1}{\mathcal{N} - 1} \sum_{j(\neq i)} d_{ij} \tag{2.9}$$

sendo que n representa o número total de nós na rede e d_{ij} é o comprimento do menor caminho entre os nós i e j, logo l_i representa a média do comprimento das menores distâncias entre i e todos os outros nós da rede. É importante ressaltar que C_i é definido como o inverso dessa média para que a centralidade de *closeness* mantenha o mesmo padrão das outras medidas, onde os nós com valor maior para a centralidade (e por consequência a menor distância geodésica média) sejam os mais centrais. Essa medida também apresenta algumas complicações, quando a rede considerada possui mais de uma componente conexa.

2.1.5 Centralidade de *betweenness*

A centralidade de *betweenness* também pode ser encontrada em alguns trabalhos pelos termos em português como centralidade de intermediação ou ainda centralidade de

interposição. A ideia desta centralidade foi proposta em 1977 por (FREEMAN, 1977) e consiste em avaliar a importância de um nó na transmissão de mensagens ou eventos entre os demais nós, ou, de maneira equivalente, como ele encontra-se no caminho entre os outros nós da rede se estes quiserem trocar informações.

Imagine uma rede onde exista alguma informação sendo transmitida entre seus diversos nós. No caso de uma rede social, pode-se imaginar esta informação como sendo uma notícia, uma mensagem ou um rumor sendo espalhado entre os vários indivíduos da rede. No caso de um grafo representando a internet essas informações poderiam ser os pacotes de dados transmitidos entre os computadores e roteadores da rede. Assume-se que cada mensagem sempre escolha percorrer um dos menores caminhos entre o nó que emite a informação e o nó que a receberá, sendo que a probabilidade de escolha entre eles é igual. Tal suposição não é totalmente correta, pois sabe-se que raramente apenas os menores caminhos são escolhidos, uma vez que, se assim fosse, haveria um congestionamento na transmissão de informações. Todos os menores caminhos seriam utilizados, enquanto que caminhos um pouco maiores estariam livres. Utiliza-se essa suposição porque desta forma, a medida avalia os nós que possuem importância por diminuir a distância média entre os nós da rede, que são aqueles comuns a comunidades distintas. Supondo que cada par de nós troca mensagens com a mesma probabilidade por unidade de tempo, portanto, após uma quantidade de tempo razoável, para que um volume de mensagens seja transmitido, a quantidade média de informações que passa por um determinado nó será proporcional ao número de menores caminhos em que ele participa. Assim, pode-se definir a centralidade de *betweenness* para vértices como:

$$x_i = \sum_{st} \frac{n_{st}^i}{g_{st}} \tag{2.10}$$

onde n_{st}^i é o número de menores caminhos entre os vértices s e t que passa pelo vértice i e g_{st} é o número total de menores caminhos entre os vértices s e t. Caso a rede seja composta por mais de uma componente, o somatório considera apenas os nós pertencentes a mesma componente do nó i, pois na maioria dos casos, a comparação de centralidade de *betweenness* entre nós de componentes diferentes não tem sentido, visto que não existem caminhos entre nós de componentes distintas.

Na Figura 1 pode-se notar que existem dois subgrafos completos, o primeiro formado pelos nós $\{A, B, C, D\}$ e o segundo formado por $\{F, G, H, I\}$, que foram conectados pelo nó E. Esses dois subgrafos podem ser imaginados como comunidades em uma rede real e todas as trocas de mensagens entre dois nós presentes em um dos dois conjuntos distintos necessariamente passam pelos nós A, E e F, logo, é de se esperar que estes tenham um valor alto para a centralidade de *betweenness* e, portanto, são essenciais para a rede. Observa-se que a remoção de qualquer um dos nós citados e suas ligações faria com que a rede passasse a ter duas componentes. Outro fato interessante que se destaca é a

existência de apenas um caminho ligando os dois conjuntos de nós. Assim, caso exista um grande fluxo de informações entre todos os nós da rede, é de se esperar que este esteja congestionado.

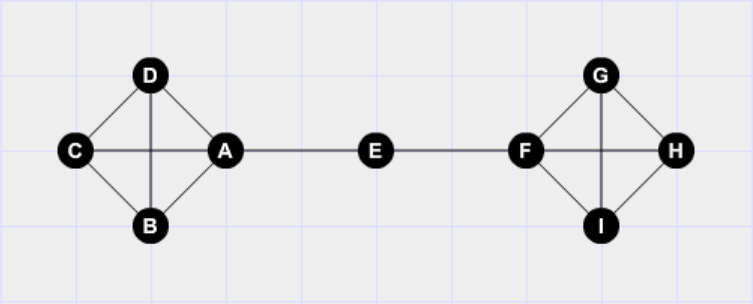

Figura 1 – Pode-se notar que o nó E tem papel fundamental para a transmissão de informação entre nós distintos em uma das duas comunidades presentes (representadas pelos dois grafos completos), a centralidade de *betweenness* tem um papel importante evidenciando este tipo de nó.

Os nós que possuem um alto valor para a centralidade de *betweenness* são essenciais pois, além de terem acesso a grande quantidade de informação transmitida, também fazem a comunicação entre elementos da rede em subconjuntos de vértices distintos. Se por algum motivo eles forem removidos ou danificados, a troca de informações na rede será prejudicada, uma vez que a remoção de nós com alto valor de *betweenness* implica no aumento da distância média entre os nós da rede complexa.

2.1.6 Centralidade de PageRank

Outra medida de centralidade baseada nos autovetores é a centralidade de PageRank, que herdou este nome do algoritmo PageRank (PAGE, 1999), foi proposto inicialmente para explorar a estrutura de *hyperlinks* da *web*, e assim, medir a relevância das páginas. Essa é a tecnologia que sustenta o motor de busca do Google, que gera listas de páginas úteis a partir de um índice de páginas que correspondam com a pesquisa feita pelo usuário. (LANGVILLE; MEYER, 2006) faz um estudo completo sobre o PageRank, mostrando conceitos gerais de máquinas de busca, *crawling*, recuperação de informação, além dos conceitos matemáticos envolvidos no cálculo dessa medida.

A centralidade de PageRank tem como princípio propagar aos vizinhos de um vértice um valor de centralidade proporcional ao número de arestas de saída. Desta forma, vértices que possuem grau de saída elevado propagam apenas uma pequena parte de centralidade adiante aos outros, mesmo que a centralidade seja alta. O cálculo do PageRank é apresentado na Equação 2.11:

$$R_p(i) = \alpha \sum_j \frac{A_{j,i}}{d_j} R_p(j) + \rho \tag{2.11}$$

onde α e ρ são constantes e d_j é o grau de saída do vértice v_j se o grau de for maior que zero, e $d_j = 1$, se o grau de saída for nulo.

2.2 Redução de Dimensionalidade

A redução de dimensionalidade consiste em transformar os dados de um espaço de alta dimensão em um espaço de baixa dimensão de forma que essa representação de baixa dimensão retenha propriedades significativas dos dados originais.

2.2.1 PCA

A Análise de Componentes Principais (Principal Components Analysis, PCA) é dos métodos mais conhecidos para redução de dimensionalidade. O PCA funciona identificando conjuntos de variáveis não correlacionadas entre si que explicam a maior parte da variabilidade dos dados. Em termos algébricos, busca-se por matrizes de menor *rank* que permitam explicar os dados originais e reconstruí-los de forma mais aproximada possível.

O PCA possui várias características atraentes (TAN, 2018). Em primeiro lugar, ele tende a identificar os padrões mais fortes nos dados. Segundo, muitas vezes a maior parte da variabilidade dos dados pode ser capturada por uma pequena fração do conjunto total de dimensões. Como resultado, a redução de dimensionalidade usando o PCA pode resultar em dados de dimensões relativamente baixas e pode ser possível aplicar técnicas que não funcionam bem com dados de alta dimensão. Por fim, uma vez que o ruído nos dados é mais fraco do que os padrões, a redução da dimensionalidade pode eliminar grande parte do ruído. Isso é benéfico tanto para mineração de dados quanto para outros algoritmos de análise de dados.

A ideia do PCA é encontrar um novo conjunto de dimensões (atributos) que melhor capture a variabilidade dos dados. Mais especificamente, a primeira dimensão é escolhida para capturar o máximo possível da variabilidade. A segunda dimensão é ortogonal à primeira e, sujeita a essa restrição, captura o máximo possível da variabilidade restante, e assim sucessivamente.

As medidas de centralidades apresentadas na Seção 2.1 atribuem um valor de importância a cada vértice. A ideia de utilização do PCA neste trabalho é aplicá-lo sobre esses valores para que se obtenha um novo *ranking*. Os k elementos desse *ranking* serão então utilizados como sementes para que os resultados sejam confrontados tanto com as sementes extraídas das medidas de centralidade quanto das heurísticas. O objetivo é avaliar a qualidade desse conjunto de sementes produzidas com o PCA.

2.3 Maximização de Influência

O Problema de Maximização de Influência (IMP) é primeiramente modelado como um problema de otimização por (KEMPE, 2003) e consiste em encontrar um subconjunto S, chamado de conjunto de sementes, de no máximo k elementos, de modo que a propagação máxima (esperada) seja alcançada por meio de um modelo de difusão, tendo S como os influenciadores iniciais em uma rede. As definições para a função de influência e o IMP, respectivamente, são as seguintes:

Definição 1 (Função de influência)**.** *Dado um grafo $G = (V, E)$ e seja m um modelo de difusão que captura o processo estocástico de difusão de informação em G. A função de influência $\sigma_m(S)$ é uma função tal que $\sigma_m(S) : 2^V \rightarrow \mathbb{R}$, em que dado $S \subseteq V$ o conjunto de influenciadores iniciais, $\sigma_m(S)$ denota o número esperado de vértices ativos no final do processo de ativação partindo de S.*

Definição 2 (Maximização de Influência)**.** *Dados um grafo $G = (V, E)$, um inteiro $1 \leq k \leq |V|$ e um modelo de difusão m, o problema de maximização de influência é definido como a seleção de um conjunto $S^* \subseteq V$ de tamanho k, tal que $\sigma_m(S^*) = max(\{\sigma_m(S)\ tal\ que\ |S| = k,\ S \subseteq V\})$*

O interesse na propagação de influência tem aumentado exponencialmente nos últimos anos, com aplicações que vão desde o *marketing* viral (MA, 2008), a difusão de *memes* e a propagação de *fake news* (CAMPAN, 2017). Assim, um aspecto importante para a compreensão da dinâmica de influência é a identificação de usuários que podem difundir informações para a maior parte possível da rede.

O *framework* genérico para um modelo de difusão é apresentado a seguir. Primeiramente associa-se a cada vértice $u \in V$ um status *inativo* ou *ativo*. Então, com base no grafo G, considera-se o seguinte processo de difusão entre os vértices. Inicialmente, considera-se o status de um conjunto de vértices escolhidos, chamado conjunto de sementes $S \subseteq V$, como ativo, enquanto os demais vértices em V estão inativos. Então, considera-se que os vértices em S podem influenciar seus vizinhos tornando-os ativos. Os vértices recentemente ativados podem posteriormente ativar seus vizinhos, e assim por diante. O processo de difusão, que se dá em passos discretos, termina quando nenhum novo vértice pode ser ativado.

2.3.1 Propriedades da função de influência

Seja U um conjunto e seja f uma função arbitrária definida como $f : 2^U \rightarrow \mathbb{R}^+$. Se o ganho marginal é não decrescente, diz-se que f é monótona (Definição 3) (KEMPE, 2003). Adicionalmente, f é *submodular* se o ganho obtido ao adicionar um determinado

elemento a um conjunto S é pelo menos tão alto quanto adicionar o mesmo elemento a um superconjunto de S (Definição 4).

Definição 3 (Monotonicidade)**.** *Sejam S, T e U conjuntos tais que $S \subseteq T \subseteq U$ e $f : 2^U \to \mathbb{R}^+$. f é monótona (não decrescente) se $f(S) \leq f(T)$.*

Definição 4 (Submodularidade)**.** *Sejam S, T e U conjuntos tais que $S \subseteq T \subseteq U$ e $f : 2^U \to \mathbb{R}^+$. f é submodular se $f(S \cup \{u\}) - f(S) \geq f(T \cup \{u\}) - f(T)$ para todo $u \in \setminus T$.*

(KEMPE, 2003) provam que a função de influência σ é monótona e submodular para os modelo IC e LT. Com relação à função σ de propagação esperada, seja S o conjunto de influenciadores iniciais e $\sigma(S \cup \{u\}) - \sigma(S)$ o *ganho marginal* de adicionar um novo vértice u ao conjunto S. A monotonicidade diz que à medida em que mais vizinhos de algum vértice v se tornam ativos, a probabilidade de v ser ativado aumenta. Enquanto que a submodularidade diz que o ganho marginal de um novo vértice diminui à medida em que o conjunto S cresce (GOYAL, 2011).

2.4 Modelos de difusão

Existem na literatura diversos modelos teóricos de difusão. Dois desses modelos representativos (KEMPE, 2003) são popularmente adotados para estudar o processo de difusão: o *Independent Cascade*, que captura o comportamento independente de agentes na rede e o *Linear Threshold*, que captura o comportamento coletivo.

2.4.1 Independent Cascade

O *Independent Cascade* (IC) (KEMPE, 2003) explora a independência dos vértices da rede no momento da difusão de influência. Neste modelo, a difusão da influência começa com um conjunto de nós ativos, sendo que a cada iteração do modelo, cada nó ativo tenta ativar independentemente seus nós vizinhos ainda não-ativados, com uma certa probabilidade de sucesso. Cada nó ativado só tem uma chance de tentativa de ativação sobre cada nó que é seu vizinho. O processo de difusão termina quando todos os nós ativados da rede realizaram suas tentativas de influenciar seus vizinhos e não há mais ativações possíveis.

2.4.2 Linear Threshold

O *Linear Threshold* (LT) (KEMPE, 2003) considera que cada nó de uma rede apresenta um limiar para adoção de uma ação, ideia ou comportamento, dependendo do número de nós que previamente adotaram o mesmo comportamento.

O limiar de decisão é um conceito que modela ações coletivas, em especial as ações onde cada indivíduo precisa tomar decisões entre duas opções de escolha, as quais são distintas e mutuamente exclusivas. O limiar pode ser interpretado como o ponto onde o indivíduo decide tomar sua decisão de aderir a uma escolha. Segundo o modelo, esse limiar é atingido quando uma determinada proporção de outros indivíduos já aderiu àquela escolha, o que influenciaria na decisão a ser tomada pelo indivíduo no momento de sua escolha.

De modo geral, a cada iteração do modelo, cada nó da rede será ativado quando o somatório das influências exercidas pelos seus nós vizinhos já ativos for igual ou maior do que seu limiar. O processo de difusão é aplicado de forma iterativa, onde cada vértice ativado continua ativo, até que nenhum nó da rede possa ser ativado.

2.4.3 O cálculo da estimativa de propagação

Segundo (CHEN, 2009; CHEN, 2010) é computacionalmente difícil calcular o valor exato de σ para o modelo IC, tornando-se necessário, portanto, fazer uma estimativa desse valor. Uma forma de calcular essa estimativa é obtendo a média do número de vértices ativados em diversas execuções do IC. Seja $\mathcal{A}$ o conjunto de vértices ativos ao final de uma execução do IC. As ativações de vértices no IC dependem de probabilidades, portanto, cada vez que o IC é executado, ele pode retornar um número de ativações diferente. Logo $|\mathcal{A}|$ é uma variável aleatória e seu valor esperado é $\sigma_{IC}(S)$. Sabendo disso, uma forma de alcançar maior precisão sobre $\sigma_{IC}(S)$ é calculando a média de todas as execuções. Esse tipo de estimativa é conhecido na literatura como Método de Monte Carlo e é definido como uma ferramenta para estimar valores através de amostragem e simulação (MITZENMACHER; UPFAL, 2017). De forma mais genérica, a estimativa de difusão de um conjunto S em um modelo de difusão m é obtida pela equação:

$$\sigma_m(S) = \frac{1}{\mathcal{R}} \sum_{i}^{\mathcal{R}} |\mathcal{A}_i| \tag{2.12}$$

onde r é o número de execuções do modelo m e $\mathcal{A}_i$ é o conjunto de nós ativados no processo de difusão na *i-ésima* execução. Tradicionalmente é necessário realizar um número grande de simulações para se obter uma boa acurácia, normalmente o valor de $\mathcal{R} = 10000$ (LESKOVEC, 2007; CHEN, 2009; GOYAL, 2011).

2.5 Metamodelos

Os metamodelos, ou modelos substitutos, são gerados a partir do ajuste de uma curva ou superfície a um conjunto inicial de pontos de amostragem pertencentes ao domínio de busca do problema, determinado a partir dos valores limites de cada variáveis de

projeto. Esses modelos aproximados tentam apresentar uma solução barata para um problema caro (HOYLE, 2006). Em algumas meta-heurísticas, por exemplo, pode-se considerar as funções objetivo como um metamodelo, que podem fornecer um resultado próximo de uma solução original do problema em questão a um custo computacional reduzido.

Trazendo para o contexto proposto neste trabalho, um metamodelo é uma função capaz de calcular de forma rápida a estimativa de difusão de um conjunto $S \subseteq V$ em um grafo $G = (V, E)$.

Nesta seção são apresentadas duas funções para calcular a estimativa de difusão, que a partir daqui serão chamados de metamodelos: o *Expected Diffusion Value* (EDV) e o *Local Influence Estimation* (LIE).

2.5.1 Expected Diffusion Value (EDV)

O Expected Diffusion Value (EDV) é um método de aproximação local proposto por (JIANG, 2011) para a propagação de influência no modelo IC, e busca uma aproximação da difusão calculada pelas simulações de Monte Carlo. Além disso, no mesmo trabalho o EDV é usado como função objetivo em um método baseado em *Simulated Annealing* (SAEDV) adaptado para o IMP e se mostrou capaz obter bons resultados com milhões de iterações. Seja G um grafo $G = (V, E)$ e S um conjunto $S \subseteq V$ de tamanho k. O EDV é calculado pela fórmula:

$$EDV(S) = k + \sum_{i \in N_S^{(1)} \setminus S} (1 - (1 - p)^{\tau(i)}) \tag{2.13}$$

onde $N_S^{(1)}$ representa os vizinhos diretos de S, $i \in N_S^{(1)} \setminus S$ representa os vértices vizinhos de i mas que não fazem parte de S, p é a probabilidade de ativação no modelo IC e $\tau(i)$ representa o número de arestas entre o vértice i e o conjunto inicial de vértices S. Quando o vértice i tem grau elevado e muitos vizinhos que não pertencem a S, sua contribuição no EDV é maior, pois a chance de i ativar algum novo vértice é maior e o termo $(1-(1-p)^{\tau(i)})$ da Equação 2.13 mostra isso. O EDV tem complexidade de tempo $\mathcal{O}(k\bar{D})$, sendo $\bar{D}$ o grau médio da rede.

2.5.2 Local Influence Estimation (LIE)

Local Influence Estimation (LIE) (GONG, 2016) também é um método de aproximação local para a propagação de influência sob o IC, mas diferente do EDV que considera apenas a vizinhança a um salto de distância do conjunto de vértices, este abrange também a vizinhança a dois saltos de distância. Assim como o EDV, o LIE foi concebido para ser usado como função objetivo em um método de otimização de enxame de partículas, que

conseguiu bons resultados com um tempo de execução aceitável sob o IC. Como função de estimativa o LIE pode ser formulado como:

$$LIE(S) = k + (1 + \frac{1}{|N_S^{(1)} \setminus S|} \sum_{u \in N_S^{(2)} \setminus S} p_u^* d_u^*) \sum_{i \in N_S^{(1)} \setminus S} (1 - \prod_{(i,j) \in E, j \in S} (1 - p_{i,j}))) \quad (2.14)$$

onde $N_S^{(1)}$ e $N_S^{(2)}$ representam a vizinhança de S a um e dois saltos de distância, respectivamente. p_u^* é a probabilidade de ativação de um nó u, que corresponde à probabilidade de propagação p no modelo IC ou à $\frac{1}{d_i^{(in)}}$ para o modelo *Weighted Cascade*. d_u^* representa o número de arestas entre as vizinhanças de salto 1 e 2 do vértice u. Por abranger uma vizinhança maior, a complexidade de tempo do LIE no pior caso é $\mathcal{O}(k\bar{D}^2)$, onde $\bar{D}$ é o grau médio da rede. (TANG, 2018) provou que o LIE tem bons resultados de aproximação como função objetivo.

2.6 Rank-Biased Overlap (RBO)

Imagine uma situação em que dois amigos, Alice e Bob, discutem quem são os cinco melhores jogadores de basquete de todos os tempos e então decidem, cada um, listar seus *rankings*, conforme é mostrado a seguir:

Lista da Alice:	**Lista do Bob:**
1. LeBron James	1. Michael Jordan
2. Michael Jordan	2. LeBron James
3. Kobe Bryant	3. Kareem Abdul-Jabbar
4. Magic Johnson	4. Larry Bird
5. Kareem Abdul-Jabbar	5. Magic Johnson

Então a pergunta é: quão semelhantes são os *rankings* listados por Bob e Alice? Por mais divertido que seja classificar jogadores de basquete, problemas semelhantes a esse ocorrem com frequência, como classificações de filmes ou classificações de produtos. Uma medida de similaridade pode ser bastante útil para ajudar a agrupar clientes, ou para avaliar uma lista de classificação gerada por um novo modelo em relação aos antigos.

Antes de prosseguir para as definições, algumas propriedades desejáveis para a medida de similaridade precisam ser enumeradas:

1. Deseja-se que tal medida seja limitada, por exemplo, entre $[0, 1]$, com 0 significando completamente diferente e 1 significando idêntico;

2. Espera-se que a medida de similaridade seja capaz de permitir itens mostrados em apenas uma das duas listas. No exemplo acima, Kobe Bryant está apenas na lista de Alice, enquanto Larry Bird está apenas na lista de Bob, mas a medida deve ser capaz de lidar com tais situações.

A primeira propriedade é importante para indicar o grau de similaridade entre as listas. No caso do IMP isso é essencial quando se quer comparar a similaridade entre os conjuntos de sementes produzidos por dois algoritmos X e Y. Se os conjuntos apresentam alta similaridade, espera-se que os valores de difusão desses conjuntos sejam semelhantes. Em outras palavras, se X seleciona boas sementes então Y também deve selecionar boas sementes. Considerando a segunda propriedade, não há garantias de que o algoritmo X seleciona os mesmos elementos que Y, ainda que em ordem diferente. Nessas situações, diz-se que as listas produzidas pelos algoritmos X e Y são indefinidas: alguns itens aparecem em uma lista mas não na outra. Portanto a segunda propriedade para a medida de similaridade é de extrema importância.

Existem vários métodos bem conhecidos para medir tanto a distância quanto a similaridade entre pares de *rankings* que compartilham o mesmo conjunto de itens, tais como os coeficientes de Spearman e Kendall (KENDALL, 1948; KRISHNAN, 2012). Porém, muitos problemas envolvem comparar classificações indefinidas e as métricas padrão não consideram tais casos. O *Rank-Biased Overlap* (RBO) é um coeficiente que calcula a similaridade de dois conjuntos ordenados e foi proposto por (WEBBER, 2010) para atuar com classificações ordinais e indefinidas.

Por se tratar de uma medida baseada em ordenação e intersecção, acredita-se que este método seja adequado para a coleta das impressões sobre a formação da lista de sementes de cada critério e de sua ordenação. A medida respeita um intervalo numérico entre 0 e 1 onde maiores valores implicam em maior nível de similaridade entre as listas. O princípio do RBO é que elementos com classificações diferentes têm pesos diferentes, dando pesos maiores para aqueles com classificações mais altas. O coeficiente RBO de duas listas S e T é calculado pela seguinte fórmula:

$$RBO(S,T,w) = (1-w)\sum_{d=1}^{\eta} w^{d-1} A(S,T,d) \tag{2.15}$$

em que η é o número de elementos na lista de classificação e o parâmetro w determina o quão acentuado é o declínio nos pesos: quanto menor w, mais ponderada é a métrica. No limite, quando $w = 0$, apenas o item com melhor classificação é considerado, e a pontuação RBO é 0 ou 1. Por outro lado, à medida que w se aproxima arbitrariamente de 1, os pesos tornam-se arbitrariamente planos e a avaliação torna-se arbitrariamente profunda. Na Equação 2.15, $A(S,T,d)$ é o valor de sobreposição entre duas listas de classificação S e T até a posição d no *ranking*, calculada pela fórmula a seguir:

$$A(S,T,d) = \frac{|S_{1:d} \cap T_{1:d}|}{|S_{1:d} \cup T_{1:d}|} \tag{2.16}$$

onde $S_{1:d}$ representa os elementos presentes na posição 1 até d da lista S, e $T_{1:d}$ indica os mesmos elementos da lista T.

3 Trabalhos Relacionados

O IMP possui diversas aplicações, como foi mencionado na Seção 2.3 e com o crescente interesse no estudo do IMP, vários trabalhos têm surgido ao logo dos anos e consequentemente diversas abordagens. Alguns desses trabalhos são comentados neste capítulo por servirem de fontes de inspiração ou por contribuírem de alguma forma com a metodologia proposta neste trabalho.

(KEMPE, 2003) propuseram uma solução gulosa com uma aproximação de 63% da solução ótima para o IMP. Esse método será denotado por *GK*. Os autores mostram ainda que os algoritmos de aproximação para o IMP podem ser desenvolvidos em um *framework* genérico baseado em funções submodulares (Subseção 2.3.1). Dessa forma, Além disso, os autores mostram que a solução proposta supera significativamente (em qualidade da solução) algumas heurísticas de seleção de nós baseadas em centralidade de grau. Ao investigar o motivo pelo qual as heurísticas baseadas em centralidade de grau não funcionam tão bem, percebe-se que ela ignoram a dinâmica das informações, em vez de depender apenas das propriedades estruturais da rede. Em particular, nenhuma das heurísticas incorpora o fato de que muitos dos nós mais centrais (ou de grau mais alto) podem estar agrupados, de modo que a seleção de todos eles é desnecessária. Na verdade, a natureza desigual desses resultados sugere que a rede de influência de muitos nós não é refletida com precisão por seu grau ou centralidade.

Ao longo dos anos, surgiram diversas abordagens para o IMP, buscando bons resultados da forma mais eficiente possível. Algumas dessas abordagens sugerem aprimoramentos no GK, atacando principalmente o problema de ineficiência computacional e preservando as propriedades de monotonicidade e submodularidade do método. Outro tipo de abordagem para o IMP é a utilização de métodos heurísticos, que buscam por resultados comparáveis ou até superiores aos métodos tradicionais. A principal vantagem da aplicação de tais métodos é, em geral, o tempo de execução consideravelmente reduzido. As próximas seções apresentam duas categorias de soluções.

3.1 Métodos baseados em simulações de Monte Carlo

Os métodos apresentados nesta seção têm em comum a utilização de simulações de Monte Carlo para estimar σ. Esses métodos buscam reduzir o número de chamadas no processo de estocástico, que normalmente é $\mathcal{R} = 10000$ repetições, para obter boa acurácia. Portanto, tais métodos correspondem a aprimoramentos no método GK. O cálculo do σ em cada iteração é feito conforme a Equação 2.12.

3.1.1 Algoritmo CELF

O *Cost-Effective Lazy Foward* (CELF) (LESKOVEC, 2007) explora a propriedade de submodularidade da função de difusão. A ideia é que o ganho marginal fornecido por um nó na iteração atual não pode ser melhor do que o ganho marginal fornecido por ele nas iterações anteriores (Definição 4). O Algoritmo 1 apresenta o pseudo-código do CELF:

Algorithm 1 CELF

```
1:  function CELF(G, k, σ_m)
2:      S ← ∅, Q ← ∅
3:      for each u ∈ V do
4:          u.mg ← σ_m({u})
5:          u.flag ← 0
6:          Q ← Q ∪ {u}
7:      end for
8:      heapify(Q)                                  ▷ baseado em u.mg
9:      while |S| < k do
10:         u ← Q[0]
11:         if u.flag = |S| then
12:             S ← S ∪ {u}
13:             Q ← Q − {u}
14:         else
15:             u.mg ← σ_m(S ∪ {u}) − σ_m(S)
16:             u.flag ← |S|
17:         end if
18:         heapify(Q)                              ▷ baseado em u.mg
19:     end while
20:     return S
21: end function
```

O algoritmo mantém uma lista de vértices ordenada de maneira decrescente pelo ganho marginal. Normalmente uma *heap* Q é empregada como fila de prioridade para manter a lista ordenada. σ_m denota a estimativa do número de vértices ativados a partir do conjunto de sementes S sob o modelo de propagação m (como IC ou LT) (Equação 2.12) e k corresponde ao número de sementes que deseja-se selecionar. Um elemento de Q corresponde a uma tupla $\langle u.mg, u.flag\rangle$, em que $u.mg = \sigma_m(S \cup \{u\}) - \sigma_m(S)$ representa o ganho marginal de u com relação ao conjunto S, enquanto $u.flag$ indica em qual iteração $u.mg$ foi atualizado pela última vez. No início do algoritmo o ganho marginal de cada vértice u é calculado e adicionado a Q (linhas $3-7$). Em seguida a *heap* Q é refeita para manter o vértice de maior ganho marginal na raiz da *heap*. Depois, em cada uma das k iterações é verificado se o ganho marginal de u foi calculado na iteração atual. Se sim, u é o vértice de maior ganho marginal na iteração atual e, portanto, será selecionado como o próximo vértice semente (linhas $10-13$). Caso contrário, é recalculado o ganho marginal de u, o atributo $u.flag$ é atualizado e a *heap* é refeita (linhas $14-18$). Devido à atualização de ganho marginal, o CELF consegue ser melhor em tempo de execução do

que o método GK em até 700 vezes, porque reduz o número de chamadas no processo que estima $\sigma_m(S)$.

3.1.2 Algoritmo CELF++

O CELF++ (GOYAL, 2011) possui estrutura semelhante ao CELF, mantendo uma *heap* Q para armazenar os vértices. Cada elemento de Q corresponde a uma tupla $\langle u.mg1, u.mg2, u.prev\_best, u.flag\rangle$. Aqui $u.mg1 = \Delta_u(S)$ corresponde ao ganho marginal de u em relação ao conjunto S; $u.prev\_best$ é o vértice que tem o máximo ganho marginal entre todos os vértices examinados na iteração atual, antes de u; $u.mg2 = \Delta_u(S \cup \{prev\_best\})$ é o ganho marginal de u em relação a $S \cup \{prev\_best\}$ e $u.flag$ é a iteração em que $u.mg1$ foi atualizado pela última vez. A ideia principal do CELF++ é que se o vértice selecionado na última iteração ainda é a raiz da *heap* Q, não é necessário recalcular os ganhos marginais. Com isso, é possível fazer menos chamadas à função σ e, consequentemente, reduzir o tempo de processamento. O algoritmo pode ser implementado de modo que $\Delta_u(S)$ e $\Delta_u(S \cup \{prev\_best\}$ sejam avaliados simultaneamente um uma única iteração do procedimento de simulação de Monte Carlo. Os resultados dos experimentos mostram uma melhoria de 17 a 61% em tempo de execução em relação ao CELF. O pseudo-código do CELF++ é apresentado no Algoritmo 2:

A variável *last_seed* corresponde ao último vértice selecionado para o conjunto de sementes S; *cur_best* corresponde ao vértice de maior ganho marginal entre todos os examinados naquela iteração; k é o número de sementes que deseja-se extrair. O aprimoramento proposto está nas linhas $19 - 20$, onde o ganho marginal $u.mg1$ é atualizado sem a necessidade de recalculá-lo, desde que $u.mg2$ tenha sido calculado com relação ao último vértice selecionado. Se nenhum dos casos são aplicados, então o ganho marginal é recalculado. Mesmo com esse ganho de desempenho, CELF e CELF++ não são escaláveis (CHEN, 2010).

3.2 Métodos heurísticos

Nesta seção serão apresentadas alguns métodos mais eficientes em tempo de execução do que os algoritmos da Seção 3.1 e que mantêm nível competitivo de propagação de influência.

O *DegreeDiscount* (CHEN, 2009) tenta conduzir uma análise mais profunda da estrutura local dos vértices para uma probabilidade de difusão específica p. Esse método foi projetado especialmente para o modelo IC. A ideia geral do algoritmo é que se um vértice for considerado como semente, então as arestas que se conectam a ele não serão contados como um grau dos outros vértices, ou seja, ao considerar o próximo nó, as arestas que se conectam com os nós já no conjunto de semente serão descontados.

Algorithm 2 CELF++

```
1: function CELF++(G, k, σ_m)
2:     S ← ∅, Q ← ∅
3:     last_seed ← NULL, cur_best ← NULL
4:     for each u ∈ V do
5:         u.mg1 ← σ_m({u})
6:         u.mg2 ← Δ_u(cur_best)
7:         u.prev_best ← cur_best
8:         u.flag ← 0
9:         Q ← Q ∪ {u}
10:    end for
11:    heapify(Q)                                   ▷ baseado em u.mg1
12:    while |S| < k do
13:        u ← Q[0]
14:        if u.flag = |S| then
15:            S ← S ∪ {u}
16:            last_seed ← u
17:            cur_best ← NULL
18:            Q ← Q − {u}
19:        else if u.prev_best = last_seed and u.flag = |S| − 1 then
20:            u.mg1 ← u.mg2
21:        else
22:            u.mg1 ← Δ_u(S)
23:            u.prev_best ← cur_best
24:            u.mg2 ← Δ_u(S ∪ {cur_best})
25:        end if
26:        u.flag = |S|
27:        update cur_best
28:        heapify(Q)                               ▷ baseado em u.mg1
29:    end while
30:    return S
31: end function
```

Em (JIANG, 2011) os autores propõem uma nova abordagem baseada na heurística *Simulated Annealing* (denotado como SA) para o IMP. Segundo os autores, este foi o primeiro algoritmo baseado em SA para o problema. Para melhorar ainda mais a eficiência do algoritmo básico, o trabalho propõe o Expected Diffusion Value (EDV, descrito em detalhes na Subseção 2.5.1), para substituir as simulações de difusão de um conjunto de nós de acordo com as propriedades das redes sociais. Adicionalmente, os autores propõem uma heurística de difusão de nó único (denotada como SH) para gerar melhores conjuntos de soluções e acelerar o processo de convergência do SA. Além disso, o EDV e SH são combinados para integrar seus valores em eficiência e precisão, respectivamente. Com isso os autores conseguiram mostrar que os algoritmos propostos baseados no SA foram capazes de superar os métodos descritos por (CHEN, 2009) e (WANG, 2010), tanto em termos de eficiência quanto de propagação de influência.

(BUCUR; IACCA, 2016) abordam o IMP por meio de um algoritmo genético. Os autores mostram que, usando operadores genéticos simples, é possível encontrar em soluções com tempos de execução viáveis de alta influência que sejam comparáveis, e em alguns casos até melhores, do que as soluções encontradas por uma série de heurísticas conhecidas (uma das quais foi previamente comprovada a melhor garantia de aproximação possível, em tempo polinomial, da solução ótima). As vantagens dos Algoritmos Genéticos mostram, no entanto, que elas não exigem nenhuma suposição sobre a rede, e nelas obtêm conjuntos de soluções mais viáveis do que as heurísticas atuais.

No estudo proposto em (CUI, 2018), os autores analisam as causas da baixa eficiência das abordagens gulosas e propõem um algoritmo mais eficiente, denominado evolução de busca grau-descendente (denotado como DDSE). Primeiro é proposta uma estratégia de busca decrescente (denotada como DDS). O DDS é capaz de gerar um conjunto de nós cuja difusão de influência é comparável à centralidade de grau. Com base no DDS, os autores desenvolveram um algoritmo evolutivo que foi capaz de melhorar significativamente a eficiência, eliminando as demoradas simulações dos algoritmos gulosos. Resultados experimentais em redes sociais do mundo real demonstraram que o DDSE é cerca de cinco ordens de magnitude mais rápido do que o método guloso proposto por (LESKOVEC, 2007), mantendo a precisão competitiva, o que pode verificar a alta eficácia e eficiência da abordagem proposta para maximização de influência.

4 Metodologia

Neste capítulo são apresentadas as modificações que eliminam por completo as simulações de Monte Carlo para estimar a difusão nos algoritmos CELF e CELF++ (algoritmos 1 e 2, respectivamente). As modificações foram feitas com base na metodologia apresentada nas próximas seções. Nos experimentos, os algoritmos resultantes superaram os métodos em que foram inspirados tanto em tempo de execução quanto em propagação de influência. A metodologia do presente trabalho pode ser descrita, sucintamente, nas seguintes etapas:

1. Obter os *rankings* das medidas de centralidade das redes e calcular a propagação de influência sob o modelo IC para os *rankings*. As centralidades escolhidas foram grau, autovetor, PageRank, *betweenness* e *closeness*;
2. Obter o *ranking* baseado no PCA dos conjuntos que alcançarem melhores resultados de propagação na etapa 1 e calcular sua propagação de influência;
3. Calcular as estimativas de difusão com os metamodelos EDV e LIE, para os conjuntos de sementes obtidos nas etapas anteriores;
4. Calcular as similaridades dos conjuntos com o RBO e analisar juntamente com os resultados de propagação de influência desses conjuntos;
5. Extrair os conjuntos de sementes com as heurísticas da literatura e os métodos propostos;
6. Calcular as similaridades e a propagação de influência dos novos conjuntos;
7. Avaliar os resultados obtidos, tanto em tempo de execução quanto em propagação de influência.

Todos os conjuntos extraídos dos *rankings* correspondem aos 50 elementos do top. Todos os cálculos da propagação de influência foram realizados variando o tamanho de cada conjunto, iniciando com os 5 primeiros elementos dos *rankings* até o máximo de 50, com incrementos de 5 em 5.

A utilização do PCA na etapa 2 da metodologia tem como objetivo obter um novo *ranking* que capture as diversas características estruturais das redes e avaliar a qualidade de difusão desse conjunto de sementes. Conforme foi comentado no Capítulo 3, a investigação realizada no trabalho de (KEMPE, 2003) mostrou que as heurísticas baseadas em centralidades não capturam o efeito da dinâmica de informação da rede.

Busca-se saber se uma agregação de medidas de centralidades através do PCA pode gerar conjuntos de sementes capazes de atingir bons resultados de difusão.

A nova metodologia consiste na união de duas abordagens encontradas na literatura, sendo a primeira metamodelos que visam estimar o cálculo de um modelo de influência, e o segundo um algoritmo (guloso) que utiliza de muitas chamadas do modelo de influência (IC).

4.1 Conjunto de redes avaliadas

Buscou-se utilizar um conjunto bastante variado de redes reais, distribuídas entre as categorias de redes sociais, de comunicação, co-autoria e redes de colaboração. A Tabela 1 descreve as características desse conjunto de dados. Cada rede passou por uma etapa de pré-processamento em que foi extraída a sua componente gigante. Portanto os valores apresentados na Tabela 1 são referentes à componente gigante de suas respectivas redes.

Tabela 1 – Características das redes utilizadas: número de nós (|V|); número de arestas (|E|); grau médio ($\hat{k}$); coeficiente de *clustering* médio ($\hat{c}$).

Rede	**Tipo**	**|V|**	**|E|**	$\hat{\mathbf{k}}$	$\hat{\mathbf{c}}$	**Categoria**
netscience†	não-direcionada	379	914	4.82	0.741	Co-autoria
email†	não-direcionada	1134	5451	9.62	0.220	Comunicação
soc-hamsterster*	não-direcionada	2000	16097	16.09	0.539	Social
CA-GrQc*	não-direcionada	4158	13422	6.45	0.556	Colaboração
CA-HepTh*	não-direcionada	8638	25998	5.74	0.481	Colaboração
fb-pages-company†	não-direcionada	14113	52310	7.38	0.239	Social
fb-pages-sport†	não-direcionada	13866	86858	12.52	0.276	Social
CA-AstroPh*	não-direcionada	17903	196972	22.00	0.632	Colaboração
CA-CondMat*	não-direcionada	21363	91286	8.54	0.641	Colaboração
soc-epinions*	direcionada	75877	508836	13.41	0.137	Social
soc-slashdot0902*	direcionada	82168	870161	21.18	0.060	Social

As redes assinaladas com $*$ foram coletadas da base de dados *SNAP*[1] (LESKOVEC; KREVL, 2014) e as redes assinaladas com † foram coletadas do *Network Repository*[2] (ROSSI; AHMED, 2015).

[1] http://snap.stanford.edu/data/index.html
[2] http://networkrepository.com/

4.2 Execução dos experimentos

Definidas as redes, a próxima etapa da metodologia consiste em realizar a construção dos *rankings* das redes para extração das sementes. Os *rankings* são obtidos através do cálculo das medidas de centralidade de cada rede, descritas na Seção 2.1. Os valores obtidos pelas medidas de centralidade são ordenados em ordem decrescente, fazendo com que os vértices mais bem avaliados pelas medidas de centralidade fiquem no topo do *ranking*. O modelo é testado com 5, 10, 15, 20, 25, 30, 35, 40, 45 e 50 elementos para cada conjunto de sementes.

Os *rankings* das medidas de maior sucesso serão usados no cálculo do PCA, afim de se obter novos conjuntos de vértices que serão as testados no modelo de propagação de influência IC, discutido na Seção 2.3. Adicionalmente, os conjuntos de sementes individuais serão avaliados pelo IC para que seja possível evidenciar o desempenho tanto das medidas de centralidade quanto do PCA, na seleção de sementes.

Em seguida, todos os pares conjuntos de sementes têm suas similaridades calculadas pelo RBO. Vale destacar que esse método foi escolhido porque ele é capaz de contornar o problema de pares de conjuntos indefinidos, ou seja, casos em que elementos de um conjunto mas não estão presentes no outro. A análise de similaridades é feita para que se verifique se os métodos propostos são capazes de selecionar conjuntos semelhantes aos métodos bem conhecidos na literatura. Além disso, acredita-se que, se as sementes selecionadas por um algoritmo A possuem alta correlação com as sementes de um algoritmo B, a propagação de influência de A e B tende a ser semelhante.

Para cada conjunto de sementes são calculados as estimativas de difusão através dos metamodelos EDV e LIE. O objetivo dessa etapa é confrontar o resultado da propagação de influência com as estimativas dos metamodelos e embasar a viabilidade de adotá-los como função de estimativa no lugar das simulações de Monte Carlo para seleção de sementes nos algoritmos CELF e CELF++.

Na próxima etapa, extrai-se sementes também através dos métodos heurísticos afim de avaliar eficiência em tempo e propagação desses. Os métodos heurísticos utilizados são DegreeDiscount, CELF, CELF++ e as heurísticas modificadas desses dois últimos, que são descritas na seção seguinte. Assim como foi feito com os conjuntos de sementes extraídos a partir das centralidades, as sementes selecionadas nesta etapa também terão suas similaridades comparadas, bem como as estimativas de difusão calculadas com os metamodelos.

4.3 Algoritmos propostos

Os algoritmos CELF e CELF++ possuem um ponto de melhoria a ser explorado. Originalmente, o cálculo da estimativa de difusão desses algoritmos é calculado através de simulações de Monte Carlo, em que são necessárias um número elevado de repetições para que se obtenha resultados satisfatórios. Normalmente utiliza-se 10000 repetições.

A ideia central é substituir as simulações de Monte Carlo por um metamodelo. Os métodos CELF e CELF++ modificados foram denominados FastCELF e FASTCELF++, respectivamente. Neste trabalho os metamodelos EDV e LIE foram usados para substituir as simulações em cada um dos algoritmos propostos. Com isso, tem-se no total quatro heurísticas: FastCELF EDV, FASTCELF++ EDV, FastCELF LIE e FASTCELF++ LIE.

4.3.1 Cálculo das estimativas de difusão com os metamodelos

A Equação 2.12 apresenta o cálculo do $\sigma_m(S)$, que corresponde à estimativa de difusão de um conjunto S para um modelo m através de simulações de Monte Carlo. Sabendo que calcular σ é a parte mais ineficiente dos algoritmos apresentados na Seção 3.1, e visando manter a mesma estrutura dos mesmos, propõe-se a substituição do σ por uma função γ.

A representação genérica da estimativa de propagação de um conjunto por um metamodelo é denotada por $\gamma_{\bar{m}}(S,\, p)$, em que S é o conjunto de sementes, $\bar{m}$ um metamodelo, e p é a probabilidade de propagação. Sendo assim, as estimativas de propagação calculadas pelos metamodelos EDV e LIE podem ser escritas de acordo com as respectivas equações:

$$\gamma_{EDV}(S,\, p) = EDV(S) \tag{4.1}$$

$$\gamma_{LIE}(S,\, p) = LIE(S) \tag{4.2}$$

que equivalem às Equações 2.13 e 2.14 apresentadas na Seção 2.5.

4.3.2 FastCELF

O pseudo-código do FastCELF é apresentado no Algoritmo 3. As modificações sugeridas se encontram nas linhas 4 e 15, em que substitui-se $\sigma_m(\{u\})$ por $\gamma(\{u\}, p)$, sendo γ um metamodelo qualquer e p a probabilidade de difusão. Como os metamodelos adotados neste trabalho exigem o parâmetro p para calcular a estimativa de difusão, o FastCELF também recebe esse parâmetro. A linha 15 poderia ser escrita também

4.3.3 FASTCELF++

Analogamente, o FastCELF++ também recebe um parâmetro p, por ser mandatório para os metamodelos aqui adotados. Em seu pseudo-código (Algoritmo 4), as

Algorithm 3 FastCELF

```
1:  function FastCELF(G, k, γ, p)
2:      S ← ∅, Q ← ∅
3:      for each u ∈ V do
4:          u.mg ← γ({u}, p)
5:          u.flag ← 0
6:          Q ← Q ∪ {u}
7:      end for
8:      heapify(Q)                                   ▷ baseado em u.mg
9:      while |S| < k do
10:         u ← Q[0]
11:         if u.flag = |S| then
12:             S ← S ∪ {u}
13:             Q ← Q − {u}
14:         else
15:             u.mg ← γ(S ∪ {u}, p) − γ(S, p)
16:             u.flag ← |S|
17:         end if
18:         heapify(Q)                               ▷ baseado em u.mg
19:     end while
20:     return S
21: end function
```

modificações com relação ao CELF++ se encontram nas linhas 5, 6, 22 e 24, correspondentes às chamadas da função σ. Assim como no FastCELF, $\sigma_m(\{u\})$ é substituído por $\gamma(\{u\}, p)$, que é a chamada de um metamodelo sendo executada apenas uma vez, no lugar das simulações de Monte Carlo. $\Delta_u^{\gamma}(cur\_best,\ p)$ corresponde ao ganho marginal de u com relação ao conjunto cur_best, calculado pelo metamodelo γ com probabilidade de difusão p. Na linha 22 do Algoritmo 4, por exemplo, o termo $u.mg1 \leftarrow \Delta_u^{\gamma}(S,\ p)$ poderia ser reescrito de forma análoga à linha 15 do Algoritmo 3, ou seja:

$$u.mg1 \leftarrow \Delta_u^{\gamma}(S,\ p) \ \equiv\ u.mg1 \leftarrow \gamma(S \cup \{u\},\ p) - \gamma(S,\ p)$$

4.3.4 Análise de complexidade

Os algoritmos CELF e CELF++ possuem complexidade de tempo idênticas para o modelo IC, $\mathcal{O}(k\mathcal{R}\mathcal{N}\mathcal{M})$ (BANERJEE, 2020; LI, 2018), onde k é o número de sementes a serem selecionadas, $\mathcal{R}$ é o número de simulações de Monte Carlo e $\mathcal{N}$ e $\mathcal{M}$ correspondem ao número de vértices e arestas da rede, respectivamente. O pior caso do modelo IC ocorre quando todos os vértices da rede são ativados durante o processo de cascata, uma vez que todas as arestas são percorridas nesse caso, portanto sua complexidade é $\mathcal{O}(\mathcal{M})$.

O primeiro laço do FastCELF (Algoritmo 3) nas linhas 3-7 tem complexidade de tempo $\mathcal{O}(\mathcal{N}\gamma_{\bar{m}})$, sendo $\mathcal{O}(\gamma_{\bar{m}})$ a complexidade do metamodelo $\bar{m}$. O laço das linhas 9-19 é executado $\mathcal{O}(k\mathcal{N})$ vezes, visto que a condição da linha 11 pode levar até $\mathcal{O}(\mathcal{N})$ iterações

Algorithm 4 FastCELF++

```
1: function FastCELF++(G, k, γ, p)
2:     S ← ∅, Q ← ∅
3:     last_seed ← NULL, cur_best ← NULL
4:     for each u ∈ V do
5:         u.mg1 ← γ({u}, p)
6:         u.mg2 ← Δ_u^γ(cur_best, p)
7:         u.prev_best ← cur_best
8:         u.flag ← 0
9:         Q ← Q ∪ {u}
10:    end for
11:    heapify(Q)                                   ▷ baseado em u.mg1
12:    while |S| < k do
13:        u ← Q[0]
14:        if u.flag = |S| then
15:            S ← S ∪ {u}
16:            last_seed ← u
17:            cur_best ← NULL
18:            Q ← Q − {u}
19:        else if u.prev_best = last_seed and u.flag = |S| − 1 then
20:            u.mg1 ← u.mg2
21:        else
22:            u.mg1 ← Δ_u^γ(S, p)
23:            u.prev_best ← cur_best
24:            u.mg2 ← Δ_u^γ(S ∪ {cur_best}, p)
25:        end if
26:        u.flag = |S|
27:        update cur_best
28:        heapify(Q)                               ▷ baseado em u.mg1
29:    end while
30:    return S
31: end function
```

para acontecer. O cálculo do ganho marginal na linha 15 tem complexidade $2\mathcal{O}(\gamma_{\bar{m}})$. Portanto a complexidade resultante do FastCELF para um metamodelo $\bar{m}$ qualquer é $\mathcal{O}(\mathcal{N}\gamma_{\bar{m}}) + 2\mathcal{O}(k\mathcal{N}\gamma_{\bar{m}}) \equiv \mathcal{O}(k\mathcal{N}\gamma_{\bar{m}})$.

No FastCELF++ (Algoritmo 4) a grande diferença está no número de vezes em que o cálculo da estimativa pelo metamodelo $\gamma_{\bar{m}}$ é realizado. No laço das linhas 4-10 as linha 5 e 6 possuem uma e duas chamadas ao $\gamma_{\bar{m}}$, respectivamente, executadas $\mathcal{O}(\mathcal{N})$ vezes, resultando em uma complexidade de $3\mathcal{O}(\mathcal{N}\gamma_{\bar{m}})$. No segundo laço (linhas 12-29) são quatro chamadas ao $\gamma_{\bar{m}}$, sendo duas na linha 22 e duas na linha 24. De forma análoga ao Algoritmo 3, o laço é executado $\mathcal{O}(k\mathcal{N})$ vezes. Como resultado, a complexidade do FastCELF++ para um metamodelo $\bar{m}$ qualquer é $3\mathcal{O}(\mathcal{N}\gamma_{\bar{m}}) + 4\mathcal{O}(k\mathcal{N}\gamma_{\bar{m}}) \equiv \mathcal{O}(k\mathcal{N}\gamma_{\bar{m}})$. Para efeitos de comparação, a Tabela 2 sumariza as complexidades descritas acima.

Tabela 2 – Complexidade de tempo dos algoritmos: k é número de sementes a serem selecionadas, $\mathcal{R}$ é o número de simulações de Monte Carlo, $\mathcal{N}$ é número de vértices, $\mathcal{M}$ é número de arestas e $\gamma_{\bar{m}}$ representa um metamodelo qualquer.

Rede	**Complexidade de tempo**
Algoritmo 1: CELF	$\mathcal{O}(k\mathcal{R}\mathcal{N}\mathcal{M})$
Algoritmo 2: CELF++	$\mathcal{O}(k\mathcal{R}\mathcal{N}\mathcal{M})$
Algoritmo 3: FastCELF	$\mathcal{O}(k\mathcal{N}\gamma_{\bar{m}})$
Algoritmo 4: FastCELF++	$\mathcal{O}(k\mathcal{N}\gamma_{\bar{m}})$

Considerando os metamodelos EDV e LIE, descritos na Seção 2.5, pode-se substituir $\gamma_{\bar{m}}$ pelas suas respectivas complexidades. Fazendo a substituição pela complexidade do EDV obtêm-se $\mathcal{O}(k\mathcal{N}k\bar{D}) \equiv \mathcal{O}(k^2\mathcal{N}\bar{D})$. Para o LIE a complexidade resultante é $\mathcal{O}(k\mathcal{N}k\bar{D}^2) \equiv \mathcal{O}(k^2\mathcal{N}\bar{D}^2)$. Apesar dos métodos propostos apresentarem complexidade quadrática em k, o valor desse termo tende a ser muito menor do que $\mathcal{N}$. A Tabela 3 apresenta as complexidades de tempo das heurísticas com os metamodelos apresentados na Seção 2.5.

Tabela 3 – Complexidade de tempo dos algoritmos com os metamodelos: k é número de sementes a serem selecionadas, $\mathcal{R}$ é o número de simulações de Monte Carlo, $\mathcal{N}$ é número de vértices, $\mathcal{M}$ é número de arestas e $\bar{D}$ é o grau médio da rede.

Rede	**Complexidade de tempo**
Algoritmo 3: FastCELF com EDV	$\mathcal{O}(k^2\mathcal{N}\bar{D})$
Algoritmo 4: FastCELF++ com EDV	$\mathcal{O}(k^2\mathcal{N}\bar{D})$
Algoritmo 3: FastCELF com LIE	$\mathcal{O}(k^2\mathcal{N}\bar{D}^2)$
Algoritmo 4: FastCELF++ com LIE	$\mathcal{O}(k^2\mathcal{N}\bar{D}^2)$

5 Resultados e discussões

Os resultados deste trabalho se dividem em duas partes. A Seção 5.2 apresenta a validação dos metamodelos, em que as estimativas de difusão calculadas pelo EDV e LIE são comparadas com o valor do σ para cada conjunto de sementes. Adicionalmente, também são analisadas as correlações entre os conjuntos de sementes extraídos com base nas medidas de centralidade das redes, utilizando o RBO como coeficiente de similaridade. A Seção 5.3 refere-se à eficiência aferida pelos métodos tradicionais e as heurísticas propostas, tanto em tempo de execução quanto na qualidade dos resultados. Também são analisadas as similaridades, com o RBO, entre os conjuntos de sementes extraídos pelos métodos.

Para não comprometer a experiência de leitura somente uma parte dos resultados está comentada nesta seção. Os resultados completos dos experimentos foram colocados no Apêndice.

5.1 Configuração do ambiente de testes

A fim de manter um padrão, foram estabelecidas algumas condições para a execução dos experimentos. O tamanho dos conjunto de sementes, k, varia de 5 até 50, com incrementos de 5 em 5, para o cálculo da propagação de influência. Para calcular a similaridade dos pares de conjuntos de sementes foi considerado o tamanho máximo dos conjuntos, 50. O modelo de difusão adotado foi o IC.

A propagação da influência não é muito sensível a diferentes algoritmos e heurísticas para o modelo IC com probabilidade de propagação p relativamente grande, porque uma componente conectado gigante existe mesmo após a remoção de todas as arestas com probabilidade $1 - p$. Isso foi relatado em (KEMPE, 2003) com $p = 0.1$. Por esse motivo, optou-se por manter a probabilidade de ativação p tanto para o IC quanto para os metamodelos em 0.01. (KEMPE, 2003) destaca ainda que a qualidade de aproximação de 10000 simulações é comparável à de 300000 ou mais, portanto, visando obter uma boa aproximação da propagação de influência pelo IC, o número de simulações de Monte Carlo adotado neste trabalho é 10000.

Todos os experimentos foram executados em uma máquina Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz, com 16 GB de memória RAM, sistema operacional Linux Ubuntu Server 20.04. Os algoritmos foram implementados na linguagem Python versão 3.8, com base nos pseudocódigos disponibilizados por (GOYAL, 2013).

5.2 Validação dos metamodelos

Nesta seção, são apresentados os resultados obtidos pelas estimativas de difusão calculadas com os metamodelos EDV e LIE com os conjuntos de sementes selecionados pelos *rankings* das medidas de centralidade. Esses experimentos incluem também os conjuntos de sementes obtidos com o PCA, cujo objetivo de utilização está descrito no Capítulo 4. Adicionalmente, também foram realizados experimentos sobre esses mesmos dados para obter a estimativa de propagação σ, ou seja, executando-se simulações de Monte Carlo com o modelo de difusão IC para cada conjunto de sementes das redes (Equação 2.12).

Previamente, observou-se que os conjuntos grau, *PageRank* e *betweenness* geralmente alcançam os melhores resultados de difusão entre os conjuntos baseados em centralidade. Sabendo disso, o conjunto PCA foi extraído a partir da redução de dimensionalidade dessas três centralidades, conforme foi justificado no Capítulo 4.

Pode-se notar nas Figuras 2 e 3 um comportamento idêntico entre os valores das estimativas γ_{EDV}, γ_{LIE} e σ. O eixo X representa a variação do tamanho do conjunto de sementes e o eixo Y o valor calculado pela respectiva estimativa. Para o σ o eixo Y representa a média de vértices ativados no processo de difusão. Na rede "CA-GrQc", os metamodelos capturaram também a queda de propagação observado nas centralidades de grau e autovetor, a partir de 30 sementes. Algo semelhante ocorreu com a rede "CA-HepTh", como é possível ver na Figura 3: em certo ponto a centralidade de autovetor foi superada pelas sementes aleatórias.

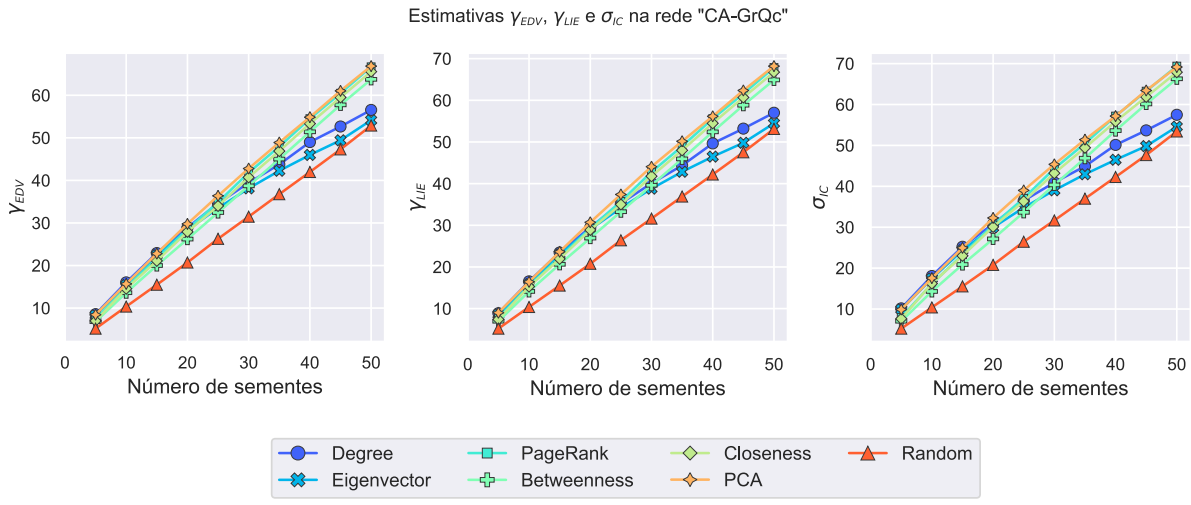

Figura 2 – Comparativo das estimativas de difusão na rede "CA-GrQc": γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

Entretanto, a principal diferença entre γ_{EDV}, γ_{LIE} e σ é que os valores de σ tendem a ser superiores aos metamodelos. A Figura 5 evidencia isso. O valor menor da estimativa

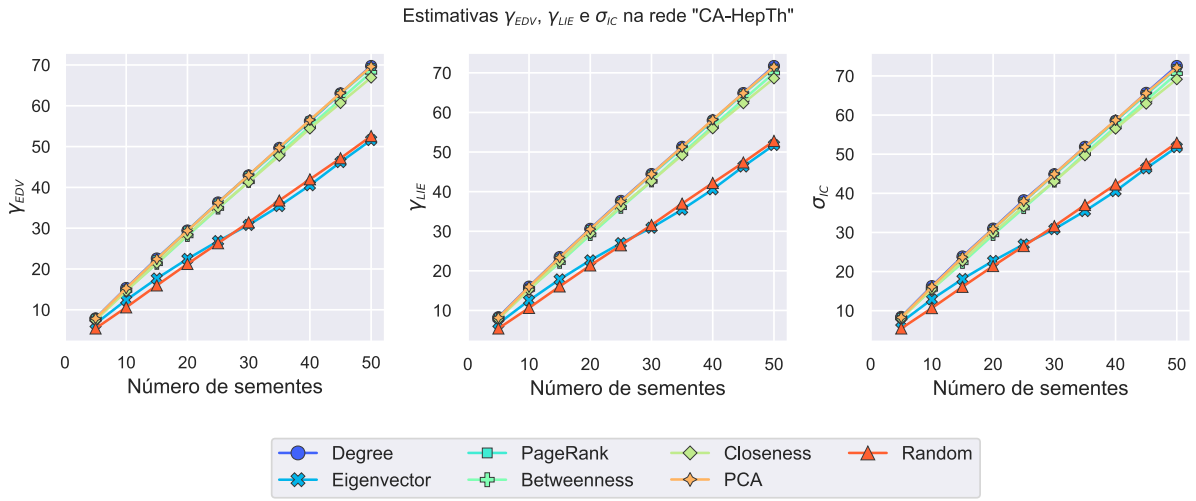

Figura 3 – Comparativo das estimativas de difusão na rede "CA-HepTh": γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

de difusão não é crucial para o que está sendo proposto neste trabalho, uma vez que os metamodelos tenham projeções semelhantes ao σ, como está sendo mostrado.

Na Figura 4 percebe-se que a correlação entre conjuntos de sementes reafirmam o que foi observado com as estimativas de difusão. As centralidades de grau e autovetor na rede "CA-GrQc" apresentam alta correlação entre si, e viu-se na Figura 2 um decaimento na estimativa de difusão justamente desses dois conjuntos de sementes. As outras centralidades se mostraram superiores, portanto já era esperado que elas tivessem uma correlação mais alta entre si e uma menor similaridade com grau e autovetor. Também na Figura 4, o que mais chama atenção é a baixa correlação entre autovetor e os demais conjuntos na rede "CA-HepTh". E vemos na Figura 3 que em suas estimativas de difusão, autovetor teve qualidade comparável a um conjunto aleatório de sementes, se distanciando consideravelmente das demais centralidades.

Nas Figuras 5 e 6 percebe-se que as centralidades de grau, PageRank e PCA foram as melhores nas estimativas de propagação. Em contrapartida, autovetor e *closeness* foram as piores. As correlações apresentadas na Figura 7 confirmam a alta similaridade entre os melhores conjuntos e a baixa similaridade dos conjuntos autovetor e *closeness* com as demais.

Com base nos resultados obtidos nessa etapa, pode-se concluir que: (1) os metamodelos utilizados neste trabalho, EDV e LIE, são capazes de estimar qualitativamente a difusão na rede, mas não quantitativamente, pois geralmente os valores de σ são mais altos do que as estimativas γ_{EDV} e γ_{LIE}; (2) o RBO mostrou-se capaz de aferir as similaridades de conjuntos de sementes, mostrando-se uma ferramenta bastante robusta para corroborar os resultados comparativos das estimavas de difusão dos metamodelos; (3) os conjuntos baseados na centralidade de grau, no geral, se saíram melhor no σ, enquanto que

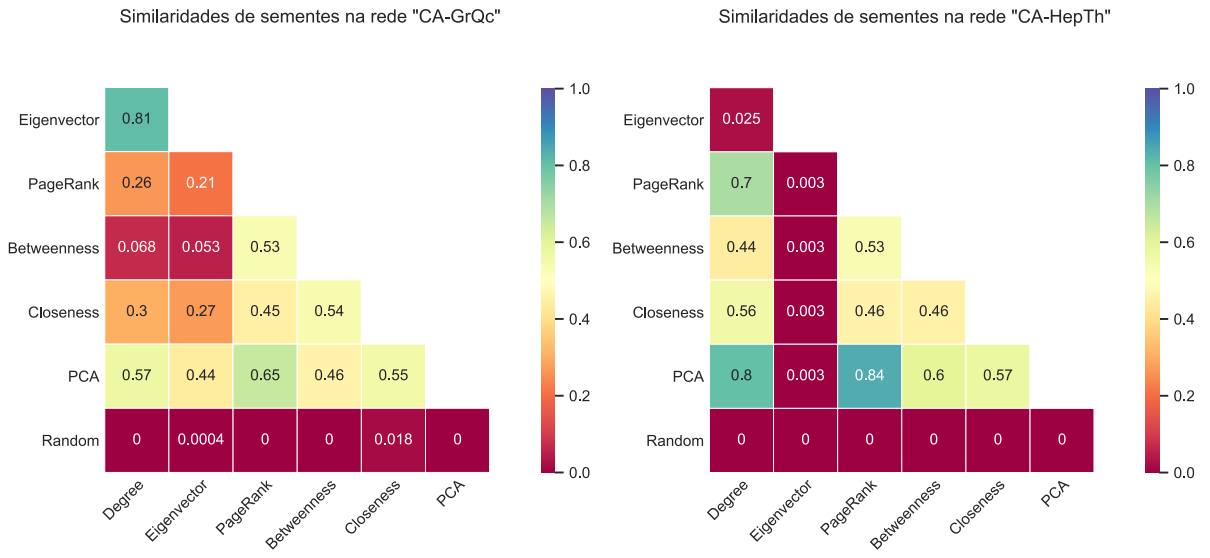

Figura 4 – Correlações dos conjuntos de sementes baseados em centralidade: rede "CA-GrQc" à esquerda rede e rede "CA-HepTh" à direita.

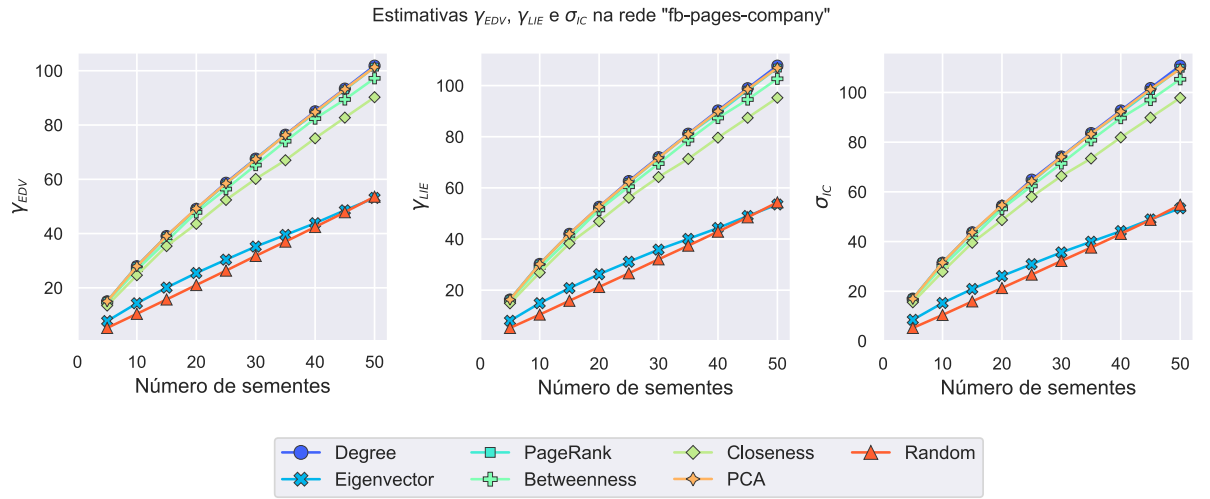

Figura 5 – Comparativo das estimativas de difusão na rede "fb-pages-company": γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

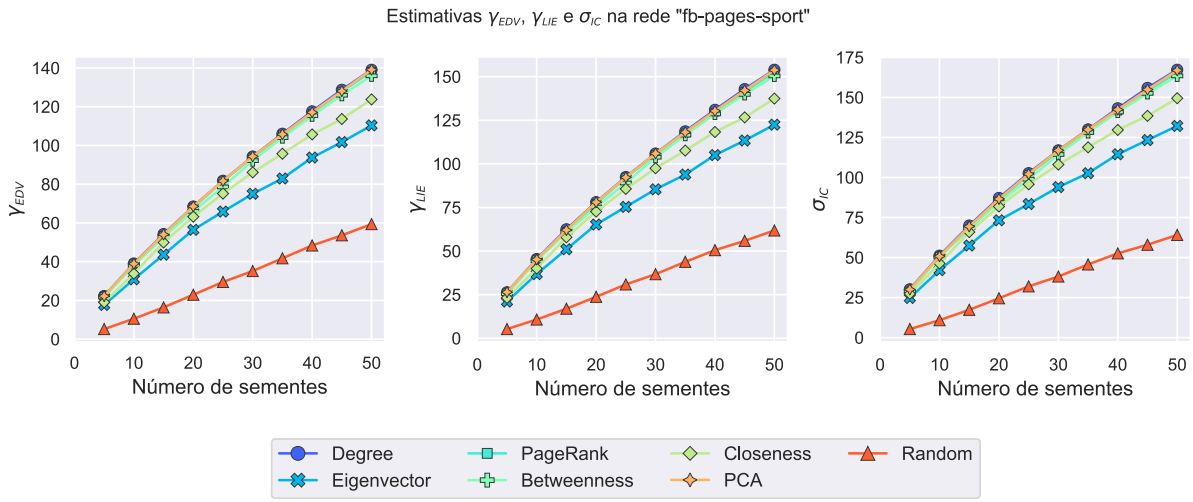

Figura 6 – Comparativo das estimativas de difusão na rede "fb-pages-sport": γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

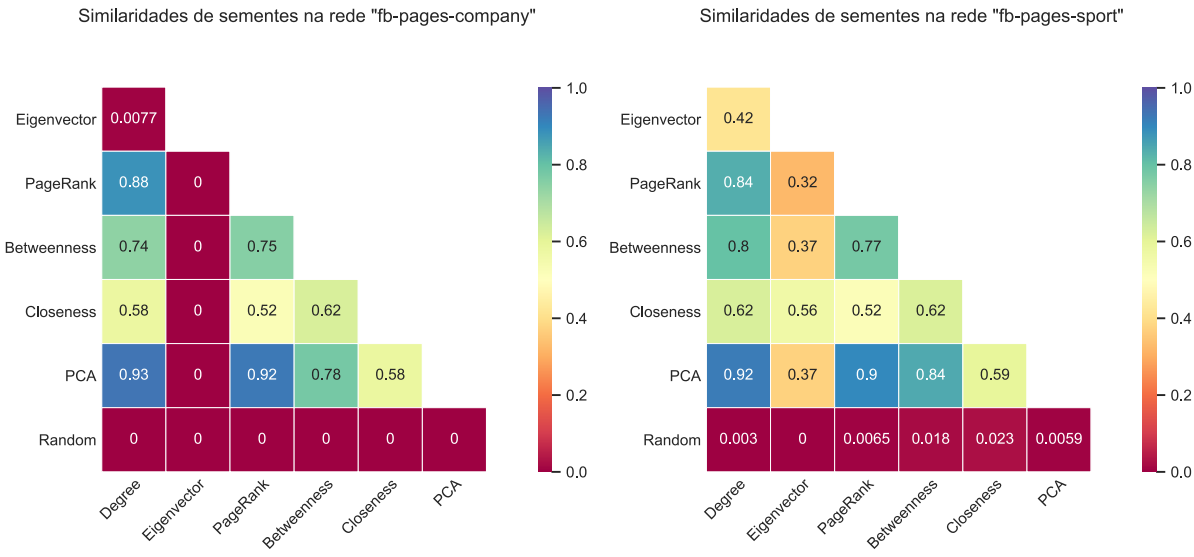

Figura 7 – Correlações dos conjuntos de sementes baseados em centralidade: rede "fb-pages-company" à esquerda rede e rede "fb-pages-sport" à direita.

o PCA se mostrou mais estável, se mantendo entre os melhores conjuntos em propagação (considerando apenas os conjuntos avaliados até este ponto).

5.3 Eficiências das Heurísticas

Na seção anterior os resultados apontaram para a viabilidade das alterações sugeridas nos algoritmos CELF e CELF++. Sabendo que os metamodelos geram estimativas condizentes com o cálculo do σ, os algoritmos propostos FastCELF e FastCELF++ foram avaliados utilizando tanto o EDV quanto o LIE. Com isso, este trabalho conta com quatro métodos inéditos, capazes de alcançar resultados competitivos em qualidade de difusão e extremamente melhores em tempo de execução, comparados com os algoritmos originais. Para efeitos de comparação, o algoritmo DegreeDiscount foi avaliado em conjunto com os algoritmos originais, CELF e CELF++, e as heurísticas propostas.

Nessa etapa, cada heurística foi utilizada para selecionar 50 sementes de cada rede. Em seguida as similaridades das sementes foram aferidas par a par com o RBO e por fim iniciou-se o processo de difusão com as sementes. Vale ressaltar que nesse processo de difusão utilizam-se simulações de Monte Carlo para todos os métodos, incluindo os propostos. O que está sendo proposto neste trabalho é a substituição das simulações de Monte Carlo que são usadas para a **seleção** de sementes nos algoritmo CELF e CELF++. A avaliação da eficiência dos métodos só pode ser feita pelo processo de difusão, que utiliza das simulações de Monte Carlo para obter boa acurácia, por se tratar de um processo estocástico. As configurações dos experimentos são as mesmas já mencionadas na Seção 5.1.

5.3.1 Tempo de execução

O tempo de execução dos algoritmos originais para a seleção de sementes, CELF e CELF++, juntamente com suas respectivas versões modificadas são apresentados na Figura 8. Os métodos propostos quando utilizados em conjunto com o EDV foram mais rápidos, demorando menos de 5 segundos na rede "slashdot0902", a maior rede da base (82 mil nós e 870 mil arestas), para selecionar as sementes. Na rede "netscience", a menor do conjunto, o tempo foi inferior a 1 segundo para as quatro heurísticas. Por outro lado, é visível a ineficiência do CELF e CELF++ nessa rede: um pouco mais de 2 minutos e 3 minutos, respectivamente. O tempo de execução aumenta exponencialmente conforme o tamanho das redes aumenta, chegando a levar cada um dos dois algoritmos aproximadamente 23 horas para terminar a execução na rede "fb-pages-company". A partir dessa rede os dois métodos não foram mais executados por ser impraticável.

As versões modificadas FastCELF e FastCELF++ que utilizam o LIE como função de estimativa demoram um pouco mais do que as versões com o EDV. Isso acontece

porque o LIE é considera uma vizinhança maior do que o EDV para calcular sua estimativa e para cada vértice avaliado por ele, o número de arestas entre as vizinhanças a 1 e a 2 saltos de distância são contados, enquanto que o EDV considera apenas a vizinhança imediata do vértice. Mesmo sendo menos eficiente em tempo de execução, as versões modificadas em conjunto com o LIE ainda assim são muito mais rápidas do que os algoritmos originais: entre os métodos propostos, o FastCELF++ LIE foi o mais demorado na rede "fb-pages-company", levando 30 segundos para terminar sua execução, um ganho formidável perto das 23 horas dos métodos originais. A complexidade dos algoritmos foi discutida na Subseção 4.3.4.

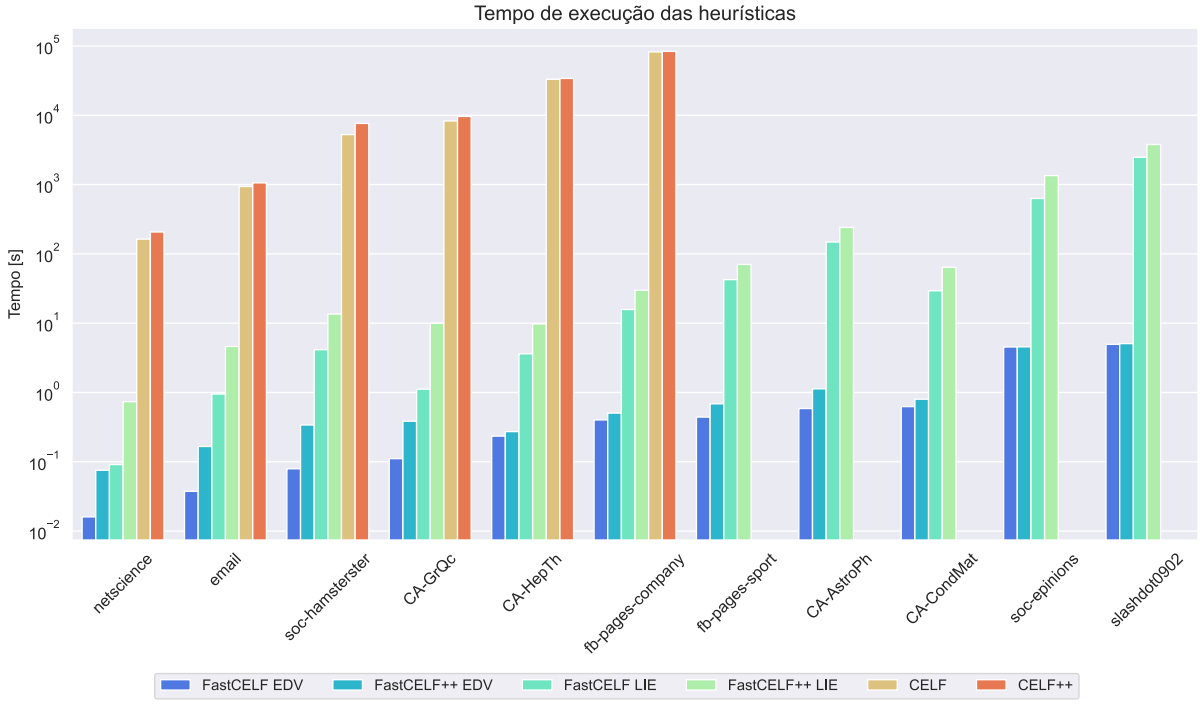

Figura 8 – Tempo de execução dos algoritmos para a seleção de sementes.

5.3.2 Propagação de influência

Quanto à qualidade de propagação de influência, os resultados mostraram que os algoritmos propostos são competitivos e superaram as versões originais em todas as redes em que a execução foi possível, mesmo que por uma margem pequena.

A Figura 9 apresenta os resultados de propagação das heurísticas e a correlação entre suas sementes na rede "CA-GrQc". Os métodos *DegreeDiscount*, FastCELF++ LIE, FastCELF++ EDV, nesta ordem, apresentam os melhores resultados e superam com folga os algoritmos CELF e CELF++. As correlações das sementes mostram que altas similaridades refletem a performance das heurísticas. CELF e CELF++ são similares aos algoritmos FastCELF que adotam EDV e LIE.

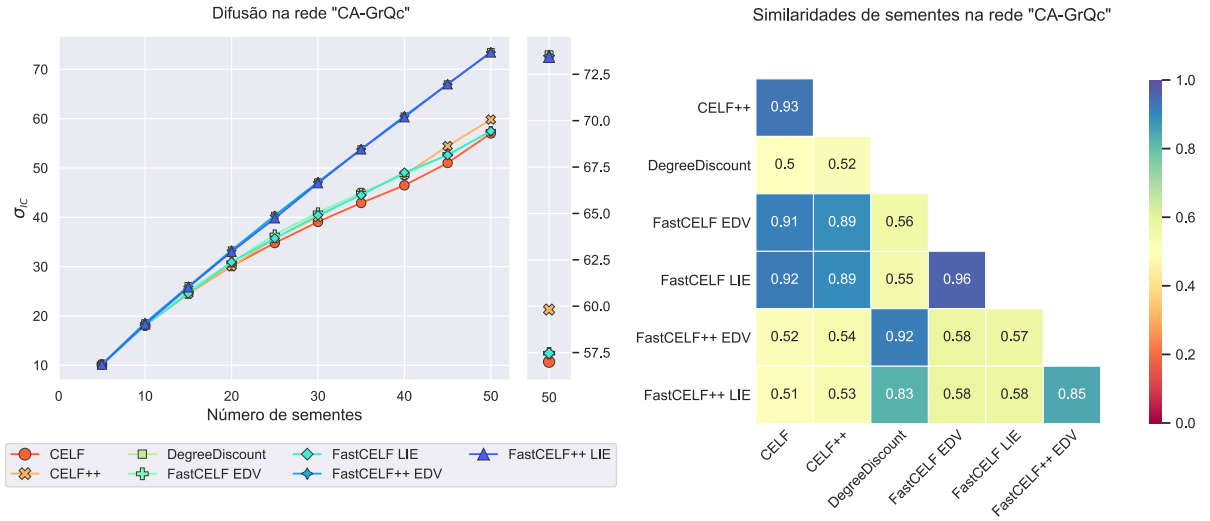

Figura 9 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede "CA-GrQc".

Na rede "soc-hamsterster" (Figura 10) as correlações são altas, mas ainda assim percebe-se que o mesmo trio de algoritmos, *DegreeDiscount*, FastCELF++ LIE, FastCELF++ EDV, são os pares de maior correlação. Aqui o FastCELF++ LIE supera o *DegreeDiscount* por uma margem pequena.

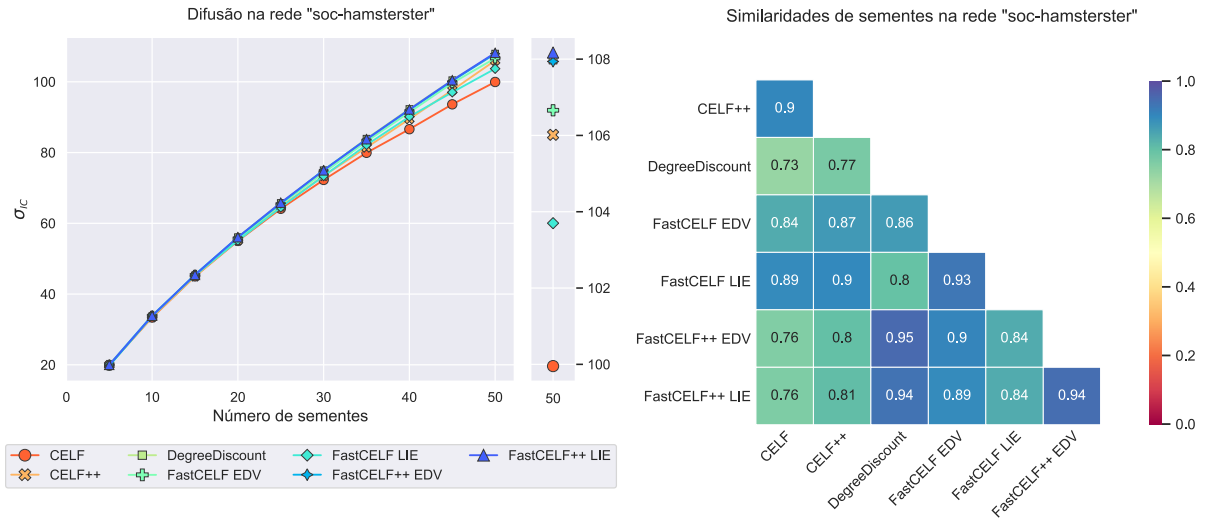

Figura 10 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede "soc-hamsterster".

Os resultados de difusão são comparáveis nas redes "CA-HepTh" e "fb-pages-company", como pode ser visto na Figura 11. Mesmo assim, o FastCELF++ e *DegreeDiscount* superam os demais.

Nas Figuras 12 e 13 os resultados da difusão mostram que o *DegreeDiscount* foi

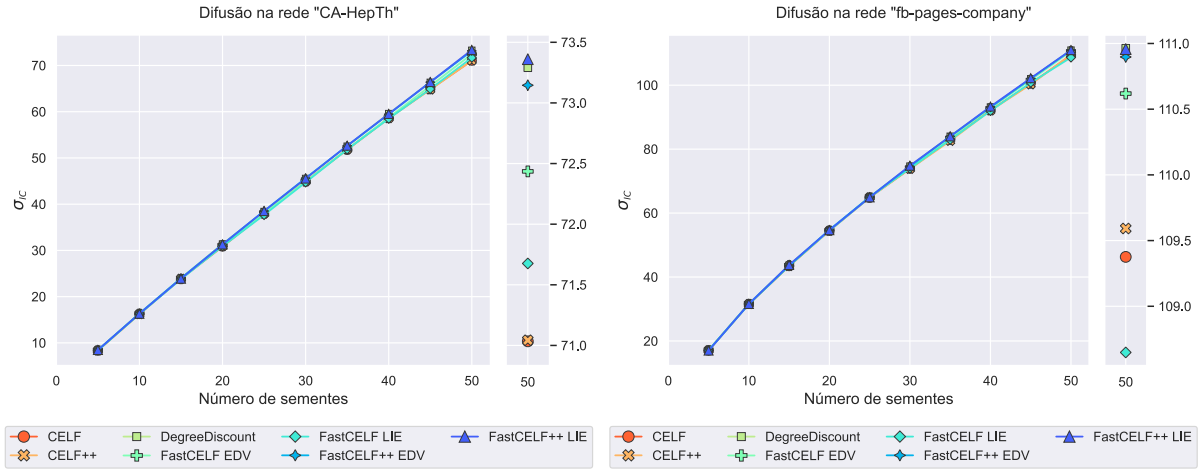

Figura 11 – Propagação de influência dos conjuntos de sementes selecionadas pelas heurísticas nas redes "CA-HepTh" (à esquerda) e "fb-pages-company" (à direita).

superior ao FastCELF++ redes "soc-epinions" e "slashdot0902". O que pode explicar a superioridade do *DegreeDiscount* nesse caso é que as sementes selecionadas em ambas as redes pelo FastCELF++ aparecem mais correlacionadas com as sementes selecionadas pelo FastCELF EDV, que possuem menor qualidade de propagação, e menos similares com as sementes do *DegreeDiscount*. O FastCELF LIE alcançou os piores resultados nesse cenário por descartar vértices que teriam melhor contribuição para os resultados, e isso é comprovado pelas correlações mais baixas entre o FastCELF LIE e os demais métodos.

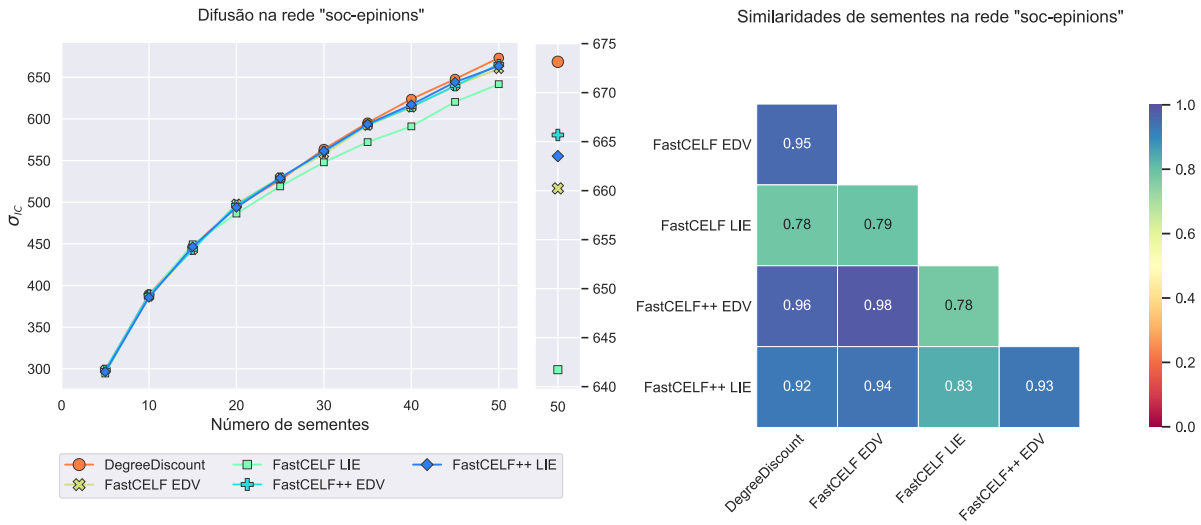

Figura 12 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede "soc-epinions".

No Apêndice estão todas as curvas de difusão e mapas de similaridades dos conjuntos de sementes.

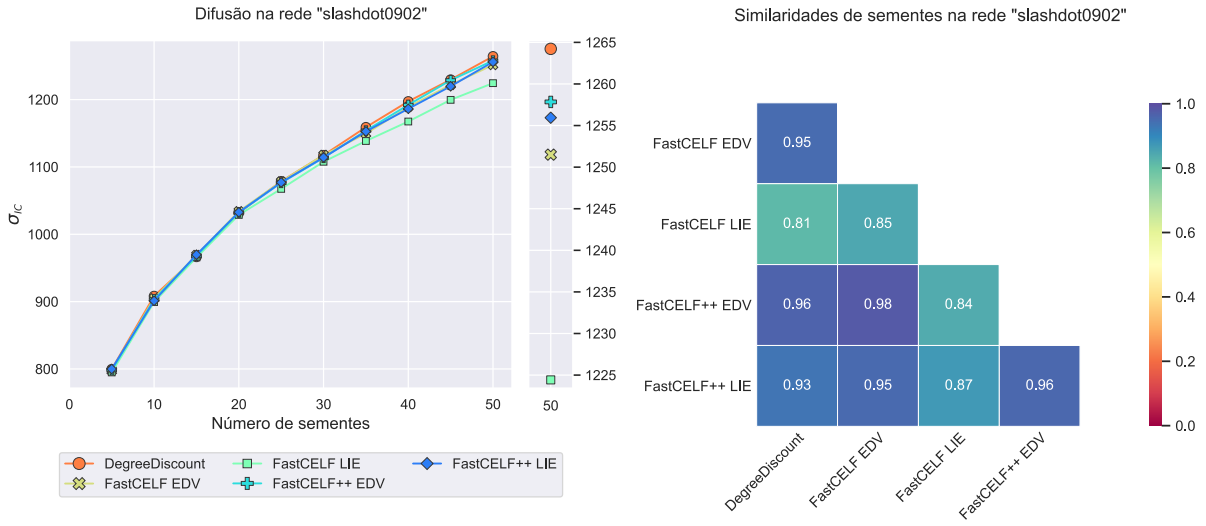

Figura 13 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “slashdot0902”.

6 Considerações finais

O Problema de Maximização de Influência em Redes Complexas (IMP) é um problema NP-difícil, portanto, faz-se necessário o uso de estratégias aproximadas capazes de garantir uma solução satisfatória com um tempo de execução viável. Este trabalho apresenta os conceitos teóricos necessários para a execução da metodologia proposta. Muitos estudos tentam superar os resultados alcançados pelas estratégias gulosas mais conhecidas na literatura. Acredita-se que seleção de sementes seja a mais importante para o problema IMP, por estar diretamente relacionada ao desempenho dos modelos de difusão, portanto a criação de novos métodos inteligentes ou o aperfeiçoamento das estratégias já existentes torna-se importante.

No geral, as heurísticas implementadas se mostraram comparáveis ao *Degree-Discount* em propagação de influência, mantendo uma margem pequena de diferença. Mostrou-se, portanto, a viabilidade de substituir as simulações de Monte Carlo nos algoritmos CELF e CELF++, por metamodelos, alcançando ainda um ganho na qualidade de propagação. A metodologia desenvolvida foi suficiente para alcançar os objetivos propostos neste trabalho. As correlações entre pares de conjuntos de sementes com o RBO auxiliaram o entendimento dos resultados de difusão e serviram para mostrar o nível de similaridade entre as sementes selecionadas pelo métodos clássicos e as sementes selecionadas pelas heurísticas apresentadas. Os metamodelos conseguem estimar a propagação de influência de forma rápida e ainda manter resultados comparáveis qualitativamente às simulações de Monte Carlo. Os métodos desenvolvidos possuem boa eficiência e se mostraram computacionalmente escaláveis.

Espera-se que os resultados deste trabalho tragam uma contribuição importante para a área, apontando para melhores estratégias que apresentem bom desempenho, eficiência e rapidez para o problema de IMP.

6.1 Trabalhos futuros

Existem algumas direções que ainda pode ser exploradas a partir deste trabalho. A primeira delas é a utilização de outros modelos de difusão, que pode estabelecer os algoritmos propostos, caso tenham bons resultados também com esses modelos. Adicionalmente, uma sequência da pesquisa realizada neste trabalho é a variação da probabilidade de ativação p, que pode indicar se os metamodelos são sensíveis a esse parâmetro. Por fim, um novo metamodelo pode ser desenvolvido e ser quantitativamente comparável ao cálculo da função de propagação.

Referências

MA, H.; YANG, H.; LYU, M. R.; KING, I. Mining social networks using heat diffusion processes for marketing candidates selection. In: ACM. *Proceedings of the 17th ACM conference on Information and knowledge management*. [S.l.], 2008. p. 233–242. Citado 2 vezes nas páginas 14 e 25.

CAMPAN, A.; CUZZOCREA, A.; TRUTA, T. M. Fighting fake news spread in online social networks: Actual trends and future research directions. In: IEEE. *2017 IEEE International Conference on Big Data (Big Data)*. [S.l.], 2017. p. 4453–4457. Citado 2 vezes nas páginas 14 e 25.

BAMBROUGH, B. *'Release The Doge!'-Elon Musk Gives The Dogecoin Price A Sudden Boost As Bitcoin And Ethereum Lead The Crypto Market Lower*. Forbes Magazine, Jul 2021. Acessado em 14/07/2021. Disponível em: <https://www.forbes.com/sites-/billybambrough/2021/07/01/release-the-doge-elon-musk-gives-the-dogecoin-price-a-sudden-boost-as-bitcoin-and-ethereum-lead-the-crypto-market-lower>. Citado na página 14.

SONG, X.; TSENG, B. L.; LIN, C.-Y.; SUN, M.-T. Personalized recommendation driven by information flow. In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. [S.l.: s.n.], 2006. p. 509–516. Citado na página 14.

IENCO, D.; BONCHI, F.; CASTILLO, C. The meme ranking problem: Maximizing microblogging virality. In: IEEE. *2010 IEEE International Conference on Data Mining Workshops*. [S.l.], 2010. p. 328–335. Citado na página 14.

BAKSHY, E.; HOFMAN, J. M.; MASON, W. A.; WATTS, D. J. Everyone's an influencer: quantifying influence on twitter. In: *Proceedings of the fourth ACM international conference on Web search and data mining*. [S.l.: s.n.], 2011. p. 65–74. Citado na página 14.

LI, Y.; ZHANG, D.; TAN, K.-L. Real-time targeted influence maximization for online advertisements. VLDB Endowment, 2015. Citado na página 14.

KEMPE, D.; KLEINBERG, J.; TARDOS, É. Maximizing the spread of influence through a social network. In: ACM. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2003. p. 137–146. Citado 6 vezes nas páginas 15, 25, 26, 32, 37 e 44.

LESKOVEC, J.; KRAUSE, A.; GUESTRIN, C.; FALOUTSOS, C.; VANBRIESEN, J.; GLANCE, N. Cost-effective outbreak detection in networks. In: ACM. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2007. p. 420–429. Citado 4 vezes nas páginas 15, 27, 33 e 36.

GOYAL, A.; LU, W.; LAKSHMANAN, L. V. Celf++ optimizing the greedy algorithm for influence maximization in social networks. In: *Proceedings of the 20th international conference companion on World wide web*. [S.l.: s.n.], 2011. p. 47–48. Citado 4 vezes nas páginas 15, 26, 27 e 34.

JIANG, Q.; SONG, G.; GAO, C.; WANG, Y.; SI, W.; XIE, K. Simulated annealing based influence maximization in social networks. In: *Twenty-fifth AAAI conference on artificial intelligence*. [S.l.: s.n.], 2011. Citado 3 vezes nas páginas 15, 28 e 35.

GONG, M.; YAN, J.; SHEN, B.; MA, L.; CAI, Q. Influence maximization in social networks based on discrete particle swarm optimization. *Information Sciences*, Elsevier, v. 367, p. 600–614, 2016. Citado 2 vezes nas páginas 16 e 28.

SCHMITH, J.; LEMKE, N.; MOMBACH, J.; BENELLI, P.; BARCELLOS, C. K.; BEDIN, G. B. Damage, connectivity and essentiality in protein–protein interaction networks. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 349, n. 3, p. 675–684, 2005. Citado na página 18.

ANDERSON, T. K.; LAEGREID, W. W.; CERUTTI, F.; OSORIO, F. A.; NELSON, E. A.; CHRISTOPHER-HENNINGS, J.; GOLDBERG, T. L. Ranking viruses: measures of positional importance within networks define core viruses for rational polyvalent vaccine development. *Bioinformatics*, Oxford Univ Press, v. 28, n. 12, p. 1624–1632, 2012. Citado na página 18.

FREEMAN, L. C. Centrality in social networks conceptual clarification. *Social networks*, Elsevier, v. 1, n. 3, p. 215–239, 1979. Citado 2 vezes nas páginas 18 e 21.

GIRVAN, M.; NEWMAN, M. E. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 99, n. 12, p. 7821–7826, 2002. Citado na página 18.

KNUTH, D. E.; KNUTH, D. E.; KNUTH, D. E. *The Stanford GraphBase: a platform for combinatorial computing*. [S.l.]: Addison-Wesley Reading, 1993. Citado na página 18.

NEWMAN, M. *Networks: an introduction*. [S.l.]: Oxford University Press, 2009. Citado 2 vezes nas páginas 20 e 21.

BONACICH, P. Power and centrality: A family of measures. *American journal of sociology*, JSTOR, p. 1170–1182, 1987. Citado 2 vezes nas páginas 20 e 21.

FREEMAN, L. C. A set of measures of centrality based on betweenness. *Sociometry*, JSTOR, p. 35–41, 1977. Citado na página 22.

PAGE, L.; BRIN, S.; MOTWANI, R.; WINOGRAD, T. The pagerank citation ranking: bringing order to the web. Stanford InfoLab, 1999. Citado na página 23.

LANGVILLE, A. N.; MEYER, C. D. *Google's PageRank and beyond: The science of search engine rankings*. [S.l.]: Princeton University Press, 2006. Citado na página 23.

TAN, P.-N. *Introduction to data mining*. [S.l.]: Pearson Education India, 2018. Citado na página 24.

CHEN, W.; WANG, Y.; YANG, S. Efficient influence maximization in social networks. In: ACM. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2009. p. 199–208. Citado 3 vezes nas páginas 27, 34 e 35.

CHEN, W.; YUAN, Y.; ZHANG, L. Scalable influence maximization in social networks under the linear threshold model. In: IEEE. *2010 IEEE international conference on data mining*. [S.l.], 2010. p. 88–97. Citado 2 vezes nas páginas 27 e 34.

MITZENMACHER, M.; UPFAL, E. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. [S.l.]: Cambridge university press, 2017. Citado na página 27.

HOYLE, N. *Automated multi-stage geometry parameterization of internal fluid flow applications*. Tese (Doutorado) — University of Southampton, 2006. Citado na página 28.

TANG, J.; ZHANG, R.; YAO, Y.; ZHAO, Z.; WANG, P.; LI, H.; YUAN, J. Maximizing the spread of influence via the collective intelligence of discrete bat algorithm. *Knowledge-Based Systems*, Elsevier, v. 160, p. 88–103, 2018. Citado na página 29.

KENDALL, M. G. Rank correlation methods. Griffin, 1948. Citado na página 30.

KRISHNAN, V. On pearson, spearman and kendall correlation coefficients. 2012. Citado na página 30.

WEBBER, W.; MOFFAT, A.; ZOBEL, J. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, ACM New York, NY, USA, v. 28, n. 4, p. 1–38, 2010. Citado na página 30.

WANG, Y.; CONG, G.; SONG, G.; XIE, K. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In: ACM. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2010. p. 1039–1048. Citado na página 35.

BUCUR, D.; IACCA, G. Influence maximization in social networks with genetic algorithms. In: SPRINGER. *European Conference on the Applications of Evolutionary Computation*. [S.l.], 2016. p. 379–392. Citado na página 36.

CUI, L.; HU, H.; YU, S.; YAN, Q.; MING, Z.; WEN, Z.; LU, N. Ddse: A novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks. *Journal of Network and Computer Applications*, Elsevier, v. 103, p. 119–130, 2018. Citado na página 36.

LESKOVEC, J.; KREVL, A. *SNAP Datasets: Stanford Large Network Dataset Collection*. jun. 2014. `http://snap.stanford.edu/data`. Citado na página 38.

ROSSI, R. A.; AHMED, N. K. The network data repository with interactive graph analytics and visualization. In: *AAAI*. [s.n.], 2015. Disponível em: <http://-/networkrepository.co>. Citado na página 38.

BANERJEE, S.; JENAMANI, M.; PRATIHAR, D. K. A survey on influence maximization in a social network. *Knowledge and Information Systems*, Springer, v. 62, n. 9, p. 3417–3455, 2020. Citado na página 41.

LI, Y.; FAN, J.; WANG, Y.; TAN, K.-L. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 30, n. 10, p. 1852–1872, 2018. Citado na página 41.

GOYAL, A. Social influence and its applications : an algorithmic and data mining study. In: . [S.l.: s.n.], 2013. Citado na página 44.

Apêndices

APÊNDICE A – Comparativo das estimativas de difusão dos metamodelos e σ

Neste capítulo estão agrupados os resultados das estimativas de difusão aferidas pelos metamodelos EDV e LIE, comparando-os com a difusão calculada pelo σ. No eixo X apresenta o número de sementes e no eixo Y mostra o valor calculado pelas estimativas.

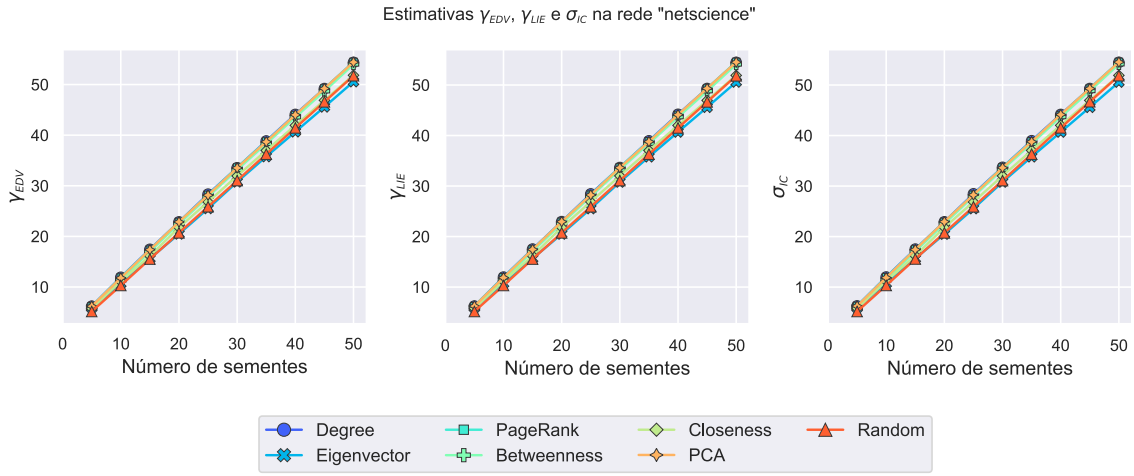

Figura 14 – Comparativo das estimativas de difusão na rede *netscience*: γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

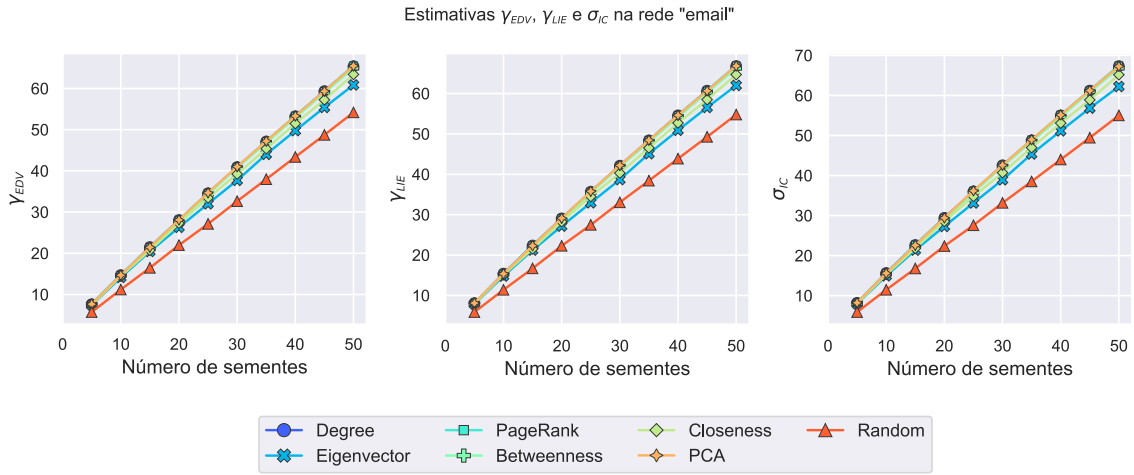

Figura 15 – Comparativo das estimativas de difusão na rede *email*: γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

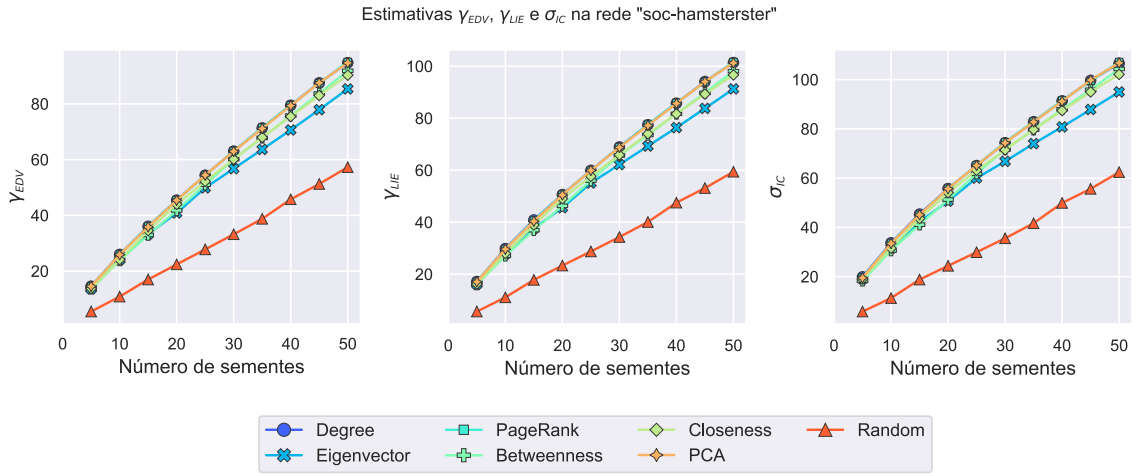

Figura 16 – Comparativo das estimativas de difusão na rede *soc-hamsterster*: γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

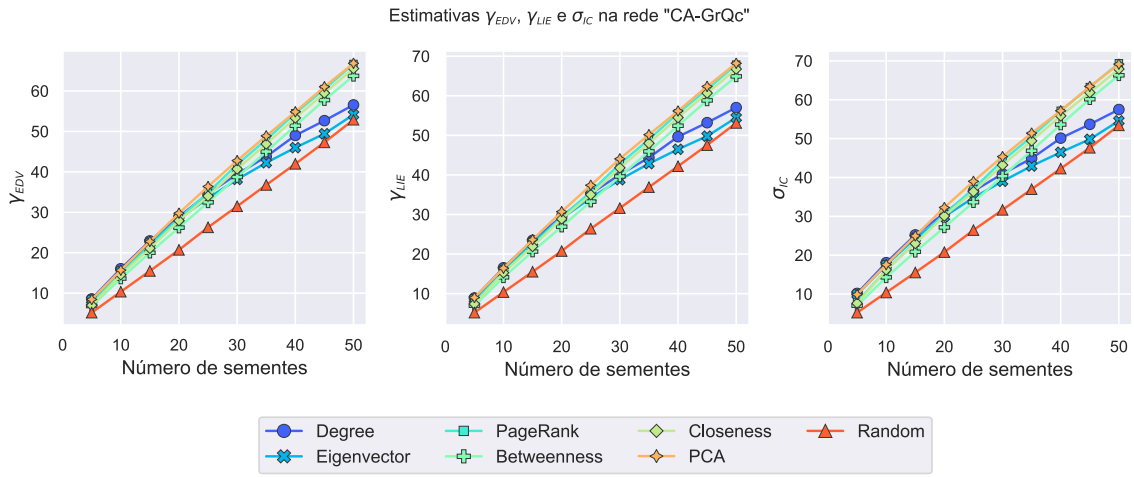

Figura 17 – Comparativo das estimativas de difusão na rede *CA-GrQc*: γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

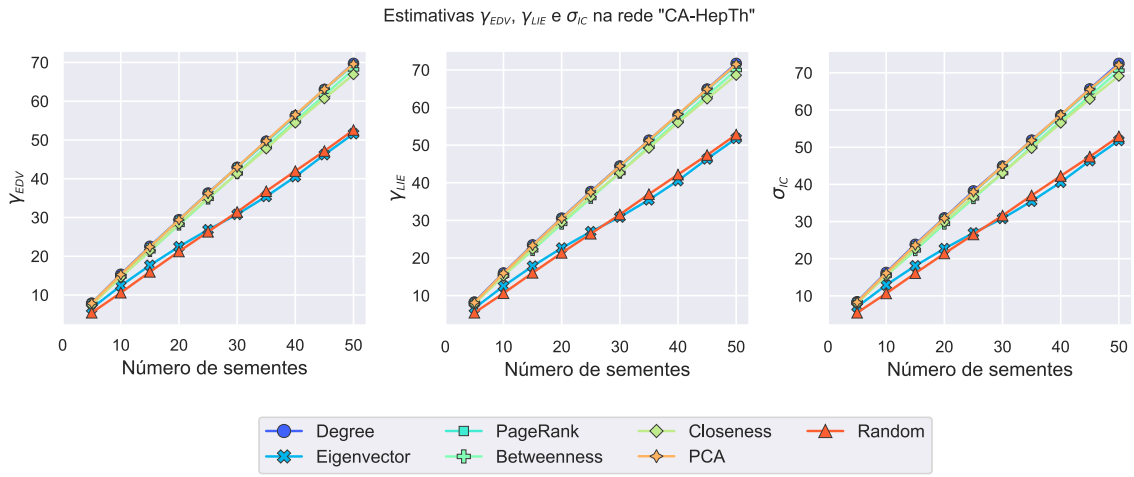

Figura 18 – Comparativo das estimativas de difusão na rede *CA-HepTh*: γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

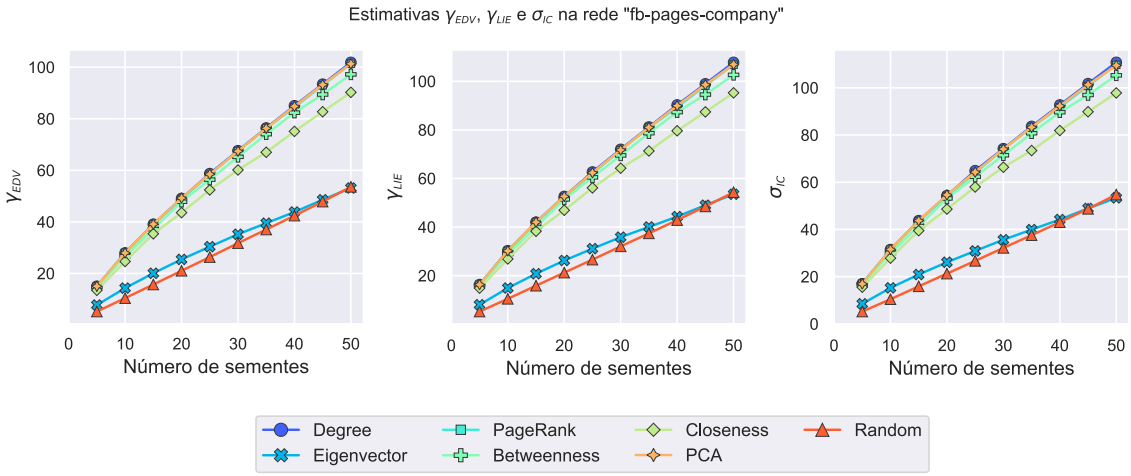

Figura 19 – Comparativo das estimativas de difusão na rede *fb-pages-company*: γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

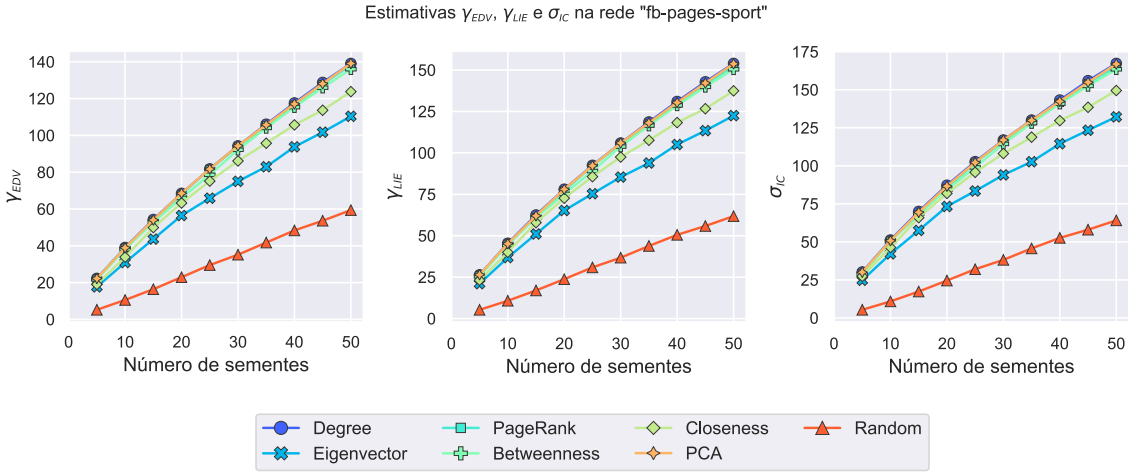

Figura 20 – Comparativo das estimativas de difusão na rede *fb-pages-sport*: γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

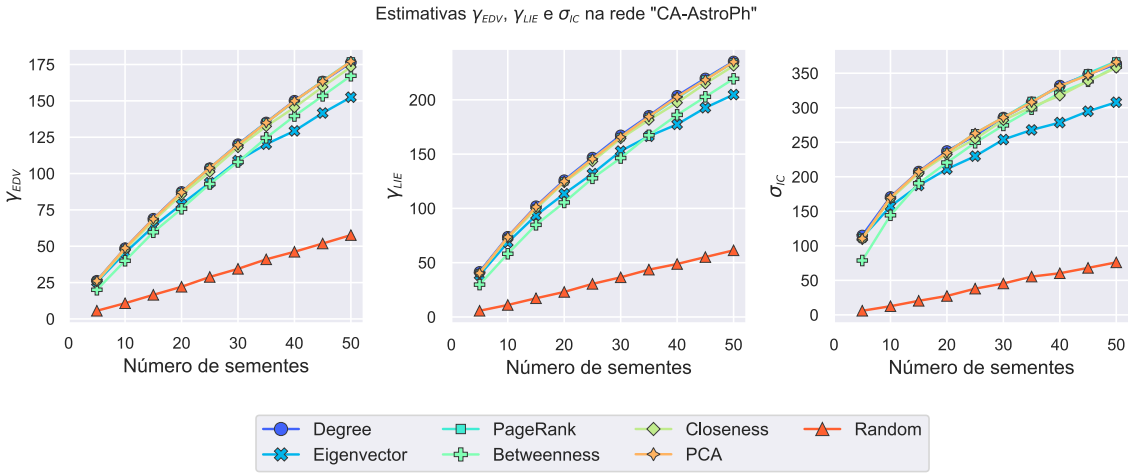

Figura 21 – Comparativo das estimativas de difusão na rede *CA-AstroPh*: γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

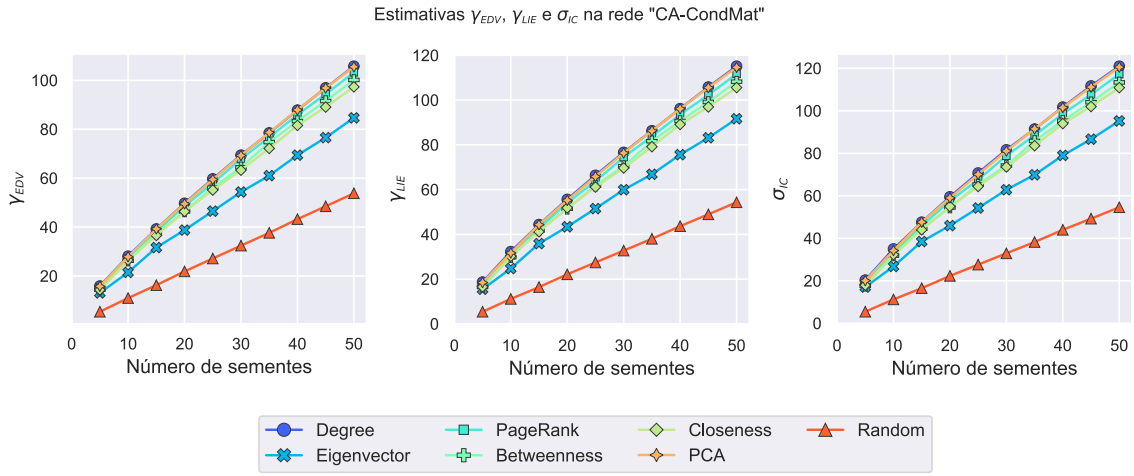

Figura 22 – Comparativo das estimativas de difusão na rede *CA-CondMat*: γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

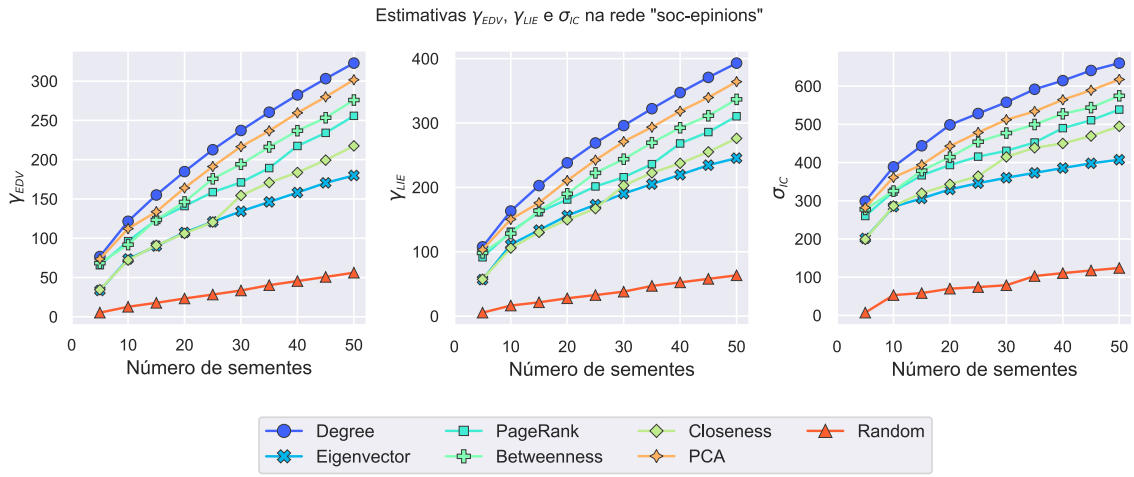

Figura 23 – Comparativo das estimativas de difusão na rede *soc-epinions*: γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

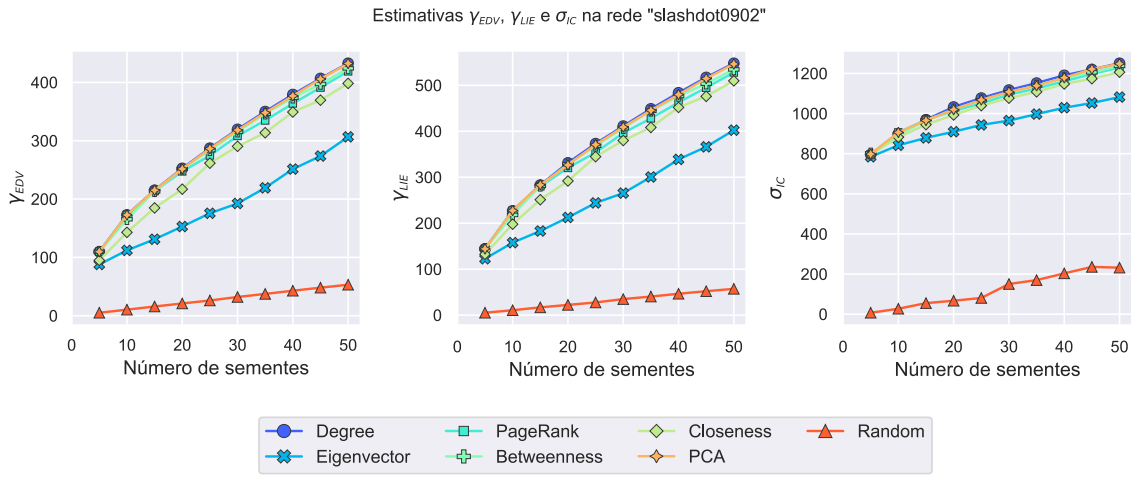

Figura 24 – Comparativo das estimativas de difusão na rede *slashdot0902*: γ_{EDV} à esquerda, γ_{LIE} ao centro e σ à direita.

APÊNDICE B – Propagação de influência e correlação entre conjuntos de sementes - centralidades

Neste capítulo estão agrupados os resultados da propagação de influência e das correlações entre os conjuntos de sementes baseados em medidas de centralidade.

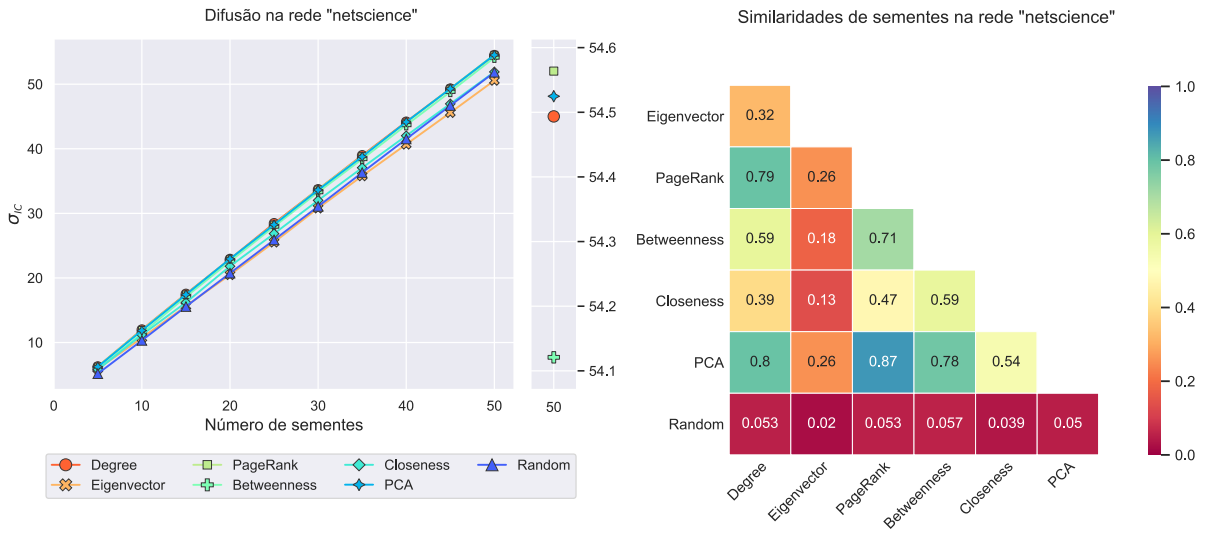

Figura 25 – Propagação de influência e correlações dos conjuntos de sementes selecionadas a partir das medidas de centralidade na rede “netscience”.

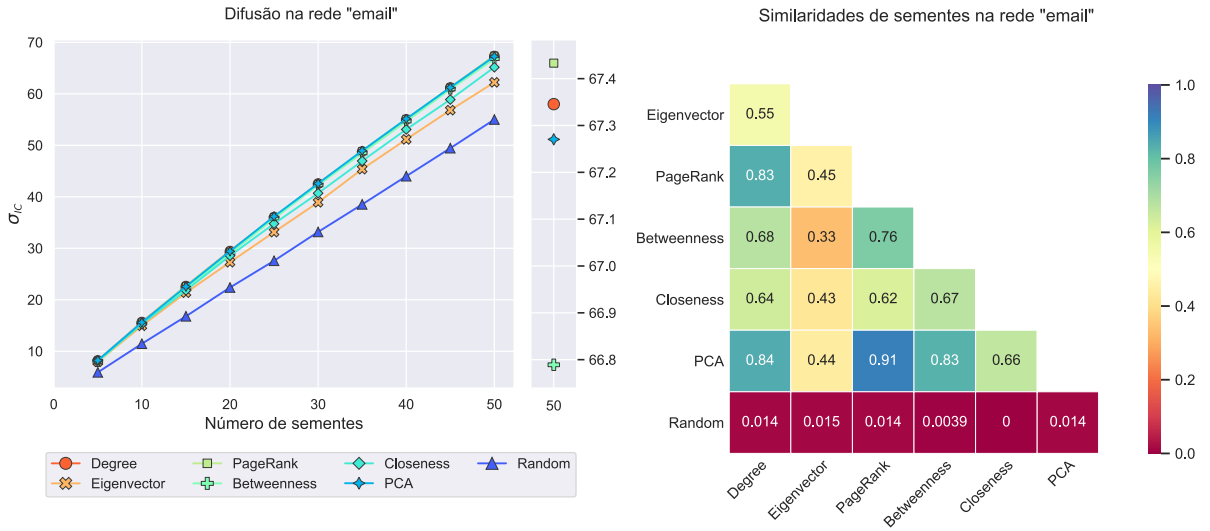

Figura 26 – Propagação de influência e correlações dos conjuntos de sementes selecionadas a partir das medidas de centralidade na rede “email”.

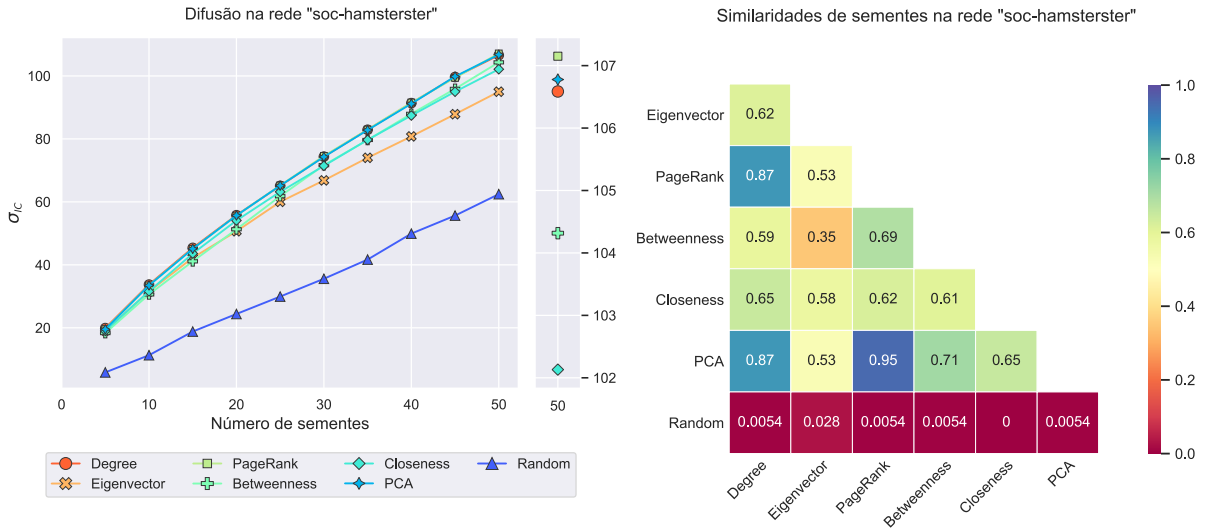

Figura 27 – Propagação de influência e correlações dos conjuntos de sementes selecionadas a partir das medidas de centralidade na rede “soc-hamsterster”.

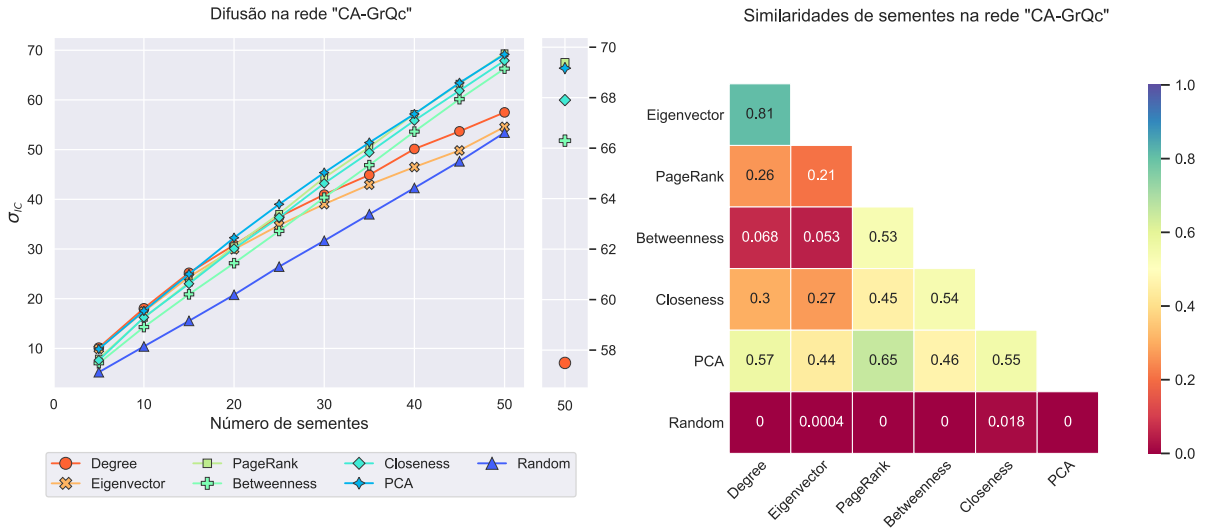

Figura 28 – Propagação de influência e correlações dos conjuntos de sementes selecionadas a partir das medidas de centralidade na rede “CA-GrQc”.

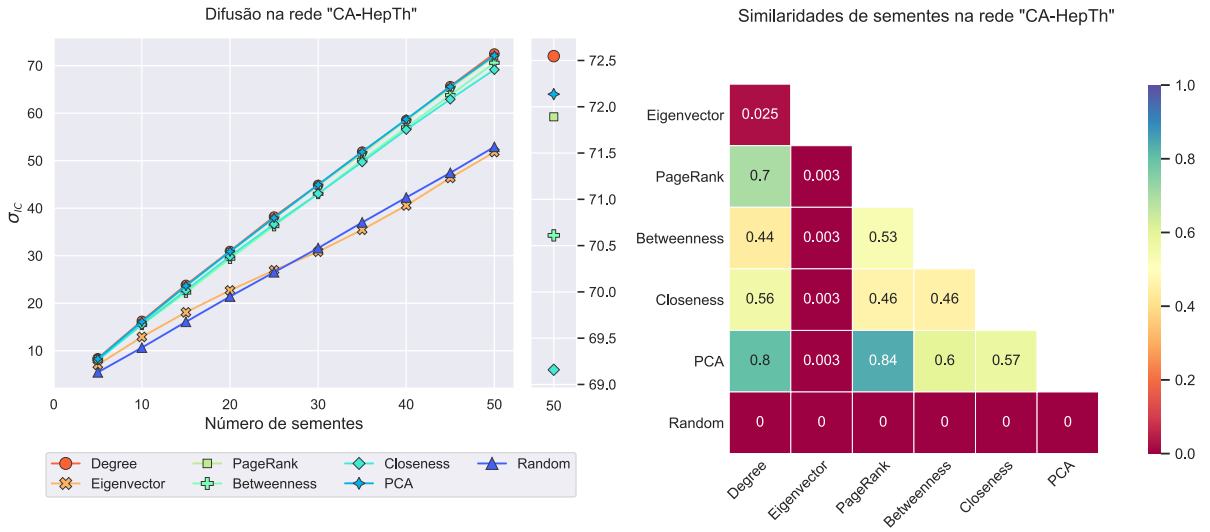

Figura 29 – Propagação de influência e correlações dos conjuntos de sementes selecionadas a partir das medidas de centralidade na rede “CA-HepTh”.

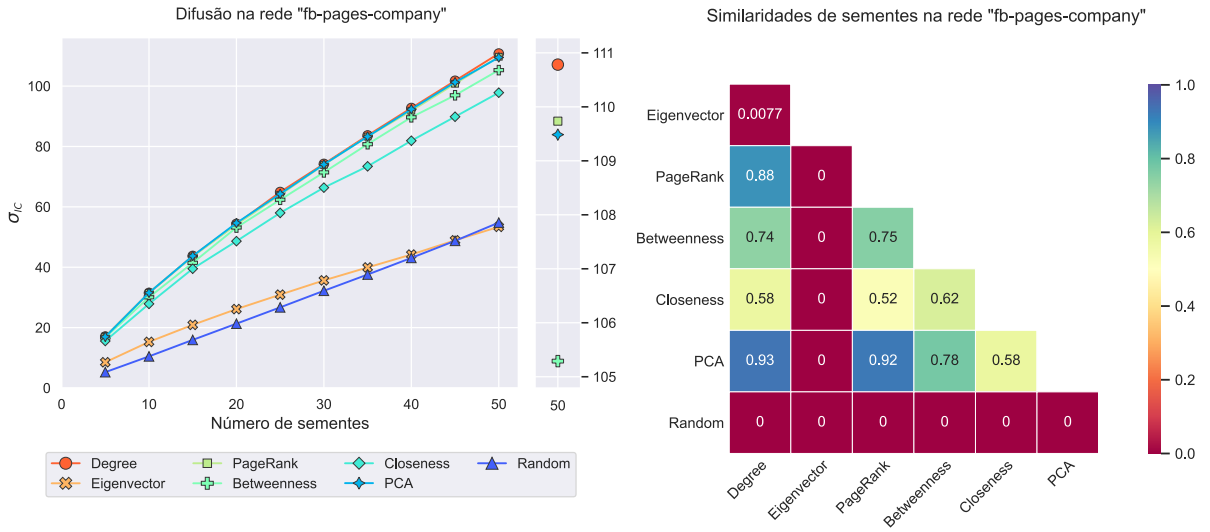

Figura 30 – Propagação de influência e correlações dos conjuntos de sementes selecionadas a partir das medidas de centralidade na rede “fb-pages-company”.

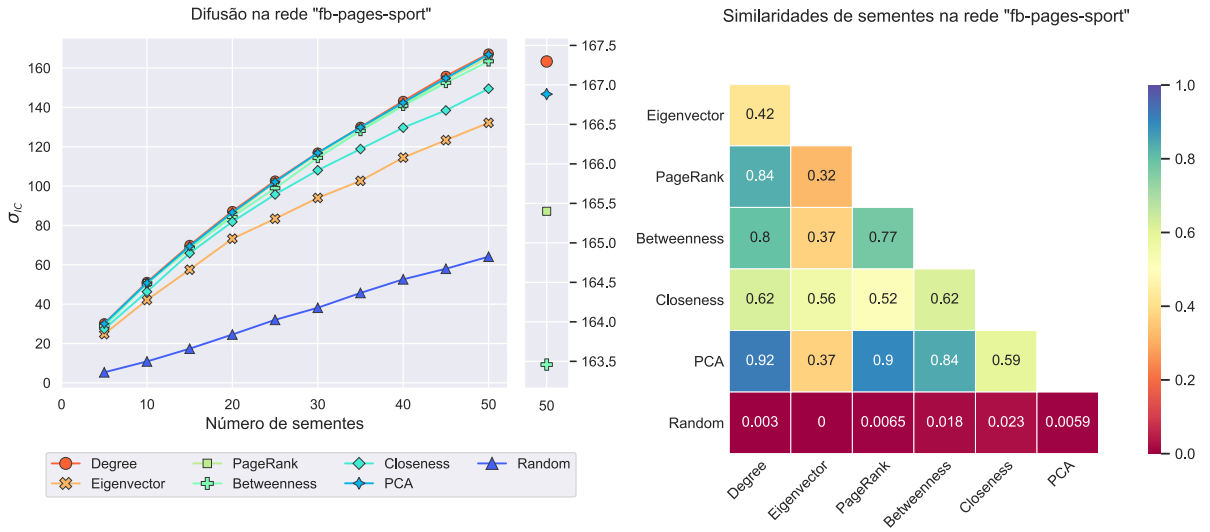

Figura 31 – Propagação de influência e correlações dos conjuntos de sementes selecionadas a partir das medidas de centralidade na rede “fb-pages-sport”.

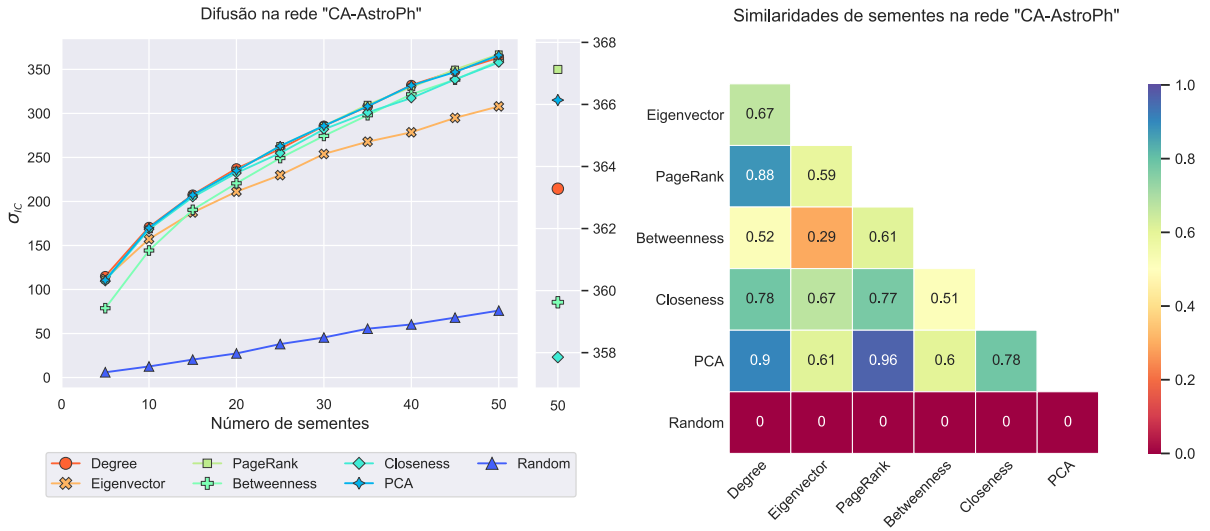

Figura 32 – Propagação de influência e correlações dos conjuntos de sementes selecionadas a partir das medidas de centralidade na rede “CA-AstroPh”.

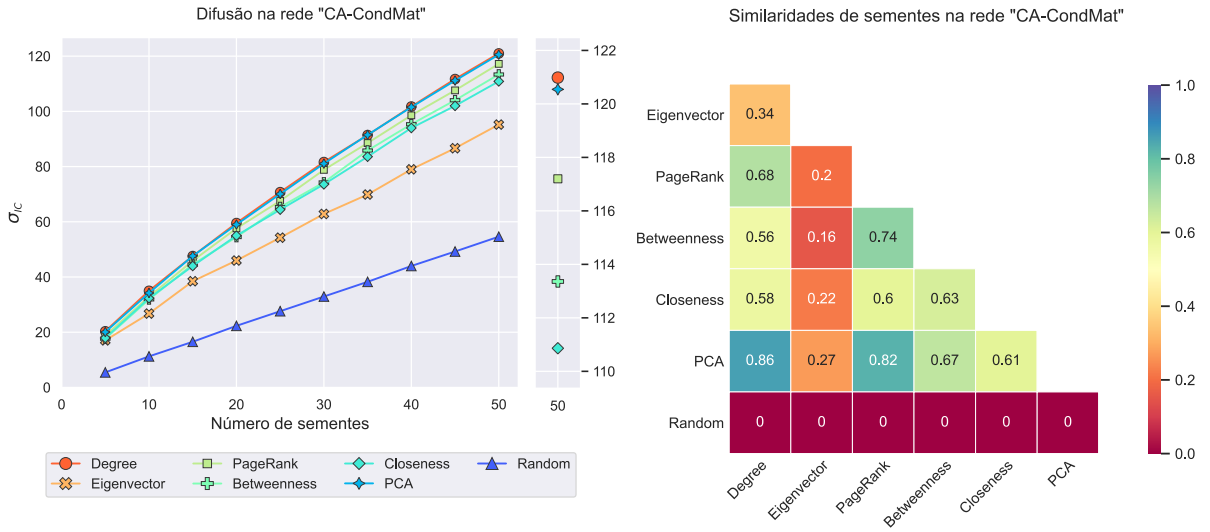

Figura 33 – Propagação de influência e correlações dos conjuntos de sementes selecionadas a partir das medidas de centralidade na rede “CA-CondMat”.

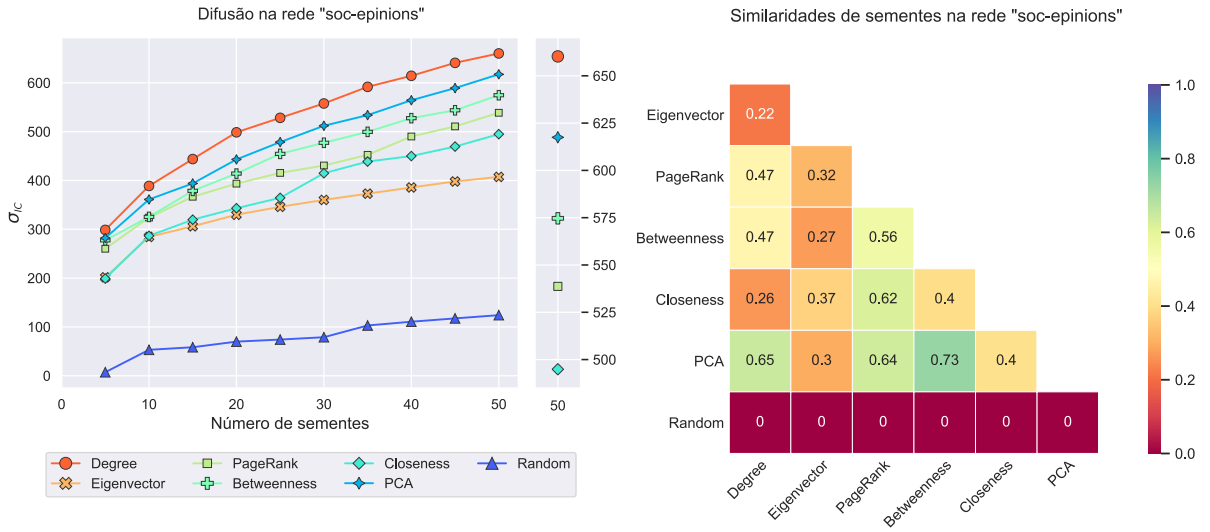

Figura 34 – Propagação de influência e correlações dos conjuntos de sementes selecionadas a partir das medidas de centralidade na rede “soc-epinions”.

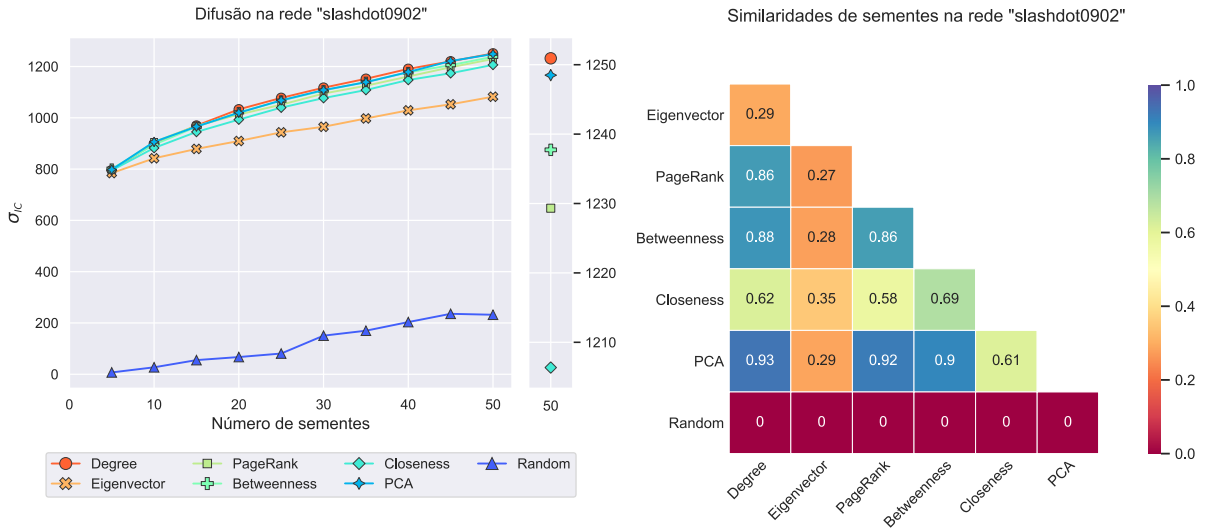

Figura 35 – Propagação de influência e correlações dos conjuntos de sementes selecionadas a partir das medidas de centralidade na rede “slashdot0902”.

APÊNDICE C – Propagação de influência e correlação entre conjuntos de sementes - heurísticas

Neste capítulo estão agrupados os resultados da propagação de influência e das correlações entre os conjuntos de sementes selecionadas pelas heurísticas.

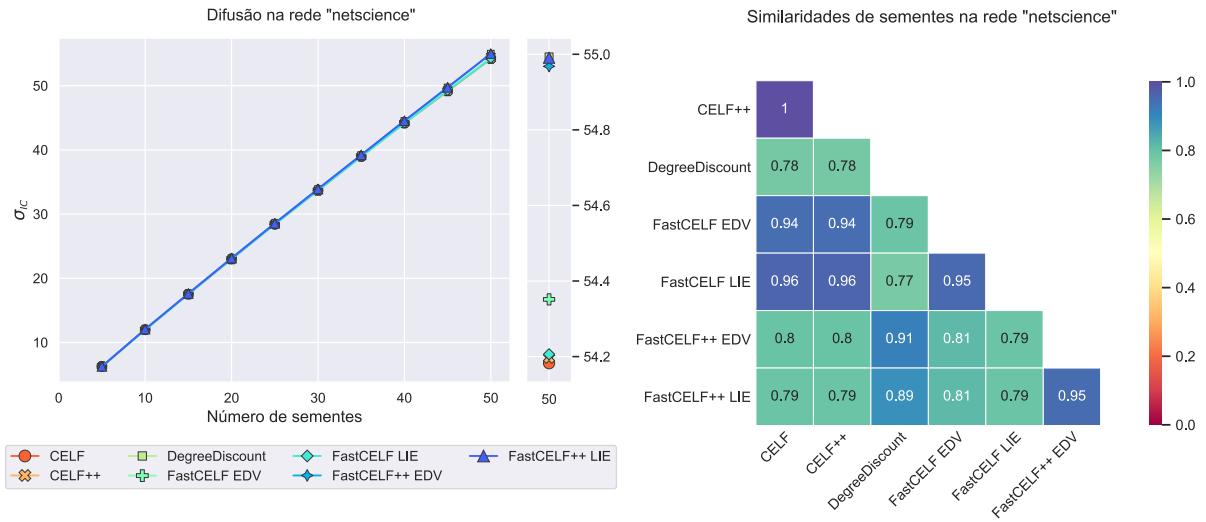

Figura 36 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “netscience”.

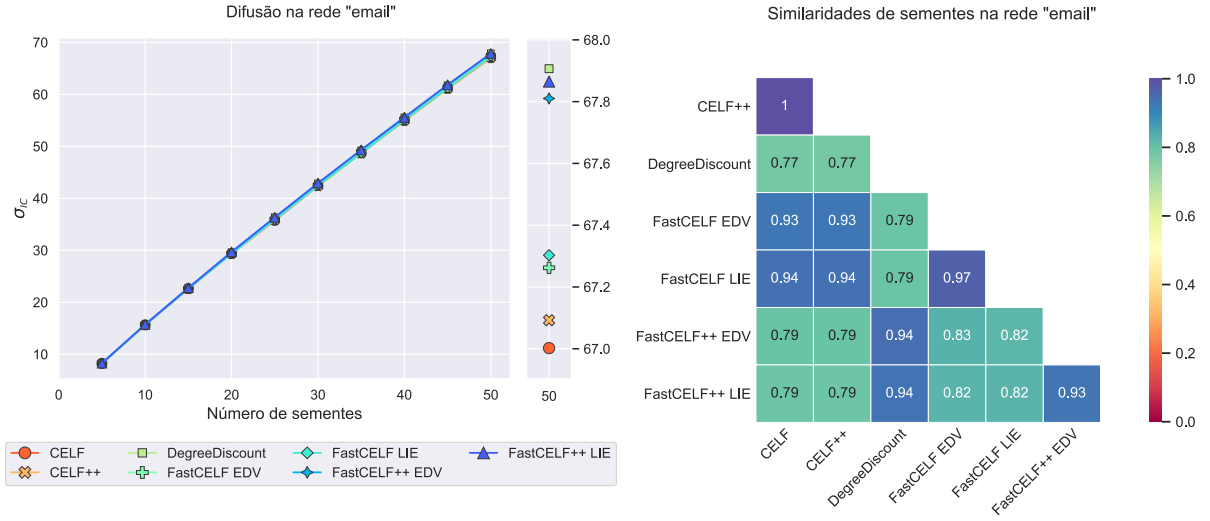

Figura 37 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “email”.

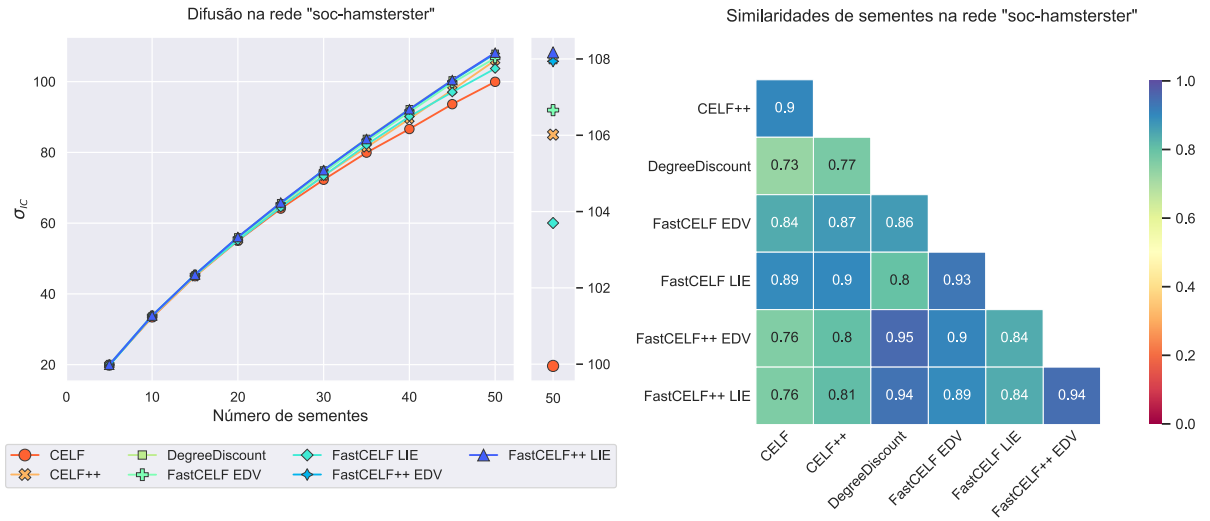

Figura 38 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “soc-hamsterster”.

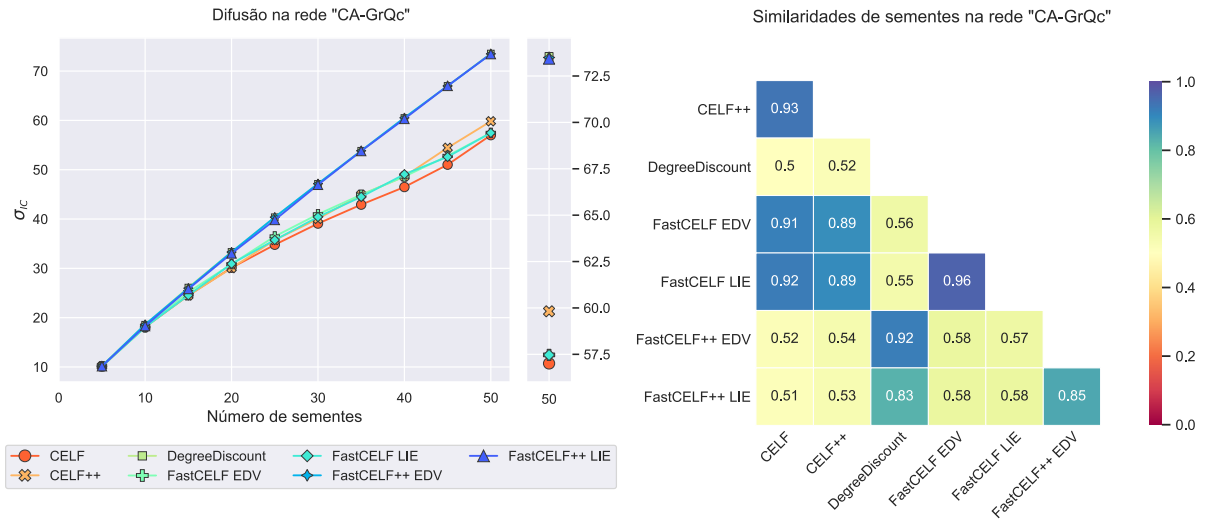

Figura 39 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “CA-GrQc”.

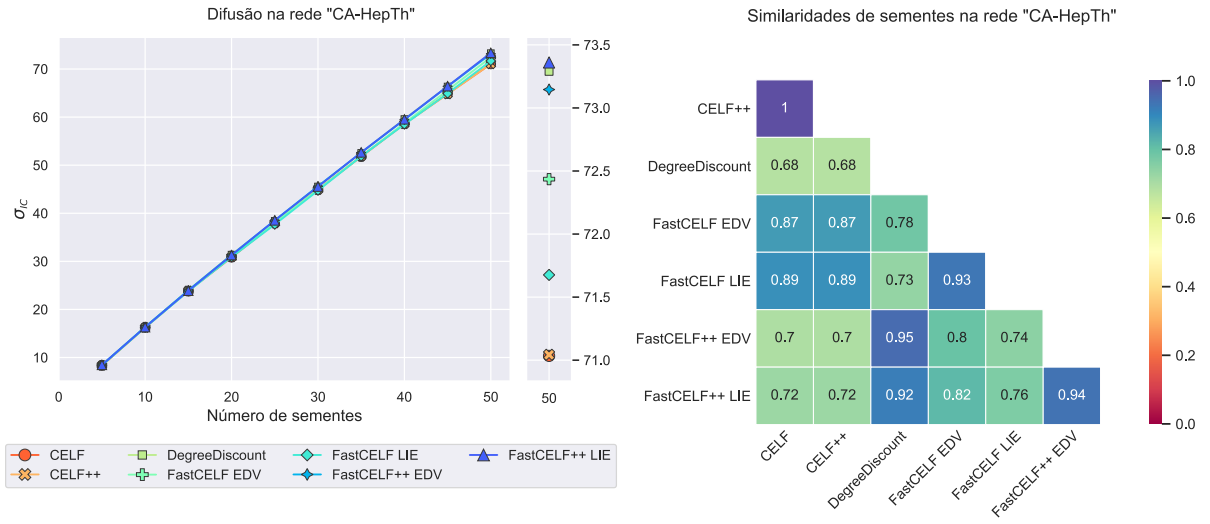

Figura 40 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “CA-HepTh”.

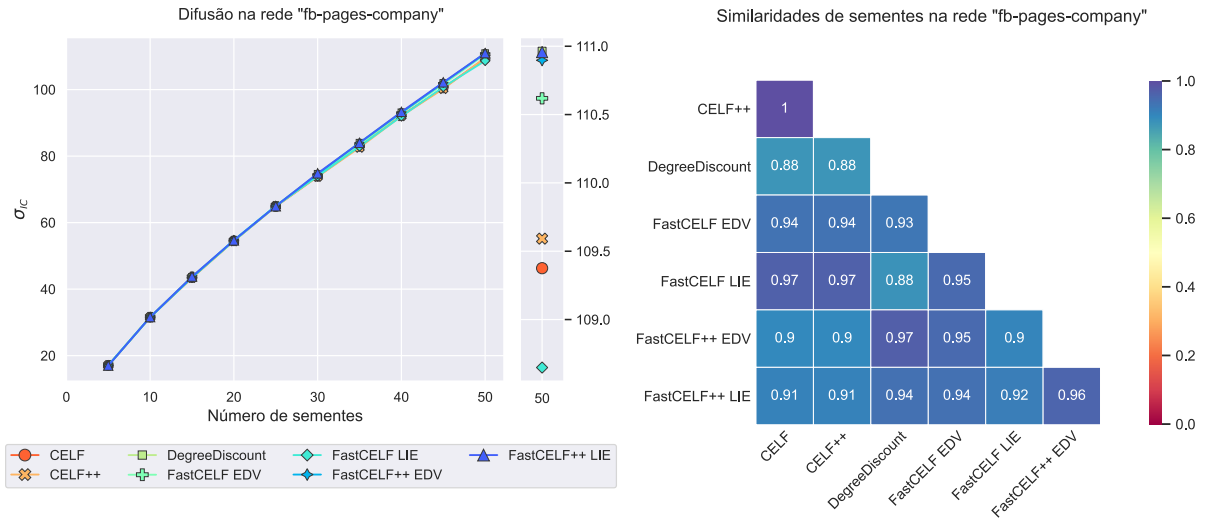

Figura 41 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “fb-pages-company”.

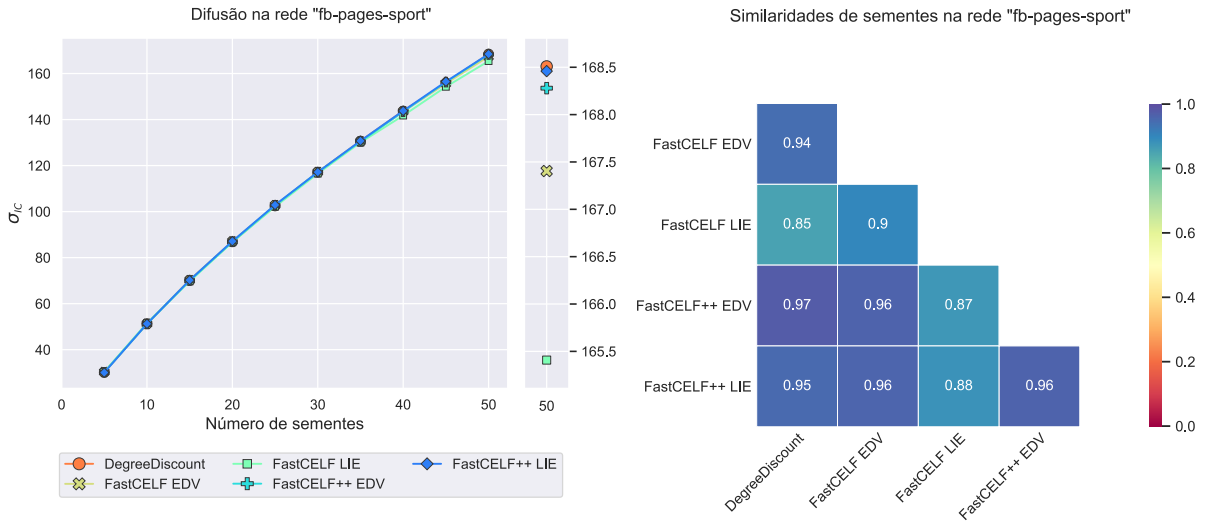

Figura 42 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “fb-pages-sport”.

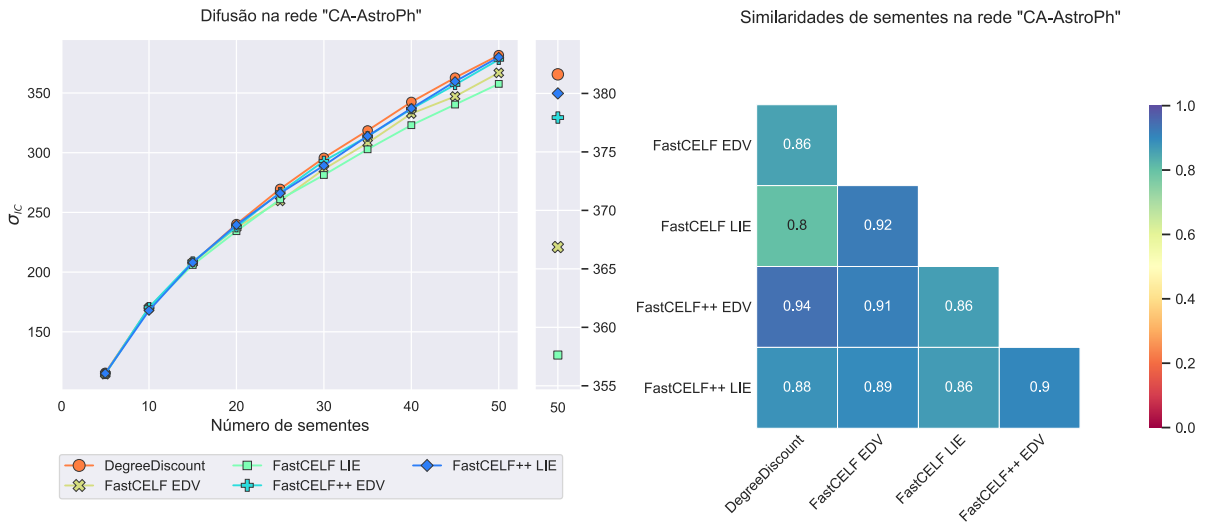

Figura 43 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “CA-AstroPh”.

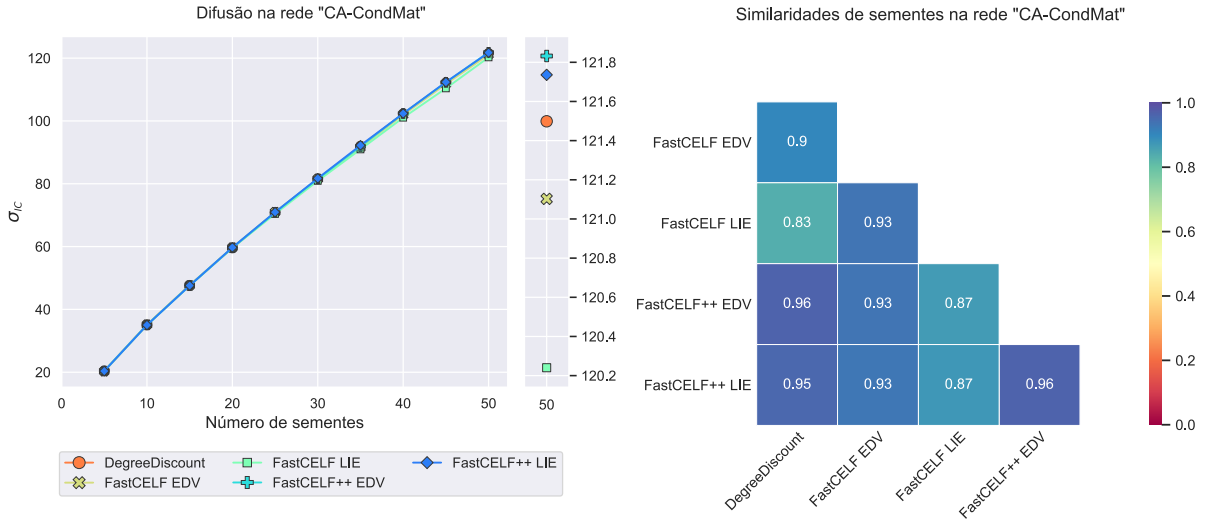

Figura 44 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “CA-CondMat”.

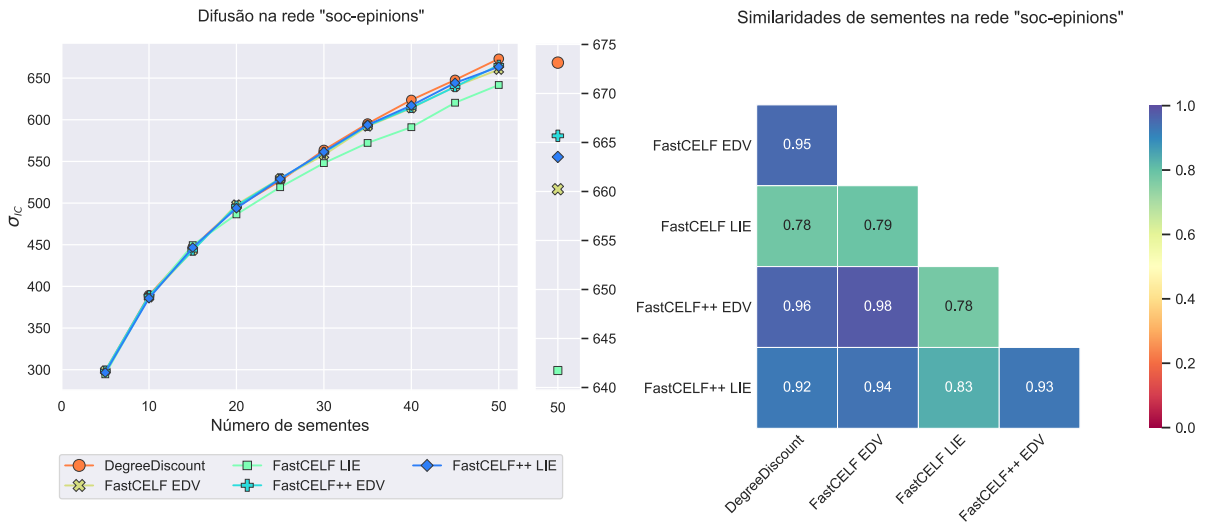

Figura 45 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “soc-epinions”.

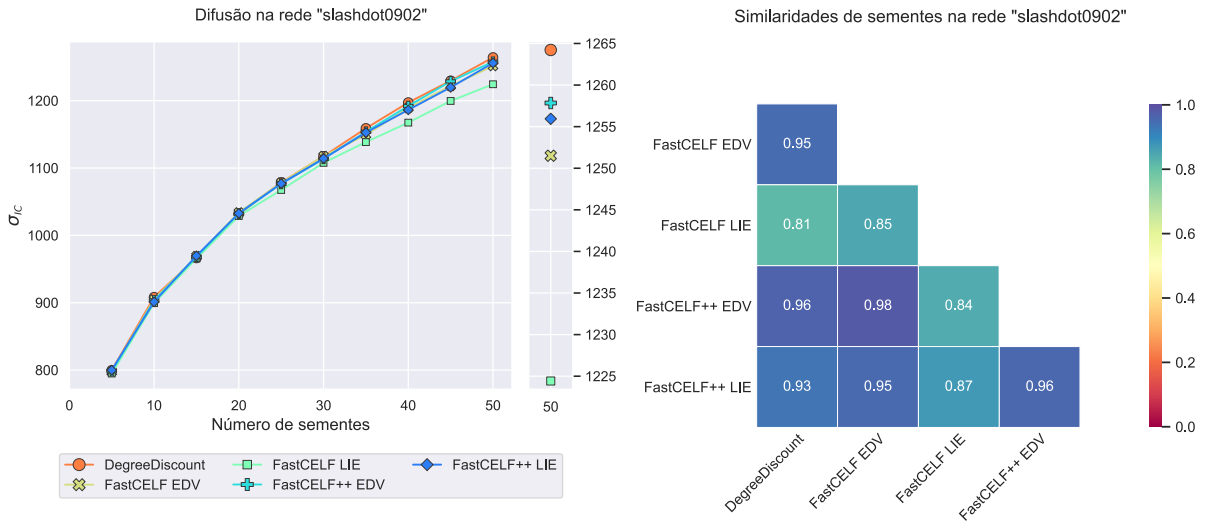

Figura 46 – Propagação de influência e correlações dos conjuntos de sementes selecionadas pelas heurísticas na rede “slashdot0902”.