



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA

Francisco Jucelino Matos Júnior

**Metaheurísticas para detecção de líderes e comunidade em redes sociais**

Recife

2021

Francisco Jucelino Matos Júnior

**Metaheurísticas para detecção de líderes e comunidade em redes sociais**

Trabalho apresentado ao Programa de Pós-graduação em Estatística do Centro de Ciências Exatas e da Natureza da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Doutor em Estatística.

**Área de Concentração:** Estatística Aplicada.

**Orientador (a):** Prof. Dr. Raydonal Ospina Martínez

**Coorientador (a):** Profa. Dra. Geiza Cristina da Silva

Recife

2021

Catálogo na fonte  
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

M433m Matos Júnior, Francisco Jucelino  
Metaheurísticas para detecção de líderes e comunidade em redes sociais /  
Francisco Jucelino Matos Júnior. – 2021.  
95 f.: il., fig., tab.

Orientador: Raydonal Ospina Martínez.  
Tese (Doutorado) – Universidade Federal de Pernambuco. CCEN,  
Estatística, Recife, 2021.  
Inclui referências.

1. Estatística aplicada. 2. Redes sociais. I. Ospina Martínez, Raydonal  
(orientador). II. Título.

310

CDD (23. ed.)

UFPE- CCEN 2021 - 40

**FRANCISCO JUCELINO MATOS JÚNIOR**

**METAHEURÍSTICAS PARA DETECÇÃO DE LÍDERES E COMUNIDADE EM  
REDES SOCIAIS**

Tese apresentada ao Programa de Pós-Graduação em Estatística da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Estatística.

Aprovada em: 11 de fevereiro de 2021.

**BANCA EXAMINADORA**

---

Prof.<sup>(a)</sup> Geiza Cristina da Silva  
UFPE

---

Prof. Leandro Chaves Rêgo  
UFPE

---

Prof. Pablo Martín Rodriguez  
UFPE

---

Prof. Luiz Satoru Ochi  
UFF

---

Prof. Vinicius da Fonseca Vieira  
UFSJ

Dedico esse trabalho aos meus pais, Meire e Juscelino, à minha irmã Caroline e vó Marina, à toda a comunidade LGBTQIA+, ao meu local de nascimento, Pirambu, e aos meus amigos e familiares.

## AGRADECIMENTOS

À minha vó, Marina, por me apoiar e me lembrar que não é do meu feito desistir dos meus sonhos, essa conquista é nossa e que diploma nenhum te dá educação como cidadão e como ser humano, isso a gente aprende com a família e muitas vezes com a vida.

Aos meus pais, Meire e Juscelino, por todo amor, carinho e dedicação. Por me amarem e permitem ganhar asas e viver um dos melhores momentos da minha vida. Essa é mais uma conquista nossa.

À minha irmã e cunhado, Carol e Ícaro, pelo apoio e amor incondicional.

À toda a minha família, sempre lembrando do orgulho que sentem por mim, em especial a Tia Socorro, Marina, Marília, Manoel e Liamara que sempre estiveram na linha de frente das pessoas mais orgulhosas pelo caminho que eu vinha traçando nos últimos 6 anos.

Aos meus colegas de pós-graduação Yuri, Vinícius, César e Bruna por todos os momentos incríveis desses últimos anos e por todo o consolo e auxílio que foram essencial para a chegada desse momento

À Lays, Júlio e João Pedro pelos melhores momentos da nossa vida juntos em Recife, barzinhos e “rolês” noturnos.

Ao Emídio, Gabriel, Janaína, José, João Kleber, Armando e Junior, por compartilharem diversos momentos comigo e por todo suporte durante esse momento, que pra mim, é o mais importante da minha vida até agora. Por serem uma família pra mim em Recife, assim como eu pude ser um pouco das suas. Por dividirem sonhos, tristezas e alegrias. Por cada conselho, por cada cerveja e por todas as vezes que você me abraçaram quando eu mais precisava de um carinho.

As minhas garotas de Fortaleza, Ianna Queiroz, Ítalo Nogueira, Mateus, Venícius Ferreira e Victor Temóteo, pelo amor, amizade, carinho, respeito e tudo que a nossa amizade trouxe nesses seis anos em que estive distante. Amo vocês.

Aos meus orientadores, Prof. Dr. Raydonal Ospina e Prof. Dr. Geiza Cristina, pela dedicação nesse trabalho. Por confiarem e acreditarem no meu potencial, muito mais do que eu acreditava, pela contribuição na minha vida pessoal e profissional e por todo carinho e amor me dado nesses anos em que esse trabalho foi produzido.

Ao Prof. Dr. João Maurício Mota pelo amor incondicional à minha pessoa e pela ajuda em todos esses seis anos, mesmo de tão distante. És uma das minha inspirações e peço descul-

pas pela distância estabelecida nos últimos meses. Muitas vezes, não queremos demonstrar fraqueza as pessoas que mais nos inspiram, porém sua solidariedade sempre foi um diferencial no meu caminho.

À todos os alunos do programa e amigos de Fortaleza, Recife, Olinda e da UFPE que de alguma forma ajudaram na minha chegada a este momento.

Ao CAPES, pelo apoio financeiro.

Ao membros da banca examinadora, por suas valiosas sugestões.

## RESUMO

Detecção de comunidades é um assunto que vem fascinando a comunidade científica pela necessidade de controlar o consumo e demanda de recursos e infraestruturas em sistemas complexos como são as redes sociais e a sua geração massiva de informações no atual paradigma do *Big Data*. Uma comunidade é definida como um conjunto de nós em uma rede que possuem propriedades similares. Em determinadas comunidades, alguns nós chamados de líderes de comunidade possuem uma grande capacidade para difundir informação e ideias. Muitos algoritmos de detecção de comunidades têm sido propostos na literatura. No entanto, poucos trabalhos têm abordado o problema de selecionar os líderes dessas comunidades. Nesse sentido, algoritmos como o *Leader Rank*, LCDA 1 e LCDA 2 foram desenvolvidos com esse propósito. Nesse trabalho propomos quatro novos algoritmos baseados na metaheurística GRASP e GRASP Reativo que utilizam adaptações do LCDA na fase de construção. Além destes algoritmos, desenvolvemos um método de avaliação da qualidade de detecção de comunidade e líderes de algoritmos através da geração de redes sintéticas que contém essas estruturas de comunidade e liderança. Os resultados experimentais mostram que, a qualidade da solução para a detecção de líderes e comunidades em redes sociais com os algoritmos propostos performam eficientemente e são tão competitivos como os relatados atualmente na literatura. Além disto, nosso gerador apresenta possibilidades de estruturas de grafos, trazendo uma representação de diversas topologias de grafos/redes, além de incluir e revelar como a estrutura de liderança influencia na estrutura de comunidade e vice-versa.

**Palavras-chaves:** Redes sociais. Detecção de Comunidades. Detecção de líderes em comunidades. *Benchmark*. GRASP.

## ABSTRACT

Detecting communities is a subject that has fascinated the scientific community due to the need to control the consumption and demand for resources and infrastructures in complex systems such as social networks and its massive generation of information in the current paradigm of Big Data. A community is defined as a set of nodes in a network that have similar properties. In certain communities, some nodes called community leaders have a great capacity to spread information and ideas. Many community detection algorithms have been proposed in the literature. However, few studies have addressed the problem of selecting the leaders of these communities. In this sense, algorithms such as Leader Rank, LCDA 1 and LCDA 2 were developed for this purpose. In this work we propose four new algorithms based on the metaheuristic GRASP and Reative GRASP that use adaptations of the LCDA in the construction phase. In addition to these algorithms, we have developed a method for assessing the quality of community detection and algorithm leaders through the generation of synthetic networks that contain these community and leadership structures. The experimental results show that the quality of the solution for the detection of leaders and communities in social networks with the proposed algorithms perform efficiently and are as competitive as those currently reported in the literature. In addition, our generator offers possibilities for graph structures, bringing a representation of various graph/network topologies, in addition to including and revealing the leadership structure in the community structure and vice versa.

**Keywords:** Social networks. Detection of Communities. Detection of community leaders. Benchmark. GRASP.

## LISTA DE FIGURAS

Figura 1 – Líderes e comunidades detectadas para a rede social <i>Zachary's Karate Club</i> .	51
Figura 2 – Líderes e comunidades detectadas para a rede social <i>American college football</i> .	51
Figura 3 – Líderes e comunidades detectadas para a rede social <i>Dolphin social network</i> .	52
Figura 4 – Líderes e comunidades detectadas para a rede social <i>Books about US politics</i> .	52
Figura 5 – Líderes e comunidades detectadas para a rede social <i>Political Blogs</i> .	53
Figura 6 – Grafos gerados através do LCGG para $n = 40$ e $K = 4$ e $n = 100$ e $K = 8$ , respectivamente.	79
Figura 7 – Grau médio obtido para as Instâncias com: (top) $n = 40$ e $K = 4$ e (bottom) $n = 100$ e $K = 8$ .	83
Figura 8 – Número de arestas obtido para as Instâncias com: (top) $n = 40$ e $K = 4$ e (bottom) $n = 100$ e $K = 8$ .	83
Figura 9 – Modularidades obtidas para as Instâncias com: (top) $n = 40$ e $K = 4$ e (bottom) $n = 100$ e $K = 8$ .	84

## LISTA DE TABELAS

Tabela 1 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Zachary's Karate Club</i> . . . . .	41
Tabela 2 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Zachary's Karate Club</i> . . . . .	41
Tabela 3 – Modularidade média obtida pelos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Zachary's Karate Club</i> . . . . .	42
Tabela 4 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>American College football</i> . . . . .	43
Tabela 5 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>American College football</i> . . . . .	43
Tabela 6 – Modularidade média obtida pelos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>American College football</i> . . . . .	44
Tabela 7 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Dolphin social network</i> . . . . .	45
Tabela 8 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Dolphin social network</i> . . . . .	45
Tabela 9 – Modularidade média obtida pelos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Dolphin social network</i> . . . . .	46
Tabela 10 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Books about US politics</i> . . . . .	47
Tabela 11 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Books about US politics</i> . . . . .	47
Tabela 12 – Modularidade média obtida pelos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Books about US politics</i> . . . . .	48
Tabela 13 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Political Blogs</i> . . . . .	49
Tabela 14 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Political Blogs</i> . . . . .	49

Tabela 15 – Modularidade média obtida pelos algoritmos propostos para cada combinação dos parâmetros $\alpha_c$ e $\alpha_s$ para a rede <i>Political Blogs</i> . . . . .	50
Tabela 16 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Zachary's Karate Club</i> . . . . .	54
Tabela 17 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Zachary's Karate Club</i> . . . . .	54
Tabela 18 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Zachary's Karate Club</i> . . . . .	55
Tabela 19 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>American College football</i> . . . . .	56
Tabela 20 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>American College football</i> . . . . .	56
Tabela 21 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>American College football</i> . . . . .	56
Tabela 22 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Dolphin social network</i> . . . . .	58
Tabela 23 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Dolphin social network</i> . . . . .	58
Tabela 24 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Dolphin social network</i> . . . . .	58
Tabela 25 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Books about US politics</i> . . . . .	59

Tabela 26 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Books about US politics</i> . . . . .	59
Tabela 27 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Books about US politics</i> . . . . .	60
Tabela 28 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Political Blogs</i> . . . . .	61
Tabela 29 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Political Blogs</i> . . . . .	61
Tabela 30 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Political Blogs</i> . . . . .	62
Tabela 31 – Comparação entre modularidades dos algoritmos propostos e da literatura. . . . .	62
Tabela 32 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Zachary's Karate Club</i> . . . . .	64
Tabela 33 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Zachary's Karate Club</i> . . . . .	64
Tabela 34 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Zachary's Karate Club</i> . . . . .	65
Tabela 35 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>American College football</i> . . . . .	66
Tabela 36 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>American College football</i> . . . . .	66
Tabela 37 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>American College football</i> . . . . .	66

Tabela 38 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Dolphin social network</i> . . . . .	68
Tabela 39 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Dolphin social network</i> . . . . .	68
Tabela 40 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Dolphin social network</i> . . . . .	68
Tabela 41 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Books about US politics</i> . . . . .	69
Tabela 42 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Books about US politics</i> . . . . .	69
Tabela 43 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Books about US politics</i> . . . . .	70
Tabela 44 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Political Blogs</i> . . . . .	71
Tabela 45 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Political Blogs</i> . . . . .	71
Tabela 46 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede <i>Political Blogs</i> . . . . .	72
Tabela 47 – Comparação entre modularidades dos algoritmos propostos e da literatura. . . . .	73
Tabela 48 – Tipos de instâncias a partir das combinações dos parâmetros $\alpha$ , $p_{leader}$ , $p_{follower}$ e $p_{prototypical\_leader}$ . . . . .	78
Tabela 49 – Resultados do estudo de simulação para as Instâncias com $n = 40$ e $K = 4$ . . . . .	81
Tabela 50 – Resultados do estudo de simulação para as Instâncias com $n = 100$ e $K = 8$ . . . . .	82
Tabela 51 – $EQM$ obtidos pelos algoritmos LCDA 1 e 2 das instâncias com $n = 40$ e $k = 4$ . . . . .	86
Tabela 52 – $EQM$ obtidos pelos algoritmos LCDA 1 e 2 das instâncias com $n = 100$ e $k = 8$ . . . . .	87
Tabela 53 – $PMA$ obtidos pelos algoritmos LCDA 1 e 2 das instâncias com $n = 40$ e $k = 4$ . . . . .	88

Tabela 54 – *PMA* obtidos pelos algoritmos LCDA 1 e 2 das instâncias com  $n = 100$  e  $k = 8$  89

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
1.1	OBJETIVOS	17
1.2	ORGANIZAÇÃO DO TRABALHO	18
<b>2</b>	<b>REFERÊNCIA TEÓRICO</b>	<b>19</b>
2.1	INTRODUÇÃO	19
2.2	ALGORITMOS LCDA	22
2.3	MÉTRICAS UTILIZADAS	23
2.4	METAHEURÍSTICA GRASP	26
2.5	METAHEURÍSTICA GRASP REATIVO	29
2.6	INSTÂNCIAS DO PROBLEMA	32
<b>3</b>	<b>ALGORITMOS PROPOSTOS</b>	<b>34</b>
3.1	ALGORITMOS LCDA-GRASP	34
3.2	ALGORITMOS LCDA-GR	37
3.3	RESULTADOS COMPUTACIONAIS	39
<b>3.3.1</b>	<b>LCDA-GRASP</b>	<b>39</b>
3.3.1.1	<i>Estudo de Simulação 1</i>	40
3.3.1.2	<i>Estudo de Simulação 2</i>	53
<b>3.3.2</b>	<b>LCDA-GR</b>	<b>63</b>
3.3.2.1	<i>Estudo de Simulação</i>	63
<b>4</b>	<b>BENCHMARK EM ALGORITMOS QUE DETECTAM LÍDERES E COMUNI- DADES</b>	<b>74</b>
4.1	GERADOR DE GRAFOS COM LIDERANÇA-COMUNIDADE (LCGG)	75
4.2	TESTES E RESULTADOS COMPUTACIONAIS	77
<b>4.2.1</b>	<b><i>Estudo de Simulação 1: Características dos Grafos Gerados</i></b>	<b>79</b>
<b>4.2.2</b>	<b><i>Estudo de Simulação 2: Comparação de desempenho dos algoritmos LCDA 1 e 2, LCDA-GRASP 1 e 2 e LCDA-GR 1 e 2</i></b>	<b>84</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>90</b>
	<b>REFERÊNCIAS</b>	<b>92</b>

## 1 INTRODUÇÃO

Desde a década de 90, a origem da Internet e os avanços promovidos pelas tecnologias da informação e comunicação causaram mudanças significativas na sociedade, principalmente na maneira como as pessoas interagem. Entre essas tecnologias, surgem as Redes Sociais, tornando-se um fenômeno global influenciando profundamente o dia-a-dia social das pessoas. As mídias sociais deram espaço para que usuários gerassem e compartilhassem conteúdo, deixando de lado o comportamento passivo de absorção da informação e caracterizando uma maneira de democratização na geração de conteúdo.

Nesse contexto, atualmente lida-se com o problema de investigar estruturas com muitas informações (propriedades), conjuntos de dados heterogêneos (como por exemplo, Redes Sociais). A Rede Social é uma estrutura que inter-relaciona empresas ou pessoas, por exemplo, que estão conectadas pelas mais diversas relações. Cada qual se relaciona de acordo com as suas preferências e particularidades. Trata-se de uma ligação social e conexão entre pessoas. Tais conjuntos de dados são melhores representados por redes ou grafos (GETOOR; DIEHL, 2005). Basicamente, uma comunidade é um conjunto de usuários que interagem uns com os outros com frequência (WASSERMAN; FAUST, 1994). De acordo com Back (1965), as interações sociais diádicas e os grupos formam a base para a instanciação social de qualquer indivíduo.

Uma das características mais tradicionais nas mídias sociais é a aglomeração de usuários em comunidades. Uma grande quantidade de agrupamentos vem surgindo nas Redes Sociais, esses têm como peculiaridades, maiores interações entre os usuários aglomerados e elevada produção de conteúdos de interesses comuns. Esse fenômeno é conhecido como o processo de formação de comunidades (BAUMES et al., 2004; SUN et al., 2007; TANG; LIU, 2010). De modo geral, as comunidades caracterizam-se pelo conjunto de seus participantes, os quais compartilham valores, interesses, metas e posturas de apoio mútuo. As comunidades podem ser identificadas de forma explícita, por exemplo, quando são formadas por assinaturas de usuários; ou implicitamente, a partir da análise dos padrões de relacionamento dos usuários (topologia da rede). No caso implícito, as comunidades podem ser automaticamente detectadas por técnicas adequadas de agrupamento gráfico (XIE; KELLEY; SZYMANSKI, 2013), algoritmos de detecção de comunidades. Grande parte dos trabalhos científicos que se dedicam ao estudo de comunidades em redes sociais têm sido sobre o processo de detecção de comunidades.

Alguns nós da rede associados a determinadas comunidades têm grande capacidade de

difusão de informações e ideias e por isso é de grande importância a sua identificação. Por exemplo, em situações em que há uma propagação epidêmica, encontrar os nós mais importantes nesse sistema ajudaria a entender e controlar a dinâmica do processo (FU; HUANG; SUN, 2015). Esses nós importantes são denotados por *líderes de comunidade*. O processo de identificação de nós líderes pode ser visto como um ranqueamento dos nós. Uma variedade de medidas de centralidade têm sido usadas para medir a capacidade de disseminação de informações de usuários em redes sociais de acordo com diferentes focos, algumas delas são centralidade de grau, centralidade de intermediação, centralidade de proximidade, centralidade de autovetor (XIA et al., 2016).

Na literatura é encontrado que de forma geral, os algoritmos são utilizados para detectar comunidades e, líderes simultaneamente. Em Ahajjam, Haddad e Badir (2018) são propostos os algoritmos *Leader-Community Detection Algorithm - LCDA 1 e 2*. Os algoritmos atuam em dois passos: o primeiro identifica os nós líderes e o segundo, detecta as comunidades usando métricas de similaridades entre os nós. Detectar prioritariamente os nós líderes e utilizá-los na formação das comunidades sem que seja necessário o conhecimento do número de comunidades da rede a priori é a abordagem inédita nesse trabalho. Contudo, quando se trata da qualidade na detecção da comunidade esses métodos não garantem a qualidade da solução. Em Atay et al. (2017) são propostos diversos algoritmos baseados em metaheurísticas para detecção de comunidades em redes sociais. Os resultados mostram que a qualidade na detecção das comunidades, medida pela modularidade, é maior do que a alcançada pelos algoritmos LCDA 1 e LCDA 2.

## 1.1 OBJETIVOS

Neste trabalho os algoritmos LCDA 1 e 2 são adaptados para a fase de construção de soluções iniciais e um novo algoritmo de busca local é desenvolvido, gerando assim dois novos algoritmos GRASP e GRASP Reativo, cada, para a detecção de líderes de comunidades em redes. Uma extensa bateria de testes é realizada em cinco das redes sociais mais utilizadas na literatura: *Zachary's karate club*, *American College football*, *Dolphin social network*, *Books about US politics* e *Political Blogs*. Os resultados computacionais demonstram que os algoritmos propostos são promissores para obter soluções de alta qualidade para o problema abordado.

Além disto, um gerador de grafo sintéticos foi desenvolvido, estes possuindo uma estrutura de comunidade e liderança e a fim de utilizá-lo como um método de avaliação de algoritmos que

detectam tanto comunidades como líderes. A partir desse estudo, pode-se ser capaz de avaliar a influência que essas estruturas de comunidade e liderança possuem em suas detecções, assim como avaliar os algoritmos em relação a diferentes topologias de grafos.

## 1.2 ORGANIZAÇÃO DO TRABALHO

O capítulo 2 apresenta uma descrição dos algoritmos LCDA 1 e 2, das principais métricas de qualidade, centralidade e similaridade utilizadas, além das instâncias estudadas no decorrer deste trabalho. No capítulo 3 são introduzidos os algoritmos LCDA-GRASP 1 e 2, desenvolvidos a partir da metaheurística GRASP utilizando os algoritmos LCDA 1 e 2, respectivamente, na fase de construção das soluções iniciais, além dos algoritmos LCDA-GR 1 e 2, desenvolvidos a partir da metaheurística GRASP Reativo também utilizando os algoritmos LCDA 1 e 2, respectivamente. Ao final do capítulo, testes realizados para avaliar suas performance para diferentes grafos reais. No capítulo 4 contém o benchmark desenvolvido, assim como os estudos de simulação realizados para avaliar os diferentes tipos de características e topologias, assim como estruturas de comunidade e liderança, que o gerador é capaz de produzir. Finalmente, no capítulo 5 são discutidas as conclusões e considerações finais do trabalho.

## 2 REFERÊNCIA TEÓRICO

### 2.1 INTRODUÇÃO

Sistemas complexos, tais como redes sociais, são comumente representados por redes complexas. As redes complexas possuem um caráter multidisciplinar e cobrem aplicações em diversas áreas do conhecimento, utilizando a teoria dos grafos como ferramenta para modelagem, tratamento e análise de dados. Redes geralmente são modeladas por meio de grafos, uma vez que esses possuem muitas características que podem ser associadas a diferentes propriedades estruturais de uma rede. Neste trabalho, uma rede complexa é representada por um grafo não direcionado e não ponderado  $G(V, E)$ , em que  $V$  é o conjunto de nós e  $E$  é o conjunto de arestas. Cada um dos nós contidos em  $G$  representam elementos ou indivíduos do conjunto de dados, em que  $|V| = n$  representa o número de nós. Cada aresta representa um relacionamento ou interação entre par de nós, em que  $|E|$  representa o número de arestas. Essas relações constituem a estrutura de rede do grafo que é representada por uma matriz de adjacência  $A = \{a_{ij}\}$ , com  $a_{ij} \in \mathbb{R}$ , onde, neste trabalho, é considerado que  $a_{ij} = 1$ , se existir uma relação entre os nós  $i$  e  $j$ , caso contrário,  $a_{ij} = 0$ .

Uma propriedade bastante comum em redes complexas é a presença de estruturas modulares chamadas de comunidades. As comunidades podem ser definidas como um conjunto de vértices que possuem propriedades comuns ou desempenham funções similares dentro da rede (ALVES; OLIVEIRA; BRITO, 2014). Problemas de agrupamento uns dos mais explorados na literatura, consiste na identificação de grupos em um grafo de forma que, todo nó seja associado a um único grupo e em cada um deles, os nós sejam mais semelhantes entre si e que a similaridade entre grupos distintos seja a menor possível. A detecção de comunidades pode ser considerada um problema de clusterização em que a rede modela as interações (comportamento social) de um grupo.

Alguns nós da rede associados a determinadas comunidades têm grande capacidade de difusão de informações e ideias e por isso é de grande importância a sua identificação. Por exemplo, em situações em que há uma propagação epidêmica, encontrar os nós mais importantes nesse sistema ajudaria a entender e controlar a dinâmica do processo (FU; HUANG; SUN, 2015). Esses nós importantes são denotados por *líderes de comunidade*. O processo de identificação de nós líderes pode ser visto como um ranqueamento dos nós. Uma variedade de medidas de centralidade têm sido usadas para medir a capacidade de disseminação de informações de usuários

em redes sociais de acordo com diferentes focos, algumas delas são centralidade de grau, centralidade de intermediação, centralidade de proximidade e centralidade de autovetor (XIA et al., 2016).

Neste trabalho os algoritmos LCDA 1 e 2 são adaptados e um novo algoritmo de busca local é desenvolvido, gerando assim dois novos algoritmos GRASP, chamados de LCDA-GRASP 1 e 2, para a detecção de líderes e de comunidades em redes. Uma extensa bateria de testes é realizada em quatro das redes sociais mais utilizadas na literatura: *Zachary's Karate Club*, *American College Football*, *Dolphin social network* e *Books about US politics*. Os resultados computacionais demonstram que os algoritmos propostos são promissores pois são capazes de detectar líderes e comunidades de maneira eficiente.

Uma das principais formas de entender como ocorre o fluxo de informações dentro de uma rede é a detecção de líderes e comunidades. Enquanto que, a detecção de comunidades revela como os indivíduos nessas comunidades se comportam e como essas comunidades se relacionam entre si, a detecção de líderes identifica os nós mais influentes, ou seja, os indivíduos com alta capacidade de disseminação das informações dentro da rede, baseada em uma métrica importância. Em resumo, a detecção de líderes e comunidades revelam como esses indivíduos se comportam dentro e fora de suas comunidades, além de revelar como ocorre a disseminação de informações da rede e a dinâmica dos líderes e suas comunidades.

Os métodos de análise de influência em redes sociais e, posteriormente, a identificação de seus líderes podem ser divididas em duas abordagens (AHAJJAM; HADDAD; BADIR, 2018):

- Global: utilizando métricas que dão ênfase ao estudo da topologia da rede, como por exemplo a centralidade de intermediação e proximidade;
- Local: o foco está em regiões locais ou em nós individuais, como por exemplo a centralidade de grau.

A identificação de comunidades também possui duas abordagens metodológicas. A primeira é o particionamento do grafo, em que o objetivo é subdividir o grafo em  $k$  sub-grafos ou grupos de tal forma que, os nós contidos em cada grupo são fortemente ou densamente conectados entre si. A segunda abordagem é a classificação, que se diferencia do particionamento ao fazer uso de medidas de similaridade de partições (FORTUNATO, 2010; YANG; MCAULEY; LESKOVEC, 2013; WU et al., 2013). Porém, ambas as abordagens possuem a desvantagem de se possuir um conhecimento prévio do número de grupos a serem detectados (NEWMAN, 2013; BADER et al.,

---

2013; FORTUNATO, 2010). Segundo Yang, McAuley e Leskovec (2013) esse problema, para os métodos de classificação, basta selecionar uma métrica que melhore a classificação dos dados.

Nos últimos anos, diversos métodos propõem detectar conjuntamente comunidades e líderes, dentre eles pode-se destacar o algoritmo do tipo *k-means*. Proposto inicialmente por Khorasgani, Chen e Zaïane (2010), o algoritmo detecta aleatoriamente  $k$  nós e, em seguida, os demais nós são aglomerados aos seus líderes mais próximos. O trabalho desenvolvido por Kernighan e Lin (1970), baseado no particionamento de grafos, realiza uma busca entre as partições a fim de minimizar o número de arestas entre elas a partir da troca de vértices entre partições.

O algoritmo *Leader-Follower* desenvolvido por Wu et al. (2013), realiza agrupamentos para detecção de líderes e comunidades tomando como base medidas de centralidade, onde o vértice com maior centralidade seria escolhido como ponto inicial do algoritmo. No algoritmo, cada comunidade é constituída de um nó líder e um nó *follower* e a restrição de que um membro de uma comunidade só deve ter conexões com membros de sua comunidade. A técnica é realizada em três etapas: **i)** realiza-se o agrupamento dos vértices, **ii)** grupos que compartilham uma alta porcentagem de conexões em comum são unidos e **iii)** ao fim, os vértices de cada grupo constituem um vértice novo que representa seu grupo. Desta forma, o algoritmo produz comunidades com poucos membros, com dois ou um nó(s).

Conforme discutem Ahajjam, Haddad e Badir (2018), as técnicas utilizadas em ciências da computação para a análise de influência dos usuários em redes sociais e detecção de seus líderes podem ser divididas em duas categorias: análises baseadas em comportamento global, que dão ênfase ao estudo da topologia da rede (por exemplo, centralidade de intermediação) e métodos locais, que focam em regiões locais ou nos nós individuais (por exemplo, centralidade de grau).

Trabalhos como os de Khorasgani, Chen e Zaïane (2010), Wang, Di e Fan (2011) usam medidas de centralidade globais e locais para identificar efetivamente os líderes em grandes redes sociais. Ahajjam, Haddad e Badir (2015) desenvolveram o algoritmo *LeaderRank* que detecta comunidades a partir dos líderes identificados, utilizando medidas de vizinhança. Como continuidade do trabalho e extensão do algoritmo *LeaderRank*, Ahajjam, Haddad e Badir (2018) propôs os algoritmos LCDA 1 e 2. Nestes, medidas de similaridade são utilizadas para aglomerar nós em uma comunidade. Uma descrição mais completa dos métodos pode ser encontrada na seção 2.2.

## 2.2 ALGORITMOS LCDA

Segundo Xia et al. (2016), medidas de centralidade (WASSERMAN; FAUST, 1994) podem ser bastante úteis na análise de redes sociais, uma vez que, essas medidas determinam a importância relativa de um nó dentro do grafo. Neste contexto Ahajjam, Haddad e Badir (2018) propõem dois algoritmos, *leader-community detection algorithm* (LCDA) 1 e 2, para detecção de líderes e comunidades em redes complexas, sem o conhecimento *a priori* do número de comunidades. Sua abordagem se divide em duas etapas:

- **Etapa 1: Detecção de Líderes:** Baseando-se em estudos sociométricos, Ahajjam, Haddad e Badir (2018) afirmam que um “verdadeiro líder” é aquele escolhido por indivíduos igualmente poderosos, enquanto líderes “populares” são aqueles mais propensos a serem escolhidos por indivíduos menos poderosos. Para contornar este problema, os autores sugerem o uso de alguma medida de centralidade global, que leva em consideração a importância do nó na topologia do grafo, para a detecção de líderes em uma rede. Na construção dos algoritmos LCDA 1 e 2, Ahajjam, Haddad e Badir (2018) adotam a medida de centralidade de autovetor, definida pela Equação (3). Os nós são, então, classificados segundo suas medidas de centralidade, e o primeiro líder  $L_1$  é definido como o nó com maior medida de centralidade, sendo então o líder  $L_1$  o nó com maior autovalor na matriz de adjacência  $A$ .

- **Etapa 2: Detecção de Comunidade:**

Para cada líder, uma função de similaridade é calculada a fim de encontrar os vizinhos do nó líder  $L_1$ . Os nós mais similares serão atribuídos ao nó líder detectado e, assim por diante, até formar uma comunidade, tomando o limiar  $k$  como ponto de corte, em que Ahajjam, Haddad e Badir (2018) utilizaram  $k \geq 0,5$ ). Como é suposto que o grafo é não direcionado e não ponderado, Ahajjam, Haddad e Badir (2018) aconselham o uso de similaridades de borda e utilizam a similaridade HPI, definida na Equação (6), em seus resultados.

As etapas 1 e 2 são repetidas até que todos os nós sejam atribuídos a uma comunidade. A principal diferença entre os algoritmos LCDA 1 e LCDA 2 é que, no segundo algoritmo, a cada nova detecção de um líder, recalcula-se as medidas de centralidade dos nós, mas apenas com os nós que ainda não foram classificados.

Os Algoritmos 1 e 2 apresenta o pseudo-código do algoritmo LCDA 1 e 2, respectivamente.

**Entrada:** Grafo não direcionado  $G$

**Saída:**  $C = (C_1, \dots, C_g)$ , contendo os nós em cada um dos  $g$  grupos.

**início**

$i = 0$

$Q = V$

Calcule a centralidade de cada vértice em  $Q \in G$

**enquanto**  $Q \neq \emptyset$  **faça**

**Detecção de líderes:** Ordene  $Q$

através da centralidade, tal que

$Q = (V_1, \dots, V_n)$  com

$Cent(V_1) > \dots > Cent(V_n)$

$i = i + 1$

Selecione  $L_i = V_1$ , o líder do grupo

$i$ , como o primeiro nó da lista  $Q$

Crie o grupo  $C_i = \{L_i\}$

$Q = Q - \{L_i\}$

**Detecção de comunidades:**

Calcule a similaridade de  $L_i$  com

os demais nós em  $Q$  para

encontrar os nós similares

Defina  $S(L_i)$  como a lista de nós

similares

$C_i = C_i + S(L_i)$

$Q = Q - S(L_i)$

**fim**

**fim**

**retorna**  $C = (C_1, \dots, C_g)$

**Algoritmo 1: LCDA 1**

**Entrada:** Grafo não direcionado  $G$

**Saída:**  $C = (C_1, \dots, C_g)$ , contendo os nós em cada um dos  $g$  grupos.

**início**

$i = 0$

$Q = V$

**enquanto**  $Q \neq \emptyset$  **faça**

Calcule a centralidade de cada vértice em  $Q \in G$

**Detecção de líderes:** Ordene  $Q$

através da centralidade, tal que

$Q = (V_1, \dots, V_n)$  com

$Cent(V_1) > \dots > Cent(V_n)$

$i = i + 1$

Selecione  $L_i = V_1$ , o primeiro nó

da lista  $Q$

Crie o grupo  $C_i = \{L_i\}$

$Q = Q - \{L_i\}$

**Detecção de comunidades:**

Calcule a similaridade de  $L_i$  com

os demais nós em  $Q$  para

encontrar os nós similares

Defina  $S(L_i)$  como a lista de nós

similares

$C_i = C_i + S(L_i)$

$Q = Q - S(L_i)$

**fim**

**fim**

**retorna**  $C = (C_1, \dots, C_g)$

**Algoritmo 2: LCDA 2**

## 2.3 MÉTRICAS UTILIZADAS

Para avaliar a qualidade da detecção de comunidades em grafos, pode-se utilizar a função de modularidade (CLAUSET; NEWMAN; MOORE, 2004), definida por:

$$Q = \frac{1}{2 \times m} \sum_{ij} \left( Adj_{(i,j)} - \frac{k_i \times k_j}{2 \times m} \right) \times \delta(C_i, C_j), \quad (1)$$

em que  $Q$  é a função de modularidade a ser maximizada,  $Adj_{(i,j)}$  é a matriz de adjacente do grafo  $G$ ,  $m$  é a quantidade de nós do grafo,  $k_i$  representa o grau do  $i$ -ésimo nó,  $C_j$  é a  $j$ -ésima comunidade e  $\delta(\cdot, \cdot)$  é um função indicadora, em que  $\delta(C_i, C_j) = 1$  se os nós  $i$  e  $j$  estão contidos na mesma comunidade e igual a 0, caso contrário.

Modularidade é a fração das arestas que se enquadram nos grupos fornecidos menos a fração esperada se as arestas forem distribuídas aleatoriamente (caso nulo). O valor da modularidade está no intervalo  $[-1, 1]$  (LI; SCHUURMANS, 2011). Assim, a modularidade será positiva caso o número de arestas dentro dos grupos exceder o número esperado com base no acaso. Logo, quanto maior o valor da modularidade, melhor é será o particionamento pois maior será a diferença em relação ao caso nulo. A função  $Q$  é muito importante pois permite resumir vários aspectos da rede e ajuda na comparação com estruturas de comunidades provenientes de um grafo aleatório (chamado modelo nulo). Além disso, a modularidade aumenta se os grupos possuem indivíduos fortemente intra-conectados e diminui se os grupos possuem indivíduos fortemente inter-conectados. A métrica de modularidade vem sendo utilizada como base de vários algoritmos de otimização para detecção de comunidades, desde otimizações gulosas até otimizações espectrais (ALMEIDA, 2011).

Como métrica de avaliação da solução dos líderes, propomos a métrica na Equação (2), que é baseada na Informação Contida em um Nó (ICN) (HE; FENG; PLANT, 2011), chamada de  $H$  e apresentada a seguir:

$$H = \frac{1}{g} \sum_{j=1}^g I(L_j), \quad (2)$$

em que,  $L_j$  é o  $j$ -ésimo líder,  $g$  é o número de grupos e

$$I(L_j) = \begin{cases} H(L_j) & , \text{ se } 0 < p_1 < 1 \\ \frac{1 + \log_2(|V|)}{|V|} & , \text{ caso contrário,} \end{cases}$$

com

$$H(L_j) = - [p_1(L_j) \log_2(p_1(L_j)) + p_0(L_j) \log_2(p_0(L_j))],$$

$p_1(L_j)$  e  $p_0(L_j)$  sendo as probabilidades de "0" e "1", de um nó se conectar ou não, respectivamente com o nó  $L_j$ .

Como estratégia para otimizar ambas as métricas optou-se por uma Otimização Hierárquica, que no nosso trabalho, consiste em otimizar a modularidade  $Q$  em relação ao par de parâmetros  $(\alpha_c, \alpha_s)$  e, em seguida, otimizar a métrica  $H$  em relação ao par de parâmetros  $(\alpha_c, \alpha_s)$  e assim determinar uma "solução ótima" a ser comparada com o ótimo obtido até aquele instante.

As medidas de centralidade utilizadas nesta trabalho são:

- a) **de autovetor** A centralidade de autovetor é uma medida de influência do nó na rede que atribui relevância alta aos vértices em função de seus vizinhos. Segundo Bonacich e Lloyd (2001), essa medida é obtida através do autovetor associado ao maior autovalor da matriz de adjacência do grafo, e é dada por:

$$c_{Eig}(v_k) = x_k, \quad (3)$$

em que  $x_k$  é a  $k$ -ésima coordenada do autovetor positivo unitário  $x$  associado ao índice do grafo, isto é,

$$x_k = \frac{1}{\lambda} \sum_{j=1}^n a_{kj} x_j, \quad k = 1, \dots, n,$$

em que  $a_{kj}$  é o elemento da linha  $k$  e coluna  $j$  da matriz de adjacências  $A$  do grafo.

- b) **de intermediação**

Proposta inicialmente por Freeman (1977), a centralidade de intermediação quantifica o número de vezes que um nó age como ponte ao longo do caminho mais curto entre dois outros nós. A centralidade de intermediação pode ser representada por:

$$c_B(v_k) = \sum_{\substack{1 \leq i < j \leq n \\ i, j \neq k}} b_{ij}(v_k), \quad (4)$$

em que  $b_{ij}(v_k)$  é a intermediação parcial de  $v_k$  com respeito a  $v_i$  e  $v_j$ , com  $i \neq j$ ,  $i \neq k$  e  $j \neq k$ , dada por:

$$b_{ij} = \frac{g_{ij}(v_k)}{g_{ij}},$$

em que,  $g_{ij}$  denota a geodésicas (comprimento do menor caminho) entre o  $v_i$  e  $v_j$  e  $g_{ij}(v_k)$  denota o número de geodésicas entre  $v_i$  e  $v_j$  que passam por  $v_k$ .

- c) **de proximidade** Segundo Sabidussi (1966), o afastamento de um nó  $k$  é definido como a soma de suas distâncias para todos os outros nós, e sua proximidade é definida como o inverso do afastamento. Desta forma, quanto mais central é o nó, menores são as distân-

cias entre o mesmo e os outros nós (NEWMAN, 2005). A centralidade de proximidade do vértice  $v_k$  é expressa por:

$$c_C(v_k) = \frac{1}{\sum_{j=1}^n \text{dist}(v_j, v_k)}. \quad (5)$$

As medidas de similaridade utilizadas nesta trabalho são:

#### **Hub Pressed Index (HPI)**

$$SIM_{HPI}(v_x, v_y) = \frac{|\Gamma(v_x) \cap \Gamma(v_y)|}{\min[\kappa_{v_x}, \kappa_{v_y}]}, \quad (6)$$

#### **Jaccard**

$$SIM_{Jaccard}(v_x, v_y) = \frac{2|\Gamma(v_x) \cap \Gamma(v_y)|}{\kappa_{v_x} + \kappa_{v_y}}, \quad (7)$$

#### **Dice**

$$SIM_{Dice}(v_x, v_y) = \frac{|\Gamma(v_x) \cap \Gamma(v_y)|}{|\Gamma(v_x) \cup \Gamma(v_y)|}, \quad (8)$$

em que  $\Gamma(X)$  denota o conjunto de vizinhos do nó  $v_x$  e  $\kappa_X$  é o grau do nó  $v_x$ . Note que, pela expressão (6), quantos mais vizinhos em comum  $v_x$  e  $v_y$  possuírem, maior será a similaridade entre esses nós.

## 2.4 METAHEURÍSTICA GRASP

Entre as metaheurísticas presentes na literatura, destaca-se a metaheurística GRASP (FEO; RESENDE, 1989; FEO; RESENDE, 1995): constituindo de múltiplas realizações de busca local, cada uma iniciando de uma solução diferente. Cada iteração do algoritmo é constituída de duas etapas. Na primeira, a fase de construção, uma solução é construída por meio de algum algoritmo adaptativo aleatório guloso, este associado a uma função de custo  $c$ . A partir desta solução inicialmente construída, a segunda etapa, a fase de busca local, é realizada visando encontrar um mínimo/máximo local durante a avaliação de um conjunto de vizinhos. A melhor solução é mantida como resultado, ou seja, a aplicação da metaheurística GRASP consiste em realizar  $B$

iterações, em que, em cada uma delas, uma solução inicial é criada e, posteriormente, a realização de uma busca local é feita como uma tentativa de melhorar a qualidade da solução.

A principal vantagem em se utilizar o GRASP é a sua facilidade de implementação, além de ser uma metaheurística com poucos parâmetros a serem definidos e ajustados. O GRASP possui dois parâmetros. O primeiro está relacionado ao critério de parada, definido pelo número de iterações que serão executadas. Segundo Feo e Resende (1995), a qualidade das soluções obtidas dependem da quantidade de iterações a serem executadas, mesmo que a probabilidade de se obter uma nova solução melhor decaia conforme o número de iterações cresce. Por outro lado, o tempo computacional da execução do GRASP aumenta de forma linear ao número de iterações. Ou seja, quanto maior o número de iterações, maior será o tempo computacional e a qualidade das soluções obtidas. O segundo parâmetro está relacionado a qualidade dos elementos que irão constituir a lista restrita de candidatos a serem utilizados na fase de construção com o objetivo de selecionar os elementos que constituíram sua solução. Entretanto, dependendo do valor escolhido para esse parâmetro, pode-se não obter soluções tão boas ou passíveis de melhora e, desta forma, dificultando o algoritmo de alcançar uma solução ótima global. O algoritmo é dividida em duas etapas que serão apresentadas a seguir:

### ***Etapa 1: Construção***

A estratégia de construção de uma solução GRASP consiste na definição de um critério de avaliação dos elementos que: **(i)** possam a ser inseridos na solução para o problema de otimização; **(ii)** adapte-se à solução já contruída de tal forma que a valoração dos elementos varie durante a construção da solução. Entretanto, esse critério não é tomado como referência absoluta para a decisão do próximo elemento a ser inserido no conjunto solução, havendo assim, uma escolha (geralmente aleatória) entre os melhores elementos a cada iteração. A cada passo desta fase, o conjunto de elementos candidatos é constituído por todos os elementos do conjunto  $E$  que possam vir a serem incorporados na solução parcial em construção. A seleção do próximo elemento é determinada pela avaliação de todos os elementos candidatos de acordo com uma função de avaliação gulosa.

Seja  $c(e)$  o custo incremental associado à incorporação do elemento  $e \in E$  a uma solução em construção. Em qualquer iteração GRASP, seja  $c_{\min}$  e  $c_{\max}$ , respectivamente, o menor e o maior custo incremental. A lista restrita de candidatos (LRC) é composta dos elementos  $e \in E$  com os melhores custos incrementais  $c(e)$ . Entretanto, à LRC está associado um parâmetro

$\alpha \in [0, 1]$  que controla a gulosidade e aleatoriedade do algoritmo. Assim, para um problema de minimização, a LRC é formada por todos os elementos  $e \in E$  que podem ser inseridos na solução parcial em construção cuja qualidade está dentro dos limites definidos da seguinte forma:  $c(e) \in [c_{\min}, c_{\min} + \alpha (c_{\max} - c_{\min})]$ . O valor de  $\alpha$  influencia na qualidade e diversidade da solução gerada na fase de construção. Valores baixos de  $\alpha$  geram soluções gulosas de boa qualidade mas com pouca diversidade, enquanto que, valores altos de  $\alpha$  geram uma grande diversidade com soluções de qualidade inferior. Isso influencia, também, o desempenho do método de busca local, uma vez que, soluções inferiores tornam o processo de busca mais lento (FEO; RESENDE, 1995). Uma solução GRASP é dita gulosa, se  $\alpha = 0$ , enquanto que  $\alpha = 1$  é equivalente a uma construção aleatória.

Um elemento é selecionado aleatoriamente da LRC (esse é o aspecto probabilístico da heurística) e incorporado à solução parcial. A lista de elementos candidatos é atualizada e os custos incrementais são reavaliados (esse é o aspecto adaptativo da heurística). Aqui cabe ressaltar a importância do parâmetro  $\alpha$  para a metaheurística GRASP. As etapas acima são repetidas até que a solução esteja completa. Vale ressaltar, que a construção aleatória gulosa não é sempre capaz de produzir uma solução viável e, em alguns casos, pode ser necessário aplicar um procedimento de “reparo” à solução para obter a sua viabilidade. Exemplos de procedimentos de reparo são discutidos em Duarte, Ribeiro e Urrutia (2007), Duarte et al. (2006), Mateus, Resende e Silva (2011), Nascimento, Resende e Toledo (2008).

## ***Etapa 2: Busca Local***

As soluções geradas por uma construção aleatória gulosa não são necessariamente ótimas, mesmo no que diz respeito a vizinhanças simples. A fase de busca local visa a melhoria da solução construída. Um algoritmo de busca local funciona de maneira iterativa, substituindo sucessivamente a solução atual por uma solução melhor em sua vizinhança e termina quando nenhuma solução melhor é encontrada na vizinhança (FEO; RESENDE, 1995). A velocidade e efetividade da busca local depende de vários aspectos a saber: estrutura de vizinhança, como será a busca na vizinhança, estratégia de avaliação da função custo nos vizinhos e a própria solução inicial. A fase de construção desempenha um papel bastante importante em relação ao último aspecto, construindo soluções de partida de alta qualidade para a busca local, mais rápida será a convergência para um ótimo global como maior será a chance de encontrá-lo.

A busca local pode ser implementada usando uma estratégia de *melhor vizinho* ou de *pri-*

*meio de melhora* (FEO; RESENDE, 1995), por exemplo. Na estratégia de melhor vizinho, todos os vizinhos são investigados e o melhor deles é selecionado para compor a solução, que será a solução a ser utilizada na próxima iteração. Para a estratégia de primeira de melhora, o primeiro vizinho cujo valor da função de custo é melhor que o da solução atual é utilizado para atualizar a solução. Segundo Feo e Resende (1995), observou-se que a convergência para um ótimo local é mais provável de ocorrer com uma estratégia de melhor vizinho.

O Algoritmo 3 apresenta o pseudo-código do algoritmo GRASP.

**Entrada:**  $numero\_iteracoes, semente, \alpha$

**Saída:**  $S^*$ , a melhor solução obtidas dentre as  $k$  iterações do algoritmo

**Início**

```

para  $k = 1, \dots, numero\_iteracoes$  faça
  |  $Solucao = Contrucao\_Aleatoria\_Gulosa(Semente, \alpha)$ 
  |  $Solucao = Busca\_Local(Solucao)$ 

```

**fim**

Avalie as  $k$  soluções e selecione  $S^*$  como a melhor entre elas

**fim**

**retorna**  $S^*$

**Algoritmo 3:** GRASP

## 2.5 METAHEURÍSTICA GRASP REATIVO

O GRASP Reativo (PRAIS; RIBEIRO, 2000) difere do tradicional apenas no que diz respeito ao parâmetro  $\alpha$ , que é auto-ajustado ao longo das iterações em relação a qualidade das soluções obtidas. Proposto por (PRAIS; RIBEIRO, 2000), os autores discutem sobre o efeito do parâmetro  $\alpha$  na qualidade do ajuste e na diversidade das soluções, uma vez que, a utilização de um único valor de  $\alpha$  fixo frequentemente impede o algoritmo de produzir soluções de melhor qualidade, caso outros valores do parâmetro fossem utilizados. Além disso, o estudo realizado pelos autores mostra que a melhor solução para o problema proposto é frequentemente encontrada ao se utilizar a estratégia de variação do parâmetro  $\alpha$ .

No GRASP Reativo, o  $\alpha$  a ser utilizado na fase de construção da iteração atual é selecionado aleatoriamente a partir de um conjunto de  $\Psi = [\alpha_1, \dots, \alpha_m]$  contendo  $m$  valores pré-definidos. Para cada  $\alpha_k \in \Psi$  está associado uma probabilidade  $p_k$  de selecioná-lo. Inicialmente, todas as

probabilidades são iguais, ou seja,  $p_i = 1/m, i = 1, \dots, m$ . Após  $y$  iterações, as probabilidades  $[p_1, \dots, p_m]$  são recalculados de maneira adaptativa de forma a favorecer valores de  $\alpha$  que tenha produzido boas soluções. Desta forma, esses valores terão maiores probabilidades de serem selecionados e, por consequência, maiores chances de serem selecionados nas iterações posteriores.

Para atualizar a distribuição de probabilidades de seleção de  $\Psi$ , utiliza-se a expressão:

$$p_i = \frac{q_i}{\sum_{i=1}^m q_i}, i = 1, \dots, m, \quad (9)$$

em que

$$q_i = \frac{m_i}{f(s^*)}, i = 1, \dots, m, \quad (10)$$

Na Equação 10, considera-se  $f(s^*)$  o valor da função objetivo da melhor solução encontrada até o momento e  $m_i$  o valor médio da função objetivo de todas as soluções encontradas utilizando  $\alpha_i, i = 1, \dots, m$ . Quanto maior valor de  $m_i$ , maior será o valor de  $q_i$  e, por consequência, a probabilidade atualizada  $p_i$  será maior. Dessa forma, nos blocos de iterações seguintes, o algoritmo tende a selecionar valores de  $\alpha$  que encontram, em média, as melhores soluções.

Observando o Algoritmo 4, nota-se que o GRASP Reativo adiciona apenas uma pequena ‘sobrecarga’ sobre a versão tradicional do GRASP, uma vez que a complexidade das fases de construção e busca local não são modificadas. A eficácia do algoritmo está no fato de que a utilização de diferentes valores de  $\alpha$  em iterações distintas permite a construção de LRC diferentes, possibilitando a geração de valores diferentes. Além dessas vantagens, o GRASP Reativo pode ser visto como um algoritmo que utiliza uma memória de longo prazo. Em resumo, o que ele faz é aprender durante as iterações quais valores de  $\alpha$  resultaram em melhores soluções a favorecer a escolha desses valores. Pode-se utilizar técnicas de Aprendizagem de Máquinas para realizar essa tarefa.

O Algoritmo 4 apresenta o pseudo-código do algoritmo GRASP Reativo.

**Entrada:**  $numero\_iteracoes, semente, \Psi = [\alpha_1, \dots, \alpha_m], y$

**Saída:**  $s^*$ , a melhor solução obtidas dentre as  $k$  iterações do algoritmo

**Início**

$f(s^*) = 0$

$it = 0$

**para**  $k = 1, \dots, m$  **faça**

$p_i = 1/m$

$m_i = 0$

**fim**

**enquanto** *Criterio\_Parada* **faça**

    Seleciona-se  $\alpha = \alpha_i \in \Psi$  com probabilidade  $p_i$

$Solucao = Contrucao\_Aleatoria\_Gulosa(Semente, \alpha)$

$Solucao = Busca\_Local(Solucao)$

**se**  $f(Solucao) > f(s^*)$  **então**

$f(s^*) = Solucao$

**fim**

**se**  $it \bmod y = 0$  **então**

        Recalcular  $m_i, \forall i \in 1, \dots, m$

        Recalcular  $q_i, \forall i \in 1, \dots, m$  pela Equação 10

        Atualizar as probabilidades  $p_i, \forall i \in 1, \dots, m$  pela Equação 9

**fim**

$it = it + 1$

**fim**

**fim**

**retorna**  $s^*$

**Algoritmo 4:** GRASP REATIVO

## 2.6 INSTÂNCIAS DO PROBLEMA

Nesta Seção as seguintes redes foram utilizadas para a avaliação dos algoritmos propostos. Neste trabalho nos concentramos em redes bem estudadas, uma vez que nosso objetivo neste artigo é comparar a qualidade da detecção de líderes e comunidades alcançadas por nossos algoritmos em relação aos algoritmos LCDA, são elas:

***Zachary's karate club.*** Trata-se de uma rede social bem conhecida e bastante usada para testar algoritmos de detecção de comunidades. A rede é composta de 34 nós e 78 arestas, onde cada nó representa um membro de um clube de karatê de uma universidade americana. Dois nós são conectados se é observada uma interação social dentro ou fora do clube de karatê. Os dados foram coletados em 1977 em Zachary (1977). Mais tarde, devido a uma disputa entre o administrador do clube e o instrutor, o clube é dividido em dois grupos centrados no administrador e no instrutor, respectivamente. Um nó líder refere-se a um indivíduo central no que diz respeito a interação com os demais membros.

***American College football.*** Esta é uma rede bastante conhecida, usada também para validar algoritmos de detecção da comunidade. Representa uma rede de jogos de futebol americano entre as faculdades da Divisão IA durante a temporada regular de outono de 2000. Ela contém 115 nós representando jogadores e 616 arestas que representam as interações entre os jogadores (GIRVAN; NEWMAN, 2002). Um jogador líder é aquele que teve mais interações ou participou de jogos com os jogadores.

***Dolphin social network*** Uma rede social não direcionada de associações frequentes entre 62 golfinhos em uma comunidade que vive em Doubtful Sound, Nova Zelândia (LUSSEAU et al., 2003). Um golfinho líder de um grupo é aquele que possui mais interações com grupos sociais de golfinhos.

***Books about US politics*** Uma rede de livros sobre política norte-americana foi publicada na época da eleição presidencial de 2004 e vendida pela livraria *online* Amazon.com. As bordas representam recompras frequentes de livros pelos mesmos compradores, conforme indicado pelo recurso "clientes que compraram este livro também compraram esses outros livros" na Amazon (KREBS, 2013). Aqui, um livro líder é aquele que possui uma correlação de compras maior com os demais livros da rede.

**Political Blogs** Uma rede de *hyperlinks* entre *web blogs* na política dos EUA, gravada em 2005 por Adamic e Glance. A rede é composta por 1490 nós e 19090 arestas (LUSSEAU et al., 2003). Aqui, um blog líder é aquele que difunde conteúdos correlatos com grande parte dos demais blogs da rede.

### 3 ALGORITMOS PROPOSTOS

Neste capítulo apresentamos os quatro algoritmos desenvolvidos utilizando os algoritmos LCDA 1 e 2 na fase de construção dos algoritmos GRASP e GRASP Reativo. Inicialmente, apresentaremos os algoritmos LCDA-GRASP 1 e 2 desenvolvidos a partir da metaheurística GRASP, nos quais apresentamos suas construções e diferenças. Em seguida, apresentamos os algoritmos LCDA-GR, uma extensão dos algoritmos LCDA-GRASP utilizando a metaheurística GRASP Reativo.

#### 3.1 ALGORITMOS LCDA-GRASP

Nesta Seção, abordaremos a adaptação dos algoritmos LCDA 1 e 2 com o objetivo de utilizá-los na fase de construção dos algoritmos LCDA-GRASP 1 e 2, respectivamente.

##### ***Etapa 1: Algoritmo de construção da solução inicial:***

###### ***Detecção de Líderes:***

Considere  $f(e)$  a função gulosa, a LRC é formada pelos nós candidatos à solução  $e$  de maiores valores de acordo com  $f(e)$ , já que nosso objetivo é de encontrar o nó com maior liderança para a comunidade. Assim, os elementos tais que  $f(e) \in [f_{\min} + \alpha_c (f_{\max} - f_{\min}), f_{\max}, ]$  são inseridos em ordem decrescente na LRC. Um elemento da LRC é selecionado aleatoriamente para ser o nó líder.

###### ***Detecção de Comunidades:***

A partir do líder determinado, os membros da comunidade são escolhidos. Para tanto, uma medida de similaridade é utilizada como função gulosa.

Para cada nó candidato, ou seja, ainda não adicionado a alguma comunidade, a função gulosa  $g(e)$  é calculada. Os elementos cujos valores de similaridade  $g(e)$  tais que  $g(e) \in [g_{\min} + \alpha_s (g_{\max} - g_{\min}), g_{\max}, ]$  são inseridos em ordem decrescente na LRC. Todos os elementos pertencentes à LRC são selecionados para formar esta comunidade. As etapas são repetidas até que todos os nós sejam classificados. É interessante observar que, para valores

altos de  $\alpha_s$ , menor será a quantidade de elementos que constituem a LRC, que, por consequência, aumenta a quantidade de grupos gerados na fase de construção. De modo geral, para o problema em questão, o parâmetro  $\alpha_s$  está diretamente relacionado a quantidade de grupos gerados na fase de construção.

Conforme Ahajjam, Haddad e Badir (2018) são propostos dois novos métodos de construção baseados nos LCDA 1 e LCDA 2 originais. No primeiro, a centralidade de nós é calculada uma única vez no início do algoritmo. Já no segundo, a função gulosa de centralidade de nós é recalculada após a construção de cada comunidade com os nós restantes. Os Algoritmos 5 e 6 apresentam o pseudo-código dos algoritmos de construção LCDA-GRASP 1 e 2 propostos, respectivamente.

**Entrada:** Grafo não direcionado  $G(V, A)$ ,  
*semente*,  $\alpha_c$ ,  $\alpha_s$

**Saída:** *Solucao*, uma solução aleatória gulosa

$i = 0$   
 $Q = V$

Calcule a centralidade, *Cent* de cada vértice em  
 $Q \in G$

**enquanto**  $Q \neq \emptyset$  **faça**

**Detecção de líderes:** Calcule a LCR por:  
 $LCR_L = \{e \in E | Cent(e) \leq \min(Cent) + \alpha_c * (\max(Cent) - \min(Cent))\}$   
 $i = i + 1$   
 Selecione aleatoriamente um elemento  
 $L_i = V_1$ , o líder do grupo  $i$ , da  $LCR_L$   
 Crie o grupo  $C_i = \{L_i\}$   
 $Q = Q - \{L_i\}$

**Detecção de comunidades:** Calcule a similaridade,  $S$ , de  $L_i$  com os demais nós em  $Q$  para encontrar os nós similares  
 Calcule a LCR por:  $LCR_S = \{v \in Q | S(e) \leq \min(S) + \alpha_s * (\max(S) - \min(S))\}$   
 $C_i = C_i + LCR_S$   
 $Q = Q - LCR_S$

**fim**

**retorna**  $Solucao = \bigcup_i C_i$   
**Algoritmo 5:** Fase de construção: LCDA-GRASP 1

**Entrada:** Grafo não direcionado  $G$ ,  
*semente*,  $\alpha_c$ ,  $\alpha_s$

**Saída:** *Solucao*, uma solução aleatória gulosa

$i = 0$   
 $Q = V$

**enquanto**  $Q \neq \emptyset$  **faça**

**Detecção de líderes:** Calcule a centralidade, *Cent* de cada vértice em  
 $Q \in G$   
 Calcule a LCR por:  
 $LCR_L = \{e \in E | Cent(e) \leq \min(Cent) + \alpha_c * (\max(Cent) - \min(Cent))\}$   
 $i = i + 1$   
 Selecione aleatoriamente um elemento  
 $L_i = V_1$ , o líder do grupo  $i$ , da  $LCR_L$   
 Crie o grupo  $C_i = \{L_i\}$   
 $Q = Q - \{L_i\}$

**Detecção de comunidades:** Calcule a similaridade,  $S$ , de  $L_i$  com os demais nós em  $Q$  para encontrar os nós similares  
 Calcule a LCR por:  $LCR_S = \{v \in Q | S(e) \leq \min(S) + \alpha_s * (\max(S) - \min(S))\}$   
 $C_i = C_i + LCR_S$   
 $Q = Q - LCR_S$

**fim**

**retorna**  $Solucao = \bigcup_i C_i$   
**Algoritmo 6:** Fase de construção: LCDA-GRASP 2

## Etapa 2: Busca Local

Dada uma solução  $s$  viável, tem-se que todo nó  $e_i \in s$ ,  $i = 1, \dots, n$ , pertence a uma comunidade  $C_j$ ,  $j = 1, \dots, d$ , em que  $d$  é o número de comunidades detectadas na solução  $s$ . A estrutura de vizinhança é definida pela troca de cada nó  $e_i$  para a comunidade  $C_k$  tal que  $C_k \neq C_j$ ,  $k = 1, \dots, d$ ,  $\forall i, j$ . Neste trabalho como estratégia de busca utilizou-se a primeiro de melhora, assim, a primeira mudança de nó que gera uma melhoria na modularidade é realizada e a busca é reiniciada.

Desta forma, nossa busca local, a cada iteração, precisa calcular uma matriz  $\Delta Q$ , com dimensão  $n \times d$ , representando a variação da modularidade ao mudar o nó  $i$  para o grupo  $j$ , para  $i = 1, \dots, n$  e  $j = 1, \dots, d$ . Em outras palavras, é a diferença da modularidade antes da mudança,  $Q^{(t)}$ , e depois da mudança,  $Q^{(t+1)}$ . Desta forma, sejam  $C_i$ , com  $i = 1, \dots, n$ , representado a comunidade ao qual o nó  $i$  pertence, podemos representar  $\Delta Q$ , a variação da modularidade entre  $Q^{(t+1)}$  e  $Q^{(t)}$  da seguinte forma:

$$\begin{aligned} \Delta Q(i, j) &= Q^{(t+1)} - Q^{(t)} \\ &= Q^{(t)} + \sum_{l=1}^n \sum_{j=1}^d (s_{l,j}^{(2)} - s_{l,j}^{(1)}) d_{l,j} - Q^{(t)} \\ &= \sum_{l=1}^n \sum_{j=1}^d (s_{l,j}^{(2)} - s_{l,j}^{(1)}) d_{l,j}, \end{aligned}$$

em que,  $d_{l,j} = \left( a_{l,j} - \frac{k_l \times k_j}{2 \times m} \right)$ , como na Equação (1) e os vetores  $s_{l,j}^{(1)}$  e  $s_{l,j}^{(2)}$ , são representados por:

$$s_{l,j}^{(1)} = \begin{cases} 1 & , \text{ se } C_l = C_i \\ 0 & , \text{ caso contrário,} \end{cases}$$

e

$$s_{l,j}^{(2)} = \begin{cases} 1 & , \text{ se } C_l = j \\ 0 & , \text{ caso contrário.} \end{cases}$$

Entretanto, para instâncias com  $n > 300$ , através de estudos empíricos observou-se que o custo computacional exigido para obter soluções através dos algoritmos LCDA-GRASP 1 e 2 se torna bastante alto em comparação as demais instâncias com menores valores de  $n$ . Além

disso, para rede com  $n$  grande, o número maior de melhoras deve ser aplicado até que se atinja o máximo local. Desta forma, para essas instâncias, propomos uma busca de Multi-Melhorias com Vizinhança Variável.

Nossa estratégia para redes maiores consiste em, primeiramente, utilizar o método de Vizinhança Variável (RIOS et al., 2017). Restringimos o conjunto de nós  $V$  para um novo conjunto  $V'$  com  $n'$  nós, tal que  $n > n'$ , com o objetivo de reduzir a dimensão da matriz  $\Delta Q$  e, conseqüentemente, o custo computacional para calculá-la. Desta forma, calcula-se  $\Delta Q' = \Delta Q|_{V'}$  restrito a  $V'$ . Em seguida, combinado a estratégia de Primeira Melhorias, utilizamos a estratégia de Multi-Melhorias (RIOS et al., 2017), em que realizamos todas as primeiras melhorias possíveis para cada nó, ou seja, é realizada a primeira melhora para cada nó  $v'_i \in V'$  caso  $\Delta Q(v'_i, j) > 0$ , para algum  $j$ . Esses passos são repetidos até que  $\max(\Delta Q') < \epsilon$ , ou seja, não há uma melhora maior ou igual a um limite  $\epsilon$ . Neste trabalho adotou-se  $\epsilon = 10^{-4}$ .

Neste trabalho, utilizamos  $n' = 300$ . A escolha do valor de  $n'$  foi decidida a partir de estudos empíricos realizados uma vez que valores superiores a este e, em vários casos, valores menores aumentavam ou possuíam o mesmo custo computacional exigido pelos algoritmos.

### ***Etapa de reparação em soluções inviáveis***

Ao utilizar essas metodologias, os algoritmos podem obter soluções em que uma ou mais comunidades podem ter mais de um líder ou nenhum líder. Desta forma, realizou-se o seguinte procedimento de reparação: para as comunidades com mais de um líder ou sem líder, calcula-se a centralidade para todos nós e o líder é definido como o nó com maior centralidade. As etapas de construção e busca local são repetidas  $B$  vezes (número de iterações do algoritmo GRASP, inicialmente definido) e, dentre as  $B$  soluções geradas, seleciona-se aquela que tiver a maior modularidade.

## 3.2 ALGORITMOS LCDA-GR

Dado que, os algoritmos LCDA-GRASP 1 e 2 são por demais custosos, principalmente, quando aplicados a redes maiores com mais de mil nós, tornando-os custosos computacionalmente quando se objetiva determinar o melhor conjunto de parâmetros desses algoritmos. Visando manter a qualidade das soluções e reduzir o custo computacional dos algoritmos LCDA-GRASP 1 e 2, desenvolve-se dois novos algoritmos, em que estes foram criados a partir da

adaptação desse algoritmos a metaheurística GRASP Reativo, aqui chamados de algoritmos LCDA-GR 1 e 2. O GRASP Reativo (PRAIS; RIBEIRO, 2000) tem a vantagem de autoajustar seus parâmetros ao longo de suas iterações. Desta forma, podemos obter soluções baseadas nos parâmetros que proporcionam soluções melhores para o grafo em questão, reduzindo, desta forma o número de iterações a serem realizadas para ajustar os parâmetros e métricas dos algoritmos LCDA-GRASP 1 e 2 originais.

Definiu-se, através de estudos empíricos, os valores  $m = 20$  e  $y = 3 * m$ . O conjunto  $\Psi$  é constituído por  $m$  vetores com  $(\alpha_{ck}, \alpha_{sk})$ , tal que  $\alpha_c \in [0, 1]$  e  $\alpha_s \in [0, 1]$ . Como uma forma de reduzir o custo computacional do algoritmo, optou-se por selecionar os valores de ambos os parâmetros  $\alpha_c$  e  $\alpha_s$  aleatoriamente dos intervalos  $[0, 1; 0, 9]$  e  $[0, 1; 0, 5]$ , respectivamente, uma vez que, através dos estudos realizados na Seção 3.3 valores de  $\alpha_s > 0, 5$  proporcionam um custo computacional maior e nem sempre garantem uma melhora nas soluções obtidas. As probabilidades de seleção de um vetor de parâmetros dentro do conjunto  $\Psi$  é definida como no Capítulo 2 e recalculadas como nas Equações 9 e 10.

Diferente dos algoritmos GRASP Reativos, optamos por avaliar as soluções em relação a uma métrica de qualidade da detecção das comunidades e a uma métrica de qualidade na detecção dos líderes. Desta forma, propomos uma Otimização Hierárquica para todos os algoritmos através da otimização de Métrica-L2 Relativa que serão melhores discutidos. Como consequência, nossos algoritmos auto-ajustam os parâmetros  $\alpha_c$  e  $\alpha_s$  dos algoritmos LCDA-GRASP 1 e 2 levando em consideração métricas de qualidade de detecção de líderes e comunidades, uma vez, os parâmetros  $\alpha_c$  e  $\alpha_s$  estão diretamente relacionados a qualidade de detecção de líderes e comunidades.

Nestes algoritmos optou-se por, além de buscar soluções de boa qualidade em relação a modularidade, optamos por melhorar, também a qualidade com que os líderes são selecionados. Como medida para avaliar a qualidade da detecção de comunidades em grafos, pode-se utilizar a função de modularidade (CLAUSET; NEWMAN; MOORE, 2004), apresentada na Equação (1). Como métrica de avaliação da solução dos líderes, propomos a métrica na Equação (2), que é baseada na Informação Contida em um Nó (ICN) (HE; FENG; PLANT, 2011), chamada de  $H$ .

Para avaliar conjuntamente a qualidade dos líderes e comunidades conjuntamente, considere a função objetivo  $f_{s^*}$ , a seguir:

$$f(s^*) = \left[ \frac{(Q - Q_{melhor})}{Q_{melhor}}, \frac{(H - H_{melhor})}{H_{melhor}} \right], \quad (11)$$

em que  $Q$  e  $H$  são a modularidade e a métrica na Equação (2), respectivamente, para uma solução  $s^*$  e  $Q_{melhor}$  e  $H_{melhor}$  são as melhores modularidade e métrica na Equação (2) obtidas até o momento pelo algoritmo.

As fases de construção dos algoritmos LCDA-GR permanecem as mesmas que as dos algoritmos LCDA-GRASP 1 e 2.

### 3.3 RESULTADOS COMPUTACIONAIS

Neste capítulo são apresentados os resultados obtidos pelos algoritmos propostos para as seguintes instâncias *Zachary's Karate Club*, *American College football*, *Dolphin social network*, *Books about US politics* e *Political Blogs* descritos no Capítulo 2

#### 3.3.1 LCDA-GRASP

Os algoritmos propostos foram implementados usando o software estatístico R (R Core Team, 2018) e compilado na versão 8.2.0. Os testes foram realizados em um máquina virtual no Google Cloud com 20 *HighCPU* núcleos e 18GB de memória RAM rodando Ubuntu 18.04.

Foram realizados três estudos de simulação a fim de avaliar o desempenho dos algoritmos. O primeiro é um estudo mais extenso sobre os efeitos das escolhas dos parâmetros  $\alpha_c$  e  $\alpha_s$  no desempenho dos algoritmos. Em seguida, após a escolha de valores de  $\alpha_c$  e  $\alpha_s$ , realizou-se um segundo estudo de simulação para avaliar o desempenho dos algoritmos sob diferentes escolhas das medidas de centralidade e similaridade. Por fim, apresentamos uma comparação entre os resultados dos algoritmos propostos com outros algoritmos presentes na literatura.

Foram realizados testes de Kruskal Wallis (KRUSKAL; WALLIS, 1952) com o objetivo de comparar as médias da modularidade dos algoritmos em relação aos parâmetros  $\alpha_c$  e  $\alpha_s$  e das métricas de similaridade e centralidade adotadas neste trabalho. Em seguida, Teste de Permutação (GOOD, 2013) foram realizados para as comparações múltiplas dos níveis dos parâmetros e métricas. O resultados de ambos os testes são apresentados nas Tabelas 3, 6, 9, 12, 15, 18, 21, 24, 27 e 30, em que, letras iguais estão representando médias iguais e letras diferentes representando médias estatisticamente diferentes, ao nível de 5% de significância.

### 3.3.1.1 Estudo de Simulação 1

Os resultados apresentados dizem respeito a 30 execuções independentes dos algoritmos LCDA-GRASP 1 e 2 para cada instância utilizando os seguintes parâmetros do GRASP,  $\alpha_c = (0, 1; 0, 3; 0, 5; 0, 7; 0, 9)$  e  $\alpha_s = (0, 1; 0, 3; 0, 5; 0, 7; 0, 9)$  e  $B = 50$  (número de iterações). Na avaliação das soluções geradas pelos algoritmos LCDA-GRASP 1 e 2, além da modularidade, foram computados o tempo de processamento (*tempo* em segundos) e iteração ótima. Foram realizados teste de permutação para as médias da modularidade para avaliar as diferenças entre os níveis dos parâmetros  $\alpha_c$  e  $\alpha_s$  em estudo, uma vez que, em alguns níveis foi observado variância zero.

No teste de permutação, comparamos o valor da estatística observada com as saídas apresentadas através do algoritmo computacional, que consiste em criar re-amostras por simples permutações dos dados originais, maiores detalhes podem ser vistos, por exemplo, em Duczmal et al. (2003). Se estas saídas produzem estatísticas com valores semelhantes à estatística observada nos dados, então a estatística observada terá sido um mero acaso. O teste terá validade, desde que se mantenha a estrutura dos dados. Os testes de permutação apresentam uma vantagem sobre os testes clássicos, que é não necessitar da suposição de aleatoriedade na coleta dos dados, e nem mesmo de uma suposição inicial da função de densidade da estatística em estudo (EDGINGTON, 1995)

#### *Zachary's Karate Club*

As Tabelas 1 e 3 apresentam as modularidades máximas e médias, respectivamente, obtidas pelos algoritmos LCDA-GRASP 1 e 2 em relação aos parâmetros  $\alpha_c$  e  $\alpha_s$ . As medidas de modularidades máximas e médias que atingiram o valor 0,4198, o mesmo encontrado como melhor solução por Atay et al. (2017), Agarwal e Kempe (2008), Li e Schuurmans (2011), foram destacadas em negrito. Analisando a Tabela 1 pode-se destacar que valores de  $\alpha_c \leq 0, 7$  apresentam as maiores modularidades quando combinados com medidas de  $\alpha_s \geq 0, 3$ , em relação ao algoritmo LCDA-GRASP 1. O mesmo comportamento pode ser observado nos resultado obtidos pelo algoritmo LCDA-GRASP 2. A Tabela 2 apresenta o tempo médio de execução, em segundos, dos algoritmos propostos para diferentes valores dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede social *Zachary's Karate Club*. Pode-se observar que, de forma geral, conforme o valor do parâmetro  $\alpha_s$  cresce, de modo geral, o tempo de execução médio também cresce para ambos os

algoritmos.

Method	$\alpha_c$	$\alpha_s$				
		0,1	0,3	0,5	0,7	0,9
LCDA-GRASP 1	0,1	0,3982	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	0,3	0,3922	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	0,5	0,3922	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	0,7	0,3922	<b>0,4198</b>	0,4020	<b>0,4198</b>	<b>0,4198</b>
	0,9	0,3922	0,4020	0,3922	0,4020	0,3918
LCDA-GRASP 2	0,1	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	0,3	0,4020	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	0,5	0,4020	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	0,7	0,4020	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	0,9	0,3922	0,3922	0,3922	<b>0,4198</b>	<b>0,4198</b>

Tabela 1 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Zachary's Karate Club*.

Method	$\alpha_c$	$\alpha_s$				
		0,1	0,3	0,5	0,7	0,9
LCDA-GRASP 1	0,1	0,44	0,45	0,54	0,97	1,16
	0,3	0,46	0,46	0,58	0,98	1,26
	0,5	0,47	0,49	0,56	0,94	1,33
	0,7	0,46	0,44	0,55	0,98	1,35
	0,9	0,40	0,39	0,45	0,99	1,30
LCDA-GRASP 2	0,1	0,44	0,45	0,52	0,99	1,20
	0,3	0,46	0,46	0,55	1,00	1,31
	0,5	0,47	0,48	0,54	0,99	1,42
	0,7	0,46	0,45	0,51	1,03	1,56
	0,9	0,39	0,37	0,44	1,04	1,66

Tabela 2 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Zachary's Karate Club*.

Na Tabela 3, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Ao observar as médias apresentadas na Tabela 3, temos que as combinações de  $\alpha_c \leq 0,7$  e  $\alpha_s \in [0,3; 0,5]$ ,  $\alpha_s = 0,7$  e  $\alpha_c = \{0,1; 0,3; 0,9\}$  e as combinações  $\alpha_s = 0,9$   $\alpha_c = \{0,1; 0,9\}$  apresentam os maiores valores médios da modularidade observada em ambos os algoritmos.

Method	$\alpha_c$	$\alpha_s$					Média	Decision
		0,1	0,3	0,5	0,7	0,9		
LCDA-GRASP 1	0,1	0,3960	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	0,4150	a
	0,3	0,3922	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	0,4110	0,4125	b
	0,5	0,3922	<b>0,4198</b>	<b>0,4198</b>	0,4168	0,4061	0,4009	c
	0,7	0,3922	<b>0,4198</b>	<b>0,4198</b>	0,4070	0,4161	0,4110	c
	0,9	0,3922	0,4020	0,3922	<b>0,4198</b>	<b>0,4198</b>	0,4052	c
	Média	0,3929	0,4162	0,4143	0,4166	0,4146	0,4109	
	Decision	A	B	BC	B	C		
LCDA-GRASP 2	0,1	0,4151	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	0,4188	a
	0,3	0,4020	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	0,4184	0,4160	b
	0,5	0,4011	<b>0,4198</b>	<b>0,4198</b>	0,4185	0,4189	0,4156	c
	0,7	0,4020	<b>0,4198</b>	<b>0,4198</b>	0,4189	<b>0,4198</b>	0,4161	c
	0,9	0,3911	0,3922	0,3922	<b>0,4198</b>	<b>0,4198</b>	0,4030	c
	Média	0,4022	0,4143	0,4143	0,4194	0,4193	0,4139	
	Decision	A	B	B	C	C		

Tabela 3 – Modularidade média obtida pelos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Zachary's Karate Club*.

Desta forma, a combinação  $\alpha_c = 0,1$  e  $\alpha_s = 0,3$  que obtém medidas de modularidade ótimas ao mesmo tempo que executam ambos os algoritmos de forma mais eficiente em relação às demais combinações dos parâmetros  $\alpha_c$  e  $\alpha_s$ .

#### *American College football*

As Tabelas 4 e 6 apresentam as modularidades máximas e média, respectivamente obtidas pelos algoritmos LCDA-GRASP 1 e 2 em relação aos parâmetros  $\alpha_c$  e  $\alpha_s$ . As medidas de modularidade máximas e médias que atingiram o valor 0,6046 foram destacadas em negrito, sendo o mesmo resultado encontrado nas soluções ótimas de Atay et al. (2017), Agarwal e Kempe (2008), Li e Schuurmans (2011). Analisando a Tabela 4 pode-se destacar que as combinações de  $\alpha_s = 0,3$  com quais valores de  $\alpha_c$  apresentaram as maiores modularidades. Além destas combinações, para o algoritmo LCDA-GRASP 1 as combinações de  $\alpha_s = 0,5$  e  $\alpha_c = \{0,1; 0,3; 0,9\}$  e, para o algoritmo LCDA-GRASP 2, as combinações de  $\alpha_s = 0,5$  e  $\alpha_c \leq 0,5$ , apresentaram também as maiores modularidades. A Tabela 5 apresenta o tempo médio de execução dos algoritmos propostos para diferentes valores dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede social *American College football*. Pode-se observar que, conforme o valor do parâmetro  $\alpha_s$  cresce, de modo geral, o tempo de execução médio também cresce para ambos os algoritmos, com exceção de

$\alpha_s = 0,5$ , que apresenta os menores tempos de execução.

Method	$\alpha_c$	$\alpha_s$				
		0,1	0,3	0,5	0,7	0,9
LCDA-GRASP 1	0,1	0,5923	<b>0,6046</b>	<b>0,6046</b>	0,6031	0,6010
	0,3	0,5851	<b>0,6046</b>	<b>0,6046</b>	0,6031	0,6031
	0,5	0,5844	<b>0,6046</b>	0,6032	0,6031	0,6032
	0,7	0,5844	<b>0,6046</b>	0,6032	0,6031	0,6031
	0,9	0,5720	<b>0,6046</b>	<b>0,6046</b>	0,6031	0,6031
LCDA-GRASP 2	0,1	0,5828	<b>0,6046</b>	<b>0,6046</b>	0,6031	0,6031
	0,3	0,5844	<b>0,6046</b>	<b>0,6046</b>	0,6031	0,6010
	0,5	0,5844	<b>0,6046</b>	<b>0,6046</b>	0,6032	0,6010
	0,7	0,5830	<b>0,6046</b>	0,6032	0,6032	0,6010
	0,9	0,5752	<b>0,6046</b>	0,6032	0,6031	0,6010

Tabela 4 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *American College football*.

Method	$\alpha_c$	$\alpha_s$				
		0,1	0,3	0,5	0,7	0,9
LCDA-GRASP 1	0,1	5,03	5,76	4,34	8,38	34,93
	0,3	5,08	5,84	4,00	8,13	34,85
	0,5	4,90	5,48	3,32	7,82	34,82
	0,7	5,18	5,32	3,18	7,72	35,66
	0,9	4,87	5,56	3,22	9,36	38,24
LCDA-GRASP 2	0,1	4,85	5,69	4,15	8,43	35,46
	0,3	4,76	5,39	3,80	7,97	35,09
	0,5	4,36	4,97	2,95	7,73	35,15
	0,7	4,41	4,81	2,70	7,83	35,24
	0,9	4,06	4,95	2,88	7,94	37,25

Tabela 5 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *American College football*.

Na Tabela 6, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 6, tem-se que  $\alpha_s = 0,3$  combinado a quaisquer valor de  $\alpha_c$ , para o algoritmo LCDA-GRASP 1, e  $\alpha_c \leq 0,7$ , para o algoritmo LCDA-GRASP 2 apresentam as maiores modularidades médias.

Method	$\alpha_c$	$\alpha_s$					Média	Decision
		0,1	0,3	0,5	0,7	0,9		
LCDA-GRASP 1	0,1	0,5672	<b>0,6046</b>	0,6035	0,6023	0,6010	0,5957	a
	0,3	0,5676	<b>0,6046</b>	0,6034	0,6026	0,6013	0,5959	a
	0,5	0,5646	<b>0,6046</b>	0,6032	0,6028	0,6012	0,5953	a
	0,7	0,5677	<b>0,6046</b>	0,6032	0,6023	0,6011	0,5958	a
	0,9	0,5577	<b>0,6046</b>	0,6033	0,6025	0,6012	0,5939	a
	Média	0,5649	<b>0,6046</b>	0,6033	0,6025	0,6012	0,5953	
	Decision	A	B	C	D	E		
LCDA-GRASP 2	0,1	0,5655	0,6045	0,6035	0,6022	0,6011	0,5954	a
	0,3	0,5666	<b>0,6046</b>	0,6033	0,6026	0,6010	0,5956	a
	0,5	0,5634	<b>0,6046</b>	0,6033	0,6027	0,6010	0,5950	a
	0,7	0,5643	<b>0,6046</b>	0,6032	0,6029	0,6010	0,5952	a
	0,9	0,5486	<b>0,6046</b>	0,6032	0,6025	0,6010	0,5920	a
	Média	0,5617	<b>0,6046</b>	0,6033	0,6026	0,6010	0,5946	
	Decision	A	B	C	D	E		

Tabela 6 – Modularidade média obtida pelos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *American College football*.

Desta forma, a combinação  $\alpha_c = 0,7$  e  $\alpha_s = 0,3$  fornecem melhores medidas de modularidade médias ao mesmo tempo que executam ambos os algoritmos de forma mais eficiente em relação a tempo computacional.

#### *Dolphin social network*

As Tabelas 7 e 9 apresentam as modularidades máximas e médias, respectivamente, obtidas pelos algoritmos LCDA-GRASP 1 e 2 em relação aos parâmetros  $\alpha_c$  e  $\alpha_s$ . As medidas de modularidade máximas e médias que atingiram o valor 0,5285 foram destacadas em negrito, uma vez que, foi o mesmo resultado encontrado nas soluções ótimas de Atay et al. (2017), Agarwal e Kempe (2008), Li e Schuurmans (2011). Analisando a Tabela 7 pode-se destacar que, para os resultados do algoritmo LCDA-GRASP 1, de modo geral, valores de  $0,3 \leq \alpha_c \leq 0,7$  combinados com  $0,3 \leq \alpha_s \leq 0,7$  obtiveram a maior modularidade dentre as demais combinações de  $\alpha_c$  e  $\alpha_s$ . Para os resultados do algoritmo LCDA-GRASP 2, as combinações  $\alpha_s = 0,1$  e  $\alpha_c \leq 0,5$ ,  $\alpha_s = 0,7$  e  $\alpha_c = 0,1$  e  $\alpha_s = 0,3$  com quaisquer valor de  $\alpha_c$ , apresentaram as maiores modularidade dentre as demais combinações. A Tabela 8 apresenta o tempo médio de execução dos algoritmos propostos para diferentes valores dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede social *Dolphin social network*. Nota-se que, conforme os valores de  $\alpha_s$  aumentam, maiores são os tempos de

execução médios dos algoritmos.

Method	$\alpha_c$	$\alpha_s$				
		0,1	0,3	0,5	0,7	0,9
LCDA-GRASP 1	0,1	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>	0,5277
	0,3	0,5268	<b>0,5285</b>	<b>0,5285</b>	0,5276	0,5202
	0,5	0,5268	<b>0,5285</b>	0,5276	<b>0,5285</b>	0,5195
	0,7	0,5268	0,5277	<b>0,5285</b>	0,5277	0,5247
	0,9	0,5268	0,5251	<b>0,5285</b>	0,5277	0,5247
LCDA-GRASP 2	0,1	0,5276	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>	0,5276
	0,3	0,5268	<b>0,5285</b>	<b>0,5285</b>	0,5277	0,5196
	0,5	0,5268	<b>0,5285</b>	<b>0,5285</b>	0,5277	0,5247
	0,7	0,5268	0,5277	<b>0,5285</b>	0,5277	0,5247
	0,9	0,5268	0,5251	<b>0,5285</b>	0,5277	0,5247

Tabela 7 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Dolphin social network*.

Method	$\alpha_c$	$\alpha_s$				
		0,1	0,3	0,5	0,7	0,9
LCDA-GRASP 1	0,1	1,60	1,98	2,52	4,99	6,95
	0,3	1,35	1,76	2,49	5,01	7,69
	0,5	1,24	1,64	2,38	5,02	7,90
	0,7	1,06	1,32	2,24	4,85	7,64
	0,9	1,03	1,29	2,18	4,55	6,78
LCDA-GRASP 2	0,1	1,63	1,95	2,54	4,98	6,75
	0,3	1,30	1,73	2,46	5,09	7,71
	0,5	1,24	1,66	2,34	5,04	7,79
	0,7	1,05	1,31	2,29	4,91	7,70
	0,9	1,05	1,30	2,15	4,56	6,94

Tabela 8 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Dolphin social network*.

Na Tabela 9, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Os resultados apresentados na Tabela 9 mostra que  $\alpha_s = 0,3$ , em ambos os algoritmos, apresentam as maiores modularidades médias. Além disto, em ambos os algoritmos, em média, valores de  $\alpha_c \in [0,3; 0,7]$  proporcionam as maiores modularidades.

Method	$\alpha_c$	$\alpha_s$					Média	Decision
		0,1	0,3	0,5	0,7	0,9		
LCDA-GRASP 1	0,1	0,5235	0,5269	0,5251	0,5238	0,5211	0,5241	a
	0,3	0,5234	0,5271	0,5249	0,5209	0,5131	0,5219	b
	0,5	0,5220	0,5274	0,5220	0,5218	0,5132	0,5213	b
	0,7	0,5253	0,5271	0,5245	0,5223	0,5130	0,5224	b
	0,9	0,5244	0,5241	0,5239	0,5190	0,5170	0,5217	b
	Média	0,5237	0,5265	0,5241	0,5216	0,5155	0,5223	
	Decision	A	B	A	C	D		
LCDA-GRASP 2	0,1	0,5245	0,5270	0,5256	0,5223	0,5203	0,5239	a
	0,3	0,5231	0,5269	0,5255	0,5216	0,5141	0,5222	b
	0,5	0,5193	0,5272	0,5223	0,5218	0,5140	0,5209	c
	0,7	0,5239	0,5275	0,5248	0,5215	0,5137	0,5223	bc
	0,9	0,5253	0,5239	0,5239	0,5205	0,5165	0,5220	bc
	média	0,5232	0,5265	0,5244	0,5215	0,5157	0,5223	
	Decision	A	B	C	D	E		

Tabela 9 – Modularidade média obtida pelos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Dolphin social network*.

A partir desses resultados, temos que a combinação  $\alpha_c = 0,3$  e  $\alpha_s = 0,3$  fornece bons resultados dos algoritmos em relação a modularidade média e máxima, assim como para os tempos de execução médios de ambos os algoritmos.

#### *Books about US politics*

As Tabelas 10 e 12 apresentam as modularidades máximas e média, respectivamente obtidas pelos algoritmos LCDA-GRASP 1 e 2 em relação aos parâmetros  $\alpha_c$  e  $\alpha_s$ . As medidas de modularidade máximas e médias que atingiram o valor 0,5272 foram destacadas em negrito, dado que, foi o mesmo resultado encontrado nas soluções ótimas de Atay et al. (2017), Agarwal e Kempe (2008), Li e Schuurmans (2011). Analisando a Tabela 10 pode-se destacar que, para os resultados do algoritmo LCDA-GRASP 2, as combinações de  $0,3 \leq \alpha_s \leq 0,5$  e qualquer valor de  $\alpha_c$  e  $\alpha_s = 1$  combinado com  $\alpha_c \leq 0,5$ , fornecem as maiores modularidades máxima observada. Para o algoritmo LCDA-GRASP 1, as combinações de  $0,3 \leq \alpha_s \leq 0,5$  e qualquer valor de  $\alpha_c$  e  $\alpha_s = 1$  combinado com  $\alpha_c \leq 0,7$ , apresentam as maiores modularidades máximas observadas. A Tabela 11 apresenta o tempo médio de execução dos algoritmos propostos para diferentes valores dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede social *Books about US politics*. Nota-se que, conforme os valores de  $\alpha_s$  aumentam, maiores são os tempos de execução médios de am-

bos os algoritmos. Para o algoritmo LCDA-GRASP 2, pode-se observar que, quanto maior os valores de  $\alpha_c$ , maior será o tempo execução do algoritmo. O mesmo comportamento é observado nos resultados do algoritmo LCDA-GRASP 1 para valores de  $\alpha_s \leq 0,5$ .

Method	$\alpha_c$	$\alpha_s$				
		0,1	0,3	0,5	0,7	0,9
LCDA-GRASP 1	0,1	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>	0,5259	0,5237
	0,3	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>	0,5264	0,5237
	0,5	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>	0,5264	0,5237
	0,7	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>	0,5264	0,5237
	0,9	0,5269	<b>0,5272</b>	<b>0,5272</b>	0,5261	0,5237
LCDA-GRASP 2	0,1	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>	0,5237	0,5207
	0,3	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>	0,5264	0,5237
	0,5	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>	0,5264	0,5237
	0,7	0,5269	<b>0,5272</b>	<b>0,5272</b>	0,5264	0,5237
	0,9	0,5269	<b>0,5272</b>	<b>0,5272</b>	0,5264	0,5207

Tabela 10 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Books about US politics*.

Method	$\alpha_c$	$\alpha_s$				
		0,1	0,3	0,5	0,7	0,9
LCDA-GRASP 1	0,1	3,77	5,13	8,35	19,89	39,84
	0,3	2,90	3,98	7,43	17,85	36,28
	0,5	2,43	3,14	7,30	17,48	36,11
	0,7	2,35	2,58	6,56	16,49	37,71
	0,9	2,31	2,00	5,61	16,77	38,03
LCDA-GRASP 2	0,1	3,97	5,39	8,75	21,32	40,58
	0,3	2,94	3,93	7,73	17,90	44,03
	0,5	2,90	2,91	6,51	16,38	40,40
	0,7	2,40	2,33	5,70	15,25	36,76
	0,9	2,20	1,91	4,00	14,78	34,47

Tabela 11 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Books about US politics*.

Na Tabela 12, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 12, tem-se que as combinações,  $\alpha_s = 0,5$  com  $\alpha_c \geq 0,7$  e  $\alpha_s = 0,3$  com quaisquer valores de  $\alpha_c$  para o algoritmo LCDA-GRASP 1, apresentam as maiores modularidades médias. O mesmo comportamento é observado no algoritmo LCDA-GRASP 2, com a adição da combinação  $\alpha_c = 0,3$  e  $\alpha_s = 0,1$ , que também proporciona uma maior modularidade média.

Method	$\alpha_c$	$\alpha_s$					Média	Decision
		0,1	0,3	0,5	0,7	0,9		
LCDA-GRASP 1	0,1	<b>0,5272</b>	<b>0,5272</b>	0,5264	0,5228	0,5197	0,5247	ab
	0,3	<b>0,5272</b>	<b>0,5272</b>	0,5269	0,5238	0,5207	0,5252	a
	0,5	0,5271	<b>0,5272</b>	0,5271	0,5238	0,5212	0,5253	ab
	0,7	0,5270	<b>0,5272</b>	<b>0,5272</b>	0,5234	0,5188	0,5247	ab
	0,9	0,5269	<b>0,5272</b>	<b>0,5272</b>	0,5227	0,5180	0,5244	b
	Média	0,5271	<b>0,5272</b>	0,5270	0,5233	0,5197	0,5248	
	Decision	A	B	C	D	E		
LCDA-GRASP 2	0,1	<b>0,5272</b>	<b>0,5272</b>	0,5261	0,5213	0,5163	0,5236	a
	0,3	<b>0,5272</b>	<b>0,5272</b>	0,5253	0,5228	0,5164	0,5238	a
	0,5	<b>0,5272</b>	<b>0,5272</b>	0,5269	0,5234	0,5167	0,5243	ab
	0,7	0,5269	<b>0,5272</b>	<b>0,5272</b>	0,5246	0,5159	0,5244	ab
	0,9	0,5269	<b>0,5272</b>	<b>0,5272</b>	0,5252	0,5175	0,5248	b
	Média	0,5217	<b>0,5272</b>	0,5266	0,5235	0,5166	0,5242	
	Decision	A	B	C	D	E		

Tabela 12 – Modularidade média obtida pelos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Books about US politics*.

A partir desses resultados, temos que a combinação  $\alpha_c = 0,9$  e  $\alpha_s = 0,3$  fornece bons resultados dos algoritmos em relação a modularidade média e máxima, assim como para os tempos de execução médios de ambos os algoritmos.

### *Political Blogs*

As Tabelas 13 e 15 apresentam as modularidades máximas e média, respectivamente obtidas pelos algoritmos LCDA-GRASP 1 e 2 em relação aos parâmetros  $\alpha_c$  e  $\alpha_s$ . Vale ressaltar que os resultados obtidos pelos algoritmos LCDA-GRASP 1 e 2, para instância em questão, são superiores que os obtidos e apresentados por Ahajjam, Haddad e Badir (2018), em que a modularidade máxima obtida por seus algoritmos é de 0,4062. Analisando a Tabela 13 pode-se destacar que, para o algoritmo LCDA-GRASP 1, as combinações de  $\alpha_s = 0, 1$  com quais valores de  $\alpha_c$  proporcionaram, de modo geral, as maiores modularidades máximas. O mesmo comportamento pode ser observado no algoritmo LCDA-GRASP 2, com a inclusão da combinação de  $\alpha_c = 0,7$  e  $\alpha_s = 0,3$ . A Tabela 14 apresenta o tempo médio de execução dos algoritmos propostos para diferentes valores dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede social *Political Blogs*. Nota-se que, conforme os valores de  $\alpha_s$  aumentam, maiores são os tempos de execução médios de ambos os algoritmos. Além disto, conforme aumentamos o valor de  $\alpha_c$ , menor se torna o tempo

de execução de ambos os algoritmos. De modo geral, nota-se, que o algoritmo LCDA-GRASP 2 possuem tempos de execuções menores que o algoritmo LCDA-GRASP 1, principalmente, conforme o valor de  $\alpha_s$  cresce.

Method	$\alpha_c$	$\alpha_s$				
		0,1	0,3	0,5	0,7	0,9
LCDA-GRASP 1	0,1	0,4321	0,4318	0,4317	0,4315	0,4311
	0,3	0,4321	0,4317	0,4318	0,4315	0,4314
	0,5	0,4322	0,4315	0,4315	0,4316	0,4314
	0,7	0,4321	0,4316	0,4317	0,4316	0,4316
	0,9	0,4319	0,4314	0,4316	0,4318	0,4316
LCDA-GRASP 2	0,1	0,4321	0,4316	0,4314	0,4317	0,4315
	0,3	0,4321	0,4319	0,4314	0,4314	0,4311
	0,5	0,4321	0,4318	0,4315	0,4315	0,4314
	0,7	0,4321	0,4321	0,4308	0,4315	0,4316
	0,9	0,4320	0,4317	0,4318	0,4317	0,4315

Tabela 13 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Political Blogs*.

Method	$\alpha_c$	$\alpha_s$				
		0,1	0,3	0,5	0,7	0,9
LCDA-GRASP 1	0,1	60,12	188,35	339,15	865,69	1971,10
	0,3	44,00	150,22	267,84	678,17	1767,72
	0,5	29,51	76,98	182,47	581,70	1714,75
	0,7	24,93	55,55	105,52	400,77	1337,63
	0,9	24,24	38,65	49,72	91,93	159,15
LCDA-GRASP 2	0,1	61,76	180,38	316,99	826,52	1856,80
	0,3	43,29	144,91	238,41	656,43	1682,30
	0,5	27,15	87,31	182,85	550,17	1425,08
	0,7	20,34	43,54	89,00	332,81	858,05
	0,9	24,35	36,24	45,00	63,96	84,07

Tabela 14 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Political Blogs*.

Na Tabela 15, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 15, tem-se que as combinações  $\alpha_s = 0, 1$  com quaisquer valores de  $\alpha_c$  apresentam as maiores modularidades médias, em ambos os algoritmos. Por outro lado,  $\alpha_c = 0, 9$ , em ambos os algoritmos, apresenta as maiores modularidades médias.

Method	$\alpha_s$					$Média$	Decision	
	$\alpha_c$	0,1	0,3	0,5	0,7			0,9
LCDA-GRASP 1	0,1	0,4319	0,4311	0,4304	0,4295	0,4292	0,4304	a
	0,3	0,4319	0,4312	0,4305	0,4297	0,4287	0,4304	a
	0,5	0,4319	0,4311	0,4300	0,4303	0,4301	0,4307	b
	0,7	0,4319	0,4311	0,4311	0,4311	0,4311	0,4313	c
	0,9	0,4317	0,4310	0,4312	0,4313	0,4312	0,4313	c
	$Média$	0,4318	0,4311	0,4306	0,4304	0,4301	0,4308	
	Decision	A	B	C	D	E		
LCDA-GRASP 2	0,1	0,4319	0,4310	0,4301	0,4294	0,4295	0,4304	a
	0,3	0,4319	0,4309	0,4306	0,4299	0,4292	0,4305	a
	0,5	0,4320	0,4313	0,4304	0,4303	0,4308	0,4310	b
	0,7	0,4320	0,4317	0,4291	0,4309	0,4311	0,4309	b
	0,9	0,4316	0,4313	0,4312	0,4312	0,4311	0,4313	c
	$Média$	0,4319	0,4313	0,4303	0,4303	0,4303	0,4308	
	Decision	A	B	C	C	C		

Tabela 15 – Modularidade média obtida pelos algoritmos propostos para cada combinação dos parâmetros  $\alpha_c$  e  $\alpha_s$  para a rede *Political Blogs*.

Baseando-se nesses resultados, temos que a combinação  $\alpha_c = 0,9$  e  $\alpha_s = 0,1$  fornece bons resultados dos algoritmos em relação a modularidade média e máxima, assim como para os tempos de execução médios de ambos os algoritmos.

As Figuras 1, 2, 3, 4 e 5 apresentam os resultados dos agrupamentos obtidos por ambos algoritmos para a redes sociais *Zachary's karate club*, *American college football*, *Dolphin social network*, *Books about US politics* e *Political Blogs*, respectivamente. Nos gráficos, nós de mesma cor, representam uma comunidade, e os nós líderes são representados por círculos maiores. Observando as Figuras 1, 2, 3, 4 e 5 temos que o número de grupos detectados, em ambos os algoritmos, são de 4, 10, 5, 5, 5 grupos para *Zachary's karate club*, *American college football*, *Dolphin social network*, *Books about US politics* e *Political Blogs*, respectivamente.

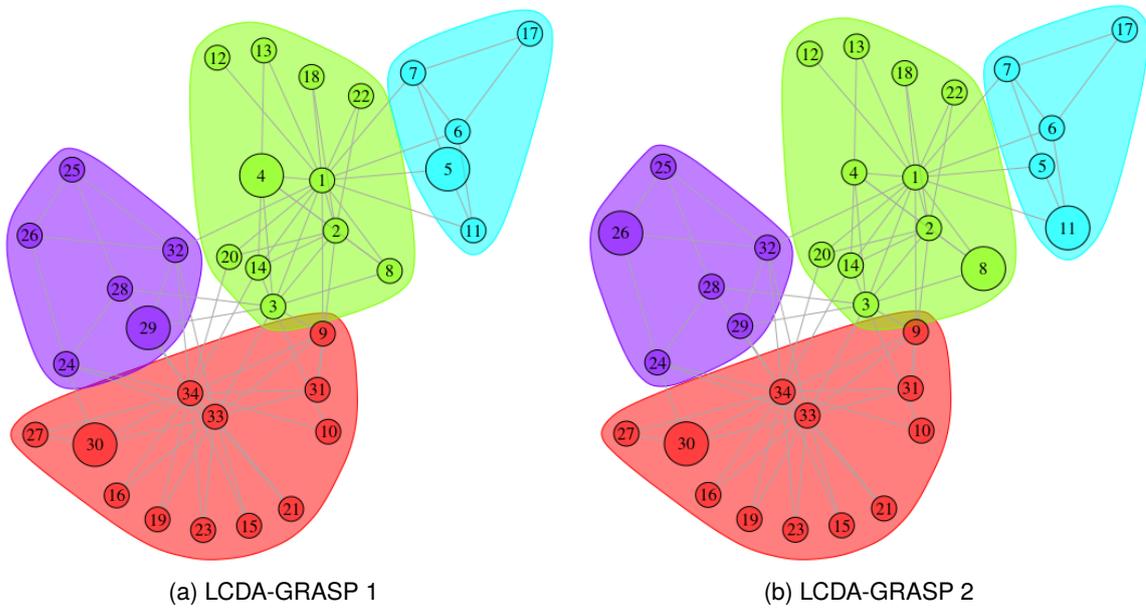


Figura 1 – Líderes e comunidades detectadas para a rede social *Zachary's Karate Club*.

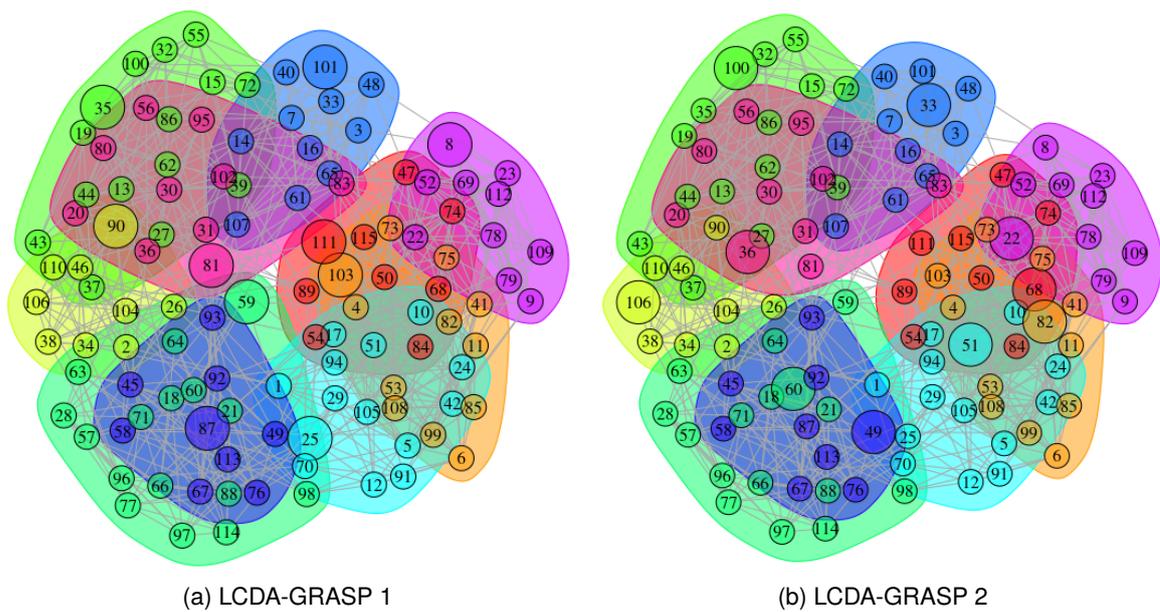


Figura 2 – Líderes e comunidades detectadas para a rede social *American college football*.

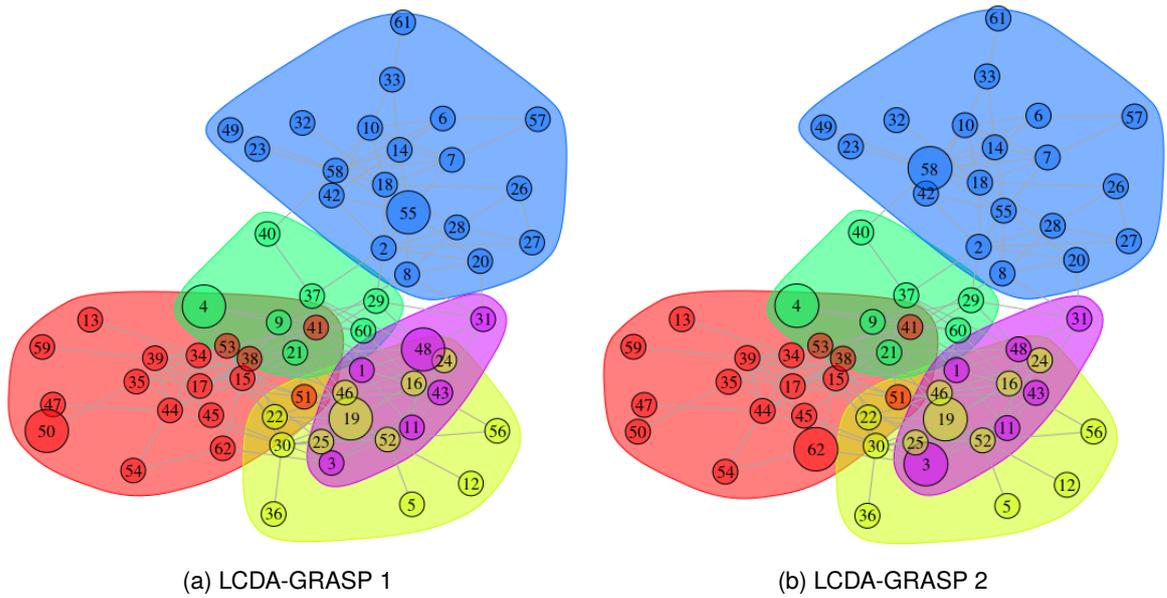


Figura 3 – Líderes e comunidades detectadas para a rede social *Dolphin social network*.

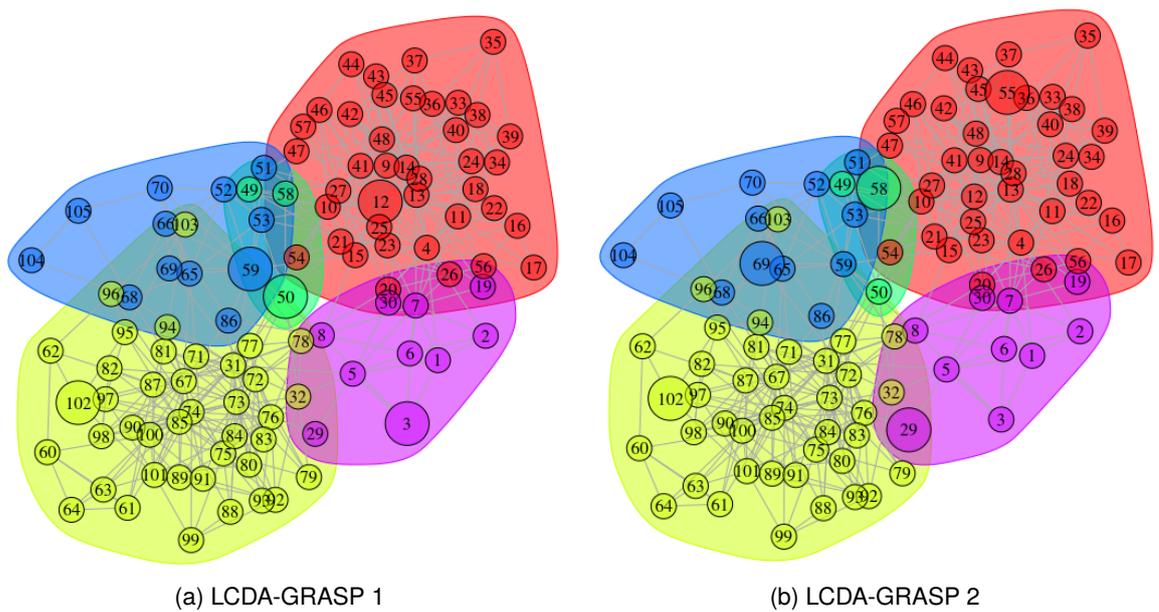


Figura 4 – Líderes e comunidades detectadas para a rede social *Books about US politics*.

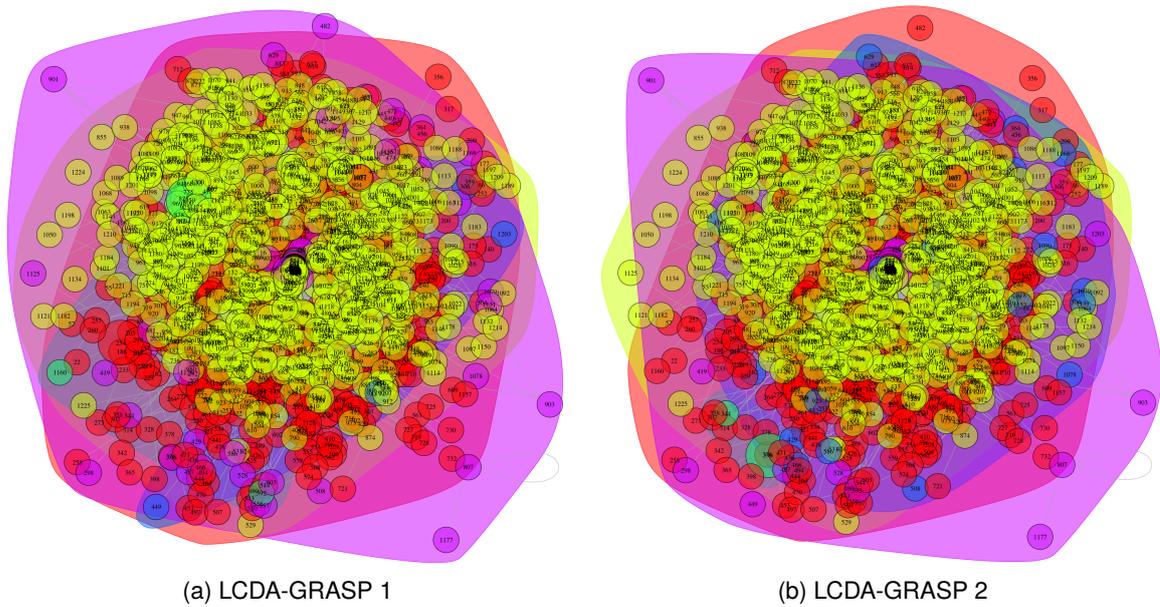


Figura 5 – Líderes e comunidades detectadas para a rede social *Political Blogs*.

### 3.3.1.2 Estudo de Simulação 2

Os resultados apresentados dizem respeito a 30 execuções independentes dos algoritmos LCDA-GRASP 1 e 2 para cada instância foi definido um valor para os parâmetros do GRASP  $\alpha_c$  e  $\alpha_s$ , e utilizou-se  $B = 50$  (número de iterações). Os valores de  $\alpha_c$  e  $\alpha_s$  foram definidos empiricamente, para cada instância, a partir dos resultados apresentados na Seção anterior, em que foram observados os parâmetros  $\alpha_c$  e  $\alpha_s$  que apresentaram as maiores medidas de Modularidade máxima e média, assim como aqueles que apresentaram menores tempos de processamento menor. Utilizou-se como parâmetros todas as combinações das medidas de centralidade (de autovetor, de proximidade e de intermediação) e similaridade (HPI, Jaccard e Dice). Na avaliação das soluções geradas pelos algoritmos LCDA-GRASP 1 e 2 além da modularidade, foram computados o tempo de processamento (*tempo*, em segundos).

#### *Zachary's Karate Club*

As Tabelas 16 e 18 apresentam as modularidades máximas e médias, respectivamente, obtidas pelos algoritmos LCDA-GRASP 1 e 2, utilizando  $\alpha_c = 0,1$  e  $\alpha_s = 0,3$ , para diferentes escolhas das funções de centralidade e similaridade. As medidas de modularidade máximas e médias que atingiram o valor 0,4198 foram destacadas em negrito.

Analisando a Tabela 16 pode-se destacar que todas as combinações das funções de similaridade e centralidade utilizadas atingiram o valor 0,4198 em ambos os algoritmos. A Tabela 17 apresenta o tempo médio de execução dos algoritmos propostos para diferentes combinações das funções de centralidade e similaridade para a rede social *Zachary's Karate Club*. Nota-se que, para ambos os algoritmos, as similaridade Dice e HPI, proporcionaram os menores tempos de execução médios em ambos os algoritmos.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GRASP 1	Intermediação	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	Proximidade	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	Autovalor	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
LCDA-GRASP 2	Intermediação	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	Proximidade	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	Autovalor	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>

Tabela 16 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Zachary's Karate Club*.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GRASP 1	Intermediação	0,50	0,47	0,52
	Proximidade	0,47	0,44	0,55
	Autovalor	0,44	0,45	0,56
LCDA-GRASP 2	Intermediação	0,48	0,46	0,56
	Proximidade	0,47	0,46	0,54
	Autovalor	0,47	0,45	0,54

Tabela 17 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Zachary's Karate Club*.

Na Tabela 18, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 18, tem-se que o uso da similaridade Dice fornece para qualquer combinação das funções de centralidade a maior média para ambos os algoritmos LCDA-GRASP 1 e 2. Pode-se observar que, para o algoritmo LCDA-GRASP 1, o uso da função de centralidade de proximidade e de autovalor combinados com qualquer função de similaridade fornece a maior média observada. Para o algoritmo LCDA-GRASP 2, a combinação da função de centralidade de autovetor combinado com qualquer função de similaridade também fornece a maior média.

Method	Similaridade				Média	Decision
	Centralidade	Dice	HPI	Jaccard		
LCDA-GRASP 1	Intermediação	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	a
	Proximidade	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	a
	Autovalor	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	a
	Média	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	
	Decision	A	A	A		

	Intermediação	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	a
	Proximidade	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	a
LCDA-GRASP 2	Autovalor	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	a
	<i>Média</i>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	
	Decision	A	A	A		

Tabela 18 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Zachary's Karate Club*.

Desta forma, o uso das funções de centralidade de autovalor e similaridade Dice proporcionam ótimas medidas de modularidade assim como um tempo computacional menor de execução, utilizando ambos os algoritmos, para a rede *Zachary's Karate Club*.

#### *American College football*

As Tabelas 19 e 21 apresentam as modularidades máximas e média, respectivamente obtidas pelos algoritmos LCDA-GRASP 1 e 2 para diferentes escolhas das funções de centralidade e similaridade. As medidas de modularidade máximas e médias que atingiram o valor 0,6046 foram destacadas em negrito.

Analisando a Tabela 16 pode-se destacar que para o algoritmo LCDA-GRASP 1, todas as combinações das funções de similaridade Dice e HPI com as centralidade de Proximidade e Autovalor utilizadas atingiram o valor 0,6046. O mesmo comportamento é observado no algoritmo LCDA-GRASP 2 com a inclusão das combinações de função de similaridade Jaccard e centralidade de proximidade e a similaridade HPI com a centralidade de intermediação que também atingiram a modularidade máxima. A Tabela 20 apresenta o tempo médio de execução dos algoritmos propostos para diferentes combinações das funções de centralidade e similaridade para a rede social *American College football*. Nota-se que, a função de similaridade Jaccard a função de centralidade de autovalor apresentam os menores tempo de execução médio dentre as demais similaridades, para ambos os algoritmos.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GRASP 1	Intermediação	0,5892	0,5905	0,5886
	Proximidade	<b>0,6046</b>	<b>0,6046</b>	0,6032
	Autovalor	<b>0,6046</b>	<b>0,6046</b>	0,6044
LCDA-GRASP 2	Intermediação	0,6032	<b>0,6046</b>	0,6032
	Proximidade	<b>0,6046</b>	<b>0,6046</b>	<b>0,6046</b>
	Autovalor	<b>0,6046</b>	<b>0,6046</b>	0,6044

Tabela 19 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *American College football*.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GRASP 1	Intermediação	6,99	7,62	6,20
	Proximidade	7,08	7,20	6,05
	Autovalor	4,81	5,61	3,90
LCDA-GRASP 2	Intermediação	6,88	6,93	6,20
	Proximidade	5,92	6,08	5,11
	Autovalor	4,44	5,02	3,07

Tabela 20 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *American College football*.

Na Tabela 21, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 21, tem-se as similaridades HPI e Dice apresentam, para as combinações das funções de centralidade de autovalor e proximidade, as maiores médias da modularidade para o o algoritmo LCDA-GRASP 2. Para o algoritmo LCDA-GRASP 1, a combinação da função de centralidade de autovalor com as similaridade Dice e HPI apresentam as maiores modularidades médias.

Method	Similaridade				Média	Decision
	Centralidade	Dice	HPI	Jaccard		
LCDA-GRASP 1	Intermediação	0,5888	0,5901	0,5877	0,5889	a
	Proximidade	0,6034	0,6040	0,6032	0,6036	b
	Autovalor	<b>0,6046</b>	<b>0,6046</b>	0,6034	0,6042	c
	Média	0,5989	0,5996	0,5981	0,5987	
	Decision	A	A	B		
LCDA-GRASP 2	Intermediação	0,6017	0,6035	0,6021	0,6024	a
	Proximidade	<b>0,6046</b>	<b>0,6046</b>	0,6034	0,6042	b
	Autovalor	<b>0,6046</b>	<b>0,6046</b>	0,6034	0,6042	b
	Média	0,6036	0,6042	0,6030	0,6036	
	Decision	A	B	C		

Tabela 21 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *American College football*.

---

Desta forma, para o algoritmo LCDA-GRASP 1, o uso das funções de similaridade HPI e centralidade de autovalor proporcionam as melhores medidas de modularidade, além de apresentar baixos valores de tempo de execução. Para o algoritmo LCDA-GRASP 2, a combinação das funções de similaridade Dice e centralidade de autovalor proporciona, dentre as demais combinações, as melhores medidas de classificação e menores tempos de execução.

#### *Dolphin social network*

As Tabelas 22 e 24 apresentam as modularidades máximas e média, respectivamente, obtidas pelos algoritmos LCDA-GRASP 1 e 2 para diferentes escolhas das funções de centralidade e similaridade. As medidas de modularidade máximas e médias que atingiram o valor 0,5285 foram destacadas em negrito.

Analisando a Tabela 22 pode-se observar que, a combinação da similaridade Dice com qualquer função centralidade, a centralidade de autovalor com quaisquer função de similaridade e a centralidade de proximidade combinada a similaridade HPI apresentam as maiores modularidades, para o algoritmo LCDA-GRASP 1. Para o algoritmo LCDA-GRASP 2, temos que todas as combinações das funções de similaridade e centralidade apresentaram as maiores modularidades observadas. A Tabela 23 apresenta o tempo médio de execução dos algoritmos propostos para diferentes combinações das funções de centralidade e similaridade para a rede social *Dolphin social network*. Pode-se observar que a função de similaridade Dice apresenta os menores tempo de execução médio dentre as demais similaridades, para ambos os algoritmos. O mesmo comportamento é observado nos resultados para a centralidade de autovalor. Em particular, a combinação da função de similaridade Dice com a centralidade de autovalor e centralidade de intermediação proporcionaram os menores tempos de execução dentre as demais combinação para os algoritmos LCDA-GRASP 1 e 2.

Method	Centralidade	Similaridade		
		Dice	HPI	Jaccard
LCDA-GRASP 1	Intermediação	<b>0,5285</b>	0,5277	0,5269
	Proximidade	<b>0,5285</b>	<b>0,5285</b>	0,5276
	Autovalor	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>
LCDA-GRASP 2	Intermediação	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>
	Proximidade	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>
	Autovalor	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>

Tabela 22 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Dolphin social network*.

Method	Centralidade	Similaridade		
		Dice	HPI	Jaccard
LCDA-GRASP 1	Intermediação	1,92	2,55	1,91
	Proximidade	1,83	1,99	2,03
	Autovalor	1,54	1,89	1,74
LCDA-GRASP 2	Intermediação	1,78	2,20	1,79
	Proximidade	1,89	2,04	2,05
	Autovalor	1,56	1,79	1,78

Tabela 23 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Dolphin social network*.

Na Tabela 24, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 24, tem-se que o uso da similaridade HPI fornece para qualquer combinação das funções de centralidade as maiores médias para os algoritmos LCDA-GRASP 1 e 2. Além disto, pode-se destacar que a combinação da centralidade de intermediação com a similaridade HPI, em ambos os algoritmos, apresentou a maior média dentre as demais combinações.

Method	Centralidade	Similaridade			Média	Decision
		Dice	HPI	Jaccard		
LCDA-GRASP 1	Intermediação	0,5266	0,5274	0,5226	0,5255	a
	Proximidade	0,5263	0,5271	0,5238	0,5257	a
	Autovalor	0,5262	0,5266	0,5255	0,5261	a
	Média	0,5264	0,5270	0,5240	0,5258	
	Decision	A	B	C		
LCDA-GRASP 2	Intermediação	0,5269	0,5276	0,5270	0,5272	a
	Proximidade	0,5251	0,5274	0,5244	0,5256	b
	Autovalor	0,5254	0,5268	0,5255	0,5259	b
	Média	0,5258	0,5273	0,5256	0,5263	
	Decision	A	B	A		

Tabela 24 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Dolphin social network*.

Portanto, para os algoritmo LCDA-GRASP 1 e 2, o uso da função de similaridade HPI com

a centralidade de intermediação proporcionou, dentre as demais métricas, os melhores valores de modularidades para a rede *Dolphin social network*.

### *Books about US politics*

As Tabelas 25 e 27 apresentam as modularidades máximas e média, respectivamente obtidas pelos algoritmos LCDA-GRASP 1 e 2 para diferentes escolhas das funções de centralidade e similaridade. As medidas de modularidade máximas e médias que atingiram o valor 0,5285 foram destacadas em negrito.

Analisando a Tabela 25 pode-se observar que todas as combinações das funções de similaridade e centralidade, para ambos os algoritmos propostos, a maior modularidade observada é de 0,5272. Pode-se observar que, com exceção da combinação da centralidade de autovalor com a similaridade Jaccard, apresentou as maiores modularidades máximas para o algoritmo LCDA-GRASP 2. Para o algoritmo LCDA-GRASP 1, as combinações da função de centralidade de intermediação e autovalor com as similaridades Dice e HPI e a combinação da centralidade de proximidade com a similaridade Jaccard, proporcionaram as maiores modularidades máximas observadas. A Tabela 26 apresenta o tempo médio de execução dos algoritmos propostos para diferentes combinações das funções de centralidade e similaridade para a rede social *Books about US politics*. Observa-se que, a similaridade HPI proporciona os menores tempo de execução médio dentre as demais similaridades, para ambos os algoritmos. O mesmo comportamento é observado ao se escolher a centralidade de autovalor.

Method	Centralidade	Similaridade		
		Dice	HPI	Jaccard
LCDA-GRASP 1	Intermediação	<b>0,5272</b>	<b>0,5272</b>	0,5256
	Proximidade	0,5241	0,5269	<b>0,5272</b>
	Autovalor	<b>0,5272</b>	<b>0,5272</b>	0,5241
LCDA-GRASP 2	Intermediação	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>
	Proximidade	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>
	Autovalor	<b>0,5272</b>	<b>0,5272</b>	0,5256

Tabela 25 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Books about US politics*.

Method	Centralidade	Similaridade		
		Dice	HPI	Jaccard
LCDA-GRASP 1	Intermediação	5,71	4,62	7,33
	Proximidade	5,21	4,95	8,13
	Autovalor	3,95	1,87	6,32
LCDA-GRASP 2	Intermediação	5,64	4,53	7,11
	Proximidade	5,11	4,58	7,06
	Autovalor	3,71	1,82	5,67

Tabela 26 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Books about US politics*.

Na Tabela 27, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 27, tem-se que para as combinações da métrica de centralidade de autovalor com as similaridades HPI e Dice, para ambos os algoritmos, proporcionam as maiores modularidades médias. Além disto, a similaridade HPI proporciona uma maior modularidade média em relação as demais similaridades, para ambos os algoritmos.

Method	Similaridade				Média	Decision
	Centralidade	Dice	HPI	Jaccard		
LCDA-GRASP 1	Intermediação	0,5247	0,5271	0,5240	0,5252	a
	Proximidade	0,5239	0,5269	0,5242	0,5250	a
	Autovalor	<b>0,5272</b>	<b>0,5272</b>	0,5239	0,5261	b
	Média	0,5253	0,5271	0,5240	0,5254	
	Decision	A	B	C		
LCDA-GRASP 2	Intermediação	0,5269	0,5270	0,5257	0,5265	a
	Proximidade	0,5266	0,5270	0,5260	0,5265	a
	Autovalor	<b>0,5272</b>	<b>0,5272</b>	0,5250	0,5265	a
	Média	0,5269	0,5271	0,5255	0,5265	
	Decision	A	B	C		

Tabela 27 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Books about US politics*.

Desta forma, para os algoritmo LCDA-GRASP 1 e 2, o uso da função de similaridade HPI com a centralidade de autovalor proporcionam, dentre as demais métricas, os melhores valores de modularidades para a rede *Books about US politics* e menores tempos de execução.

### *Political Blogs*

As Tabelas 28 e 30 apresentam as modularidades máximas e média, respectivamente obtidas pelos algoritmos LCDA-GRASP 1 e 2 para diferentes escolhas das funções de centralidade e similaridade. Ressalta-se que os resultados obtidos pelos algoritmos LCDA-GRASP 1 e 2, para instância em questão, são superiores que os obtidos e apresentados por (AHAJJAM; HADDAD; BADIR, 2018), em que a modularidade máxima obtida por seus algoritmos é de 0,4062.

Observando a Tabela 28, temos que, em ambos os algoritmos, as combinações de centralidade

dade e similaridade em estudo, possuem máximos próximos, não diferindo mais que 0,0001. A Tabela 29 apresenta o tempo médio de execução dos algoritmos propostos para diferentes combinações das funções de centralidade e similaridade para a rede social *Political Blogs*. Pode-se observar que a função de similaridade Dice combinada a qualquer função de centralidade apresentam os menores tempo de execução médio dentre as demais similaridades, para ambos os algoritmos. Em relação a função de centralidade, podemos destacar, para o algoritmo LCDA-GRASP 1, a centralidade de intermediação apresenta os menores tempos de execução médio combinado com qualquer função de similaridade, o mesmo comportamento pode ser observado para a centralidade de autovalor para o algoritmo LCDA-GRASP 2. Pode-se destacar a combinação da similaridade Dice com a centralidade de intermediação, para o algoritmo LCDA-GRASP 1, e centralidade de autovalor, para o algoritmo LCDA-GRASP 2, apresentam os menores tempos computacionais médias para as demais combinações em estudo.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GRASP 1	Intermediação	0,4319	0,4320	0,4320
	Proximidade	0,4320	0,4319	0,4319
	Autovalor	0,4320	0,4320	0,4320
LCDA-GRASP 2	Intermediação	0,4319	0,4321	0,4319
	Proximidade	0,4319	0,4320	0,4320
	Autovalor	0,4320	0,4319	0,4317

Tabela 28 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Political Blogs*.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GRASP 1	Intermediação	25,33	27,13	26,12
	Proximidade	56,72	66,71	59,04
	Autovalor	24,93	25,31	26,06
LCDA-GRASP 2	Intermediação	26,36	27,12	25,68
	Proximidade	55,95	74,76	60,41
	Autovalor	24,67	25,01	23,86

Tabela 29 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Political Blogs*.

Na Tabela 30, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 30, tem-se que a similaridade HPI, em ambos os algoritmos, proporciona as maiores modularidades médias. Por outro lado, as centralidades de proximidade e autovalor, em ambos os algoritmos, apresentam as maiores modularidades médias, em relação as demais centralidades.

Method	Similaridade					Média	Decision
	Centralidade	Dice	HPI	Jaccard			
LCDA-GRASP 1	Intermediação	0,4315	0,4314	0,4314	0,4314	0,4314	a
	Proximidade	0,4317	0,4317	0,4316	0,4316	0,4317	b

	Autovalor	0,4316	0,4318	0,4315	0,4316	c
	<i>Média</i>	0,4316	0,4316	0,4315	0,4315	
	Decision	A	A	B		
	Intermediação	0,4305	0,4316	0,4314	0,4311	a
	Proximidade	0,4317	0,4317	0,4317	0,4317	b
LCDA-GRASP 2	Autovalor	0,4315	0,4316	0,4314	0,4315	b
	<i>Média</i>	0,4312	0,4316	0,4315	0,4314	
	Decision	A	B	B		

Tabela 30 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Political Blogs*.

Portanto, para os algoritmos LCDA-GRASP 1 e 2, o uso da função de similaridade HPI com a centralidade de autovalor proporcionam, dentre as demais métricas, os melhores valores de modularidades para a rede *Political Blogs* e os menores tempos de execução dos algoritmos.

A Tabela 31 apresenta uma comparação entre as modularidades obtidas pelos algoritmos LCDA-GRASP 1 e 2 e os algoritmos da literatura LCDA 1 e 2 (AHAJJAM; HADDAD; BADIR, 2018), *LeadersRank* 1 e 2 (AHAJJAM; HADDAD; BADIR, 2015) ao serem aplicadas nas instâncias trabalhadas nesta Seção. Analisando a Tabela 31 podemos observar que, para a rede social *Zachary's karate club*, as modularidades obtidas pelos algoritmos LCDA-GRASP 1 e 2 são maiores que as obtidas pelos algoritmos LCDA 1 e 2 e *LeaderRank* 1 e 2. O mesmo pode-se afirmar para a rede social *American college football*, em que a modularidade obtida por ambos os algoritmos propostos superam as modularidades obtidas pelos algoritmos LCDA 1 e 2 e *LeaderRank* 1 e 2. De fato, as modularidades obtidas para ambas as instâncias pelos algoritmos propostos corresponde ao valor ótimo reportado na literatura, conforme pode ser visto em Agarwal e Kempe (2008).

Algoritmos/Redes Sociais	<i>Zachary's karate club</i>	<i>American College football</i>	<i>Dolphin social network</i>	<i>Books about US politics</i>	<i>Political Blogs</i>
<i>LeaderRank</i> 1	0,3180	0,3780	-	-	0,2359
LCDA 1	0,3266	0,5162	-	-	0,3977
LCDA-GRASP 1	<b>0,4198</b>	<b>0,6046</b>	<b>0,5285</b>	<b>0,5272</b>	<b>0,4322</b>
<i>LeaderRank</i> 2	0,3014	0,3750	-	-	0,2388
LCDA 2	0,3266	0,4842	-	-	0,4062
LCDA-GRASP 2	<b>0,4198</b>	<b>0,6046</b>	<b>0,5285</b>	<b>0,5272</b>	<b>0,4321</b>

Tabela 31 – Comparação entre modularidades dos algoritmos propostos e da literatura.

### 3.3.2 LCDA-GR

Os algoritmos propostos foram implementados usando o software estatístico R (R Core Team, 2018) e compilado na versão 8.2.0. Os testes foram realizados em uma máquina virtual no Google Cloud com 20 *HighCPU* núcleos e 18GB de memória RAM rodando Ubuntu 18.04.

Foram realizados dois estudos de simulação a fim de avaliar o desempenho dos algoritmos. O primeiro é um estudo extenso do desempenho dos algoritmos em relação as diferentes escolhas das métricas de centralidades e similaridades.

#### 3.3.2.1 Estudo de Simulação

Os resultados apresentados dizem respeito a 30 execuções independentes dos algoritmos LCDA-GR 1 e 2 para cada instância utilizando as métricas de centralidades e similaridades nas Equações 3, 4, 4 e 7, 6, 8, respectivamente, e  $B = 150$  (número de iterações). Na avaliação das soluções geradas pelos algoritmos LCDA-GR 1 e 2, além da modularidade, foram computados o tempo de processamento (*tempo* em segundos). Foram realizados teste de permutação para as médias da modularidade para avaliar as diferenças entre as diferentes métricas de centralidades e similaridades em estudo, uma vez que, em alguns níveis foi observado variância zero.

No teste de permutação, comparamos o valor da estatística observada com as saídas apresentadas através do algoritmo computacional, que consiste em criar re-amostras por simples permutações dos dados originais, maiores detalhes podem ser vistos, por exemplo, em (DUCZMAL et al., 2003). Se estas saídas produzem estatísticas com valores semelhantes à estatística observada nos dados, então a estatística observada terá sido um mero acaso. O teste terá validade, desde que se mantenha a estrutura dos dados. Os testes de permutação apresentam uma vantagem sobre os testes clássicos, que é não necessitar da suposição de aleatoriedade na coleta dos dados, e nem mesmo de uma suposição inicial da função de densidade da estatística em estudo (EDGINGTON, 1995).

#### *Zachary's Karate Club*

As Tabelas 32 e 34 apresentam as modularidades máximas e médias, respectivamente, obtidas pelos algoritmos LCDA-GR 1 e 2 para diferentes escolhas das funções de centralidade e similaridade. As medidas de modularidade máximas e médias que atingiram o valor 0,4198

foram destacadas em negrito.

Analisando a Tabela 32 pode-se destacar que todas as combinações das funções de similaridade e centralidade utilizadas atingiram o valor 0,4198 em ambos os algoritmos. A Tabela 33 apresenta o tempo médio de execução dos algoritmos propostos para diferentes combinações das funções de centralidade e similaridade para a rede social *Zachary's Karate Club*. Nota-se que, para ambos os algoritmos, as similaridade Dice e HPI, proporcionaram os menores tempos de execução médios em ambos os algoritmos.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GR 1	Intermediação	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	Proximidade	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	Autovalor	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
LCDA-GR 2	Intermediação	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	Proximidade	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>
	Autovalor	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>

Tabela 32 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Zachary's Karate Club*.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GR 1	Intermediação	3,56	2,83	4,07
	Proximidade	3,57	3,22	4,10
	Autovalor	3,46	3,04	3,99
LCDA-GR 2	Intermediação	3,56	2,86	4,08
	Proximidade	3,51	3,23	4,02
	Autovalor	3,53	3,00	4,01

Tabela 33 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Zachary's Karate Club*.

Na Tabela 34, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 34, tem-se que o uso da similaridade Dice fornece para qualquer combinação das funções de centralidade a maior média para ambos os algoritmos LCDA-GR 1 e 2. Pode-se observar que, para o algoritmo LCDA-GR 1, o uso da função de centralidade de proximidade e de autovalor combinados com qualquer função de similaridade fornece a maior média observada. Para o algoritmo LCDA-GR 2, a combinação da função de centralidade de autovetor combinado com qualquer função de similaridade também fornece a maior média.

Method	Similaridade				Média	Decision
	Centralidade	Dice	HPI	Jaccard		
LCDA-GR 1	Intermediação	<b>0,4198</b>	0,4192	<b>0,4198</b>	0,4196	a
	Proximidade	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	a
	Autovalor	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	a
	Média	<b>0,4198</b>	0,4196	<b>0,4198</b>	0,4197	

	Decision	A	A	A	
	Intermediação	<b>0,4198</b>	0,4181	<b>0,4198</b>	0,4192 a
	Proximidade	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b> a
LCDA-GR 2	Autovalor	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b>	<b>0,4198</b> a
	<i>Média</i>	<b>0,4198</b>	0,4192	<b>0,4198</b>	0,4196
	Decision	A	A	A	

Tabela 34 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Zachary's Karate Club*.

Desta forma, o uso das funções de centralidade de autovalor e similaridade Dice proporcionam ótimas medidas de modularidade assim como um tempo computacional menor de execução, utilizando ambos os algoritmos, para a rede *Zachary's Karate Club*.

#### *American College football*

As Tabelas 35 e 37 apresentam as modularidades máximas e média, respectivamente obtidas pelos algoritmos LCDA-GR 1 e 2 para diferentes escolhas das funções de centralidade e similaridade. As medidas de modularidade máximas e médias que atingiram o valor 0,6046 foram destacadas em negrito.

Analisando a Tabela 32 pode-se destacar que para o algoritmo LCDA-GR 1, todas as combinações das funções de similaridade Dice e HPI com as centralidade de Proximidade e Autovalor utilizadas atingiram o valor 0,6046. O mesmo comportamento é observado no algoritmo LCDA-GR 2 com a inclusão das combinações de função de similaridade Jaccard e centralidade de proximidade e a similaridade HPI com a centralidade de intermediação que também atingiram a modularidade máxima. A Tabela 36 apresenta o tempo médio de execução dos algoritmos propostos para diferentes combinações das funções de centralidade e similaridade para a rede social *American College football*. Nota-se que, a função de similaridade Jaccard a função de centralidade de autovalor apresentam os menores tempo de execução médio dentre as demais similaridades, para ambos os algoritmos.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GR 1	Intermediação	<b>0,6046</b>	<b>0,6046</b>	<b>0,6046</b>
	Proximidade	<b>0,6046</b>	<b>0,6046</b>	<b>0,6046</b>
	Autovalor	<b>0,6046</b>	<b>0,6046</b>	<b>0,6046</b>
LCDA-GR 2	Intermediação	<b>0,6046</b>	<b>0,6046</b>	0,6044
	Proximidade	<b>0,6046</b>	<b>0,6046</b>	<b>0,6046</b>
	Autovalor	<b>0,6046</b>	<b>0,6046</b>	<b>0,6046</b>

Tabela 35 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *American College football*.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GR 1	Intermediação	41,12	42,57	39,91
	Proximidade	41,21	42,96	40,03
	Autovalor	37,18	38,79	35,57
LCDA-GR 2	Intermediação	41,75	42,25	40,42
	Proximidade	38,80	40,89	37,38
	Autovalor	33,13	35,46	31,74

Tabela 36 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *American College football*.

Na Tabela 37, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 37, tem-se as similaridades HPI e Dice apresentam, para as combinações das funções de centralidade de autovalor e proximidade, as maiores médias da modularidade para o o algoritmo LCDA-GR 2. Para o algoritmo LCDA-GR 1, a combinação da função de centralidade de autovalor com as similaridade Dice e HPI apresentam as maiores modularidades médias.

Method	Similaridade				Média	Decision
	Centralidade	Dice	HPI	Jaccard		
LCDA-GR 1	Intermediação	0,6036	0,6040	0,6032	0,6036	a
	Proximidade	0,6038	0,6040	0,6034	0,6037	b
	Autovalor	0,6042	0,6042	0,6036	0,6040	c
	Média	0,6038	0,6041	0,6034	0,6038	
	Decision	A	A	B		
LCDA-GR 2	Intermediação	0,6036	0,6039	0,6033	0,6036	a
	Proximidade	0,6041	0,6044	0,6036	0,6040	b
	Autovalor	0,6041	0,6043	0,6037	0,6040	b
	Média	0,6039	0,6042	0,6035	0,6039	
	Decision	A	B	C		

Tabela 37 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *American College football*.

Desta forma, para o algoritmo LCDA-GR 1, o uso das funções de similaridade HPI e centralidade de autovalor proporciona a melhores medidas de modularidade, além de apresentar baixos valores de tempo de execução. Para o algoritmo LCDA-GR 2, a combinação das funções de similaridade Dice e centralidade de autovalor proporciona, dentre as demais combinações, as melhores medidas de classificação e menores tempos de execução.

### *Dolphin social network*

As Tabelas 38 e 40 apresentam as modularidades máximas e média, respectivamente, obtidas pelos algoritmos LCDA-GR 1 e 2 para diferentes escolhas das funções de centralidade e similaridade. As medidas de modularidade máximas e médias que atingiram o valor 0,5285 foram destacadas em negrito.

Analisando a Tabela 38 pode-se observar que, a combinação da similaridade Dice com qualquer função centralidade, a centralidade de autovalor com quaisquer função de similaridade e a centralidade de proximidade combinada a similaridade HPI apresentam as maiores modularidades, para o algoritmo LCDA-GR 1. Para o algoritmo LCDA-GR 2, temos que todas as combinações das funções de similaridade e centralidade apresentaram as maiores modularidades observadas. A Tabela 39 apresenta o tempo médio de execução dos algoritmos propostos para diferentes combinações das funções de centralidade e similaridade para a rede social *Dolphin social network*. Pode-se observar que a função de similaridade Dice apresenta os menores tempo de execução médio dentre as demais similaridades, para ambos os algoritmos. O mesmo comportamento é observado nos resultados para a centralidade de autovalor. Em particular, a combinação da função de similaridade Dice com a centralidade de autovalor e centralidade de intermediação proporcionaram os menores tempos de execução dentre as demais combinação para os algoritmos LCDA-GR 1 e 2.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GR 1	Intermediação	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>
	Proximidade	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>
	Autovalor	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>
LCDA-GR 2	Intermediação	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>
	Proximidade	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>
	Autovalor	<b>0,5285</b>	<b>0,5285</b>	<b>0,5285</b>

Tabela 38 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Dolphin social network*.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GR 1	Intermediação	16,81	18,29	17,98
	Proximidade	15,84	16,82	16,79
	Autovalor	13,82	15,01	15,42
LCDA-GR 2	Intermediação	15,50	16,21	16,95
	Proximidade	15,77	16,22	17,06
	Autovalor	13,74	13,68	16,04

Tabela 39 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Dolphin social network*.

Na Tabela 40, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 40, tem-se que o uso da similaridade HPI fornece para qualquer combinação das funções de centralidade as maiores médias para os algoritmos LCDA-GR 1 e 2. Além disto, pode-se destacar que a combinação da centralidade de intermediação com a similaridade HPI, em ambos os algoritmos, apresentou a maior média dentre as demais combinações.

Method	Similaridade				Média	Decision
	Centralidade	Dice	HPI	Jaccard		
LCDA-GR 1	Intermediação	0,5257	0,5261	0,5235	0,5251	a
	Proximidade	0,5252	0,5266	0,5239	0,5253	a
	Autovalor	0,5267	0,5273	0,5262	0,5267	a
	Média	0,5258	0,5267	0,5245	0,5257	
	Decision	A	B	C		
LCDA-GR 2	Intermediação	0,5260	0,5272	0,5239	0,5257	a
	Proximidade	0,5249	0,5272	0,5235	0,5252	b
	Autovalor	0,5251	0,5268	0,5243	0,5254	b
	Média	0,5253	0,5271	0,5239	0,5254	
	Decision	A	B	A		

Tabela 40 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Dolphin social network*.

Portanto, para os algoritmo LCDA-GR 1 e 2, o uso da função de similaridade HPI com a

centralidade de intermediação proporcionou, dentre as demais métricas, os melhores valores de modularidades para a rede *Dolphin social network*.

### *Books about US politics*

As Tabelas 41 e 43 apresentam as modularidades máximas e média, respectivamente obtidas pelos algoritmos LCDA-GR 1 e 2 para diferentes escolhas das funções de centralidade e similaridade. As medidas de modularidade máximas e médias que atingiram o valor 0,5285 foram destacadas em negrito. Analisando a Tabela 41 pode-se observar que todas as combinações das funções de similaridade e centralidade, para ambos os algoritmos propostos, a maior modularidade observada é de 0,5272. Pode-se observar que, com exceção da combinação da centralidade de autovalor com a similaridade Jaccard, apresentou as maiores modularidades máximas para o algoritmo LCDA-GR 2. Para o algoritmo LCDA-GR 1, as combinações da função de centralidade de intermediação e autovalor com as similaridades Dice e HPI e a combinação da centralidade de proximidade com a similaridade Jaccard, proporcionaram as maiores modularidades máximas observadas. A Tabela 42 apresenta o tempo médio de execução dos algoritmos propostos para diferentes combinações das funções de centralidade e similaridade para a rede social *Books about US politics*. Observa-se que, a similaridade HPI proporciona os menores tempo de execução médio dentre as demais similaridades, para ambos os algoritmos. O mesmo comportamento é observado ao se escolher a centralidade de autovalor.

Method	Centralidade	Similaridade		
		Dice	HPI	Jaccard
LCDA-GR 1	Intermediação	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>
	Proximidade	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>
	Autovalor	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>
LCDA-GR 2	Intermediação	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>
	Proximidade	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>
	Autovalor	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>

Tabela 41 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Books about US politics*.

Method	Centralidade	Similaridade		
		Dice	HPI	Jaccard
LCDA-GR 1	Intermediação	45,77	32,59	56,96
	Proximidade	45,21	37,85	58,13
	Autovalor	40,45	28,87	53,80
LCDA-GR 2	Intermediação	45,93	35,88	57,65
	Proximidade	46,97	39,82	59,65
	Autovalor	38,60	28,33	51,15

Tabela 42 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Books about US politics*.

Na Tabela 43, letras maiúsculas iguais ou letras minúsculas iguais significam médias es-

taticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 43, tem-se que para as combinações da métrica de centralidade de autovalor com as similaridades HPI e Dice, para ambos os algoritmos, proporcionam as maiores modularidades médias. Além disto, a similaridade HPI proporciona uma maior modularidade média em relação as demais similaridades, para ambos os algoritmos.

Method	Similaridade					Decision
	Centralidade	Dice	HPI	Jaccard	Média	
LCDA-GR 1	Intermediação	0,5267	<b>0,5272</b>	0,5269	0,5269	a
	Proximidade	<b>0,5272</b>	0,5271	0,5271	0,5271	a
	Autovalor	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>	0,5272	b
	Média	0,5270	0,5272	0,5271	0,5271	
	Decision	A	B	C		
LCDA-GR 2	Intermediação	0,5271	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>	a
	Proximidade	0,5270	<b>0,5272</b>	0,5272	0,5271	a
	Autovalor	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>	<b>0,5272</b>	a
	Média	0,5271	<b>0,5272</b>	0,5271	<b>0,5272</b>	
	Decision	A	B	C		

Tabela 43 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Books about US politics*.

Desta forma, para os algoritmos LCDA-GR 1 e 2, o uso da função de similaridade HPI com a centralidade de autovalor proporcionam, dentre as demais métricas, os melhores valores de modularidades para a rede *Books about US politics* e menores tempos de execução.

### *Political Blogs*

As Tabelas 44 e 46 apresentam as modularidades máximas e média, respectivamente obtidas pelos algoritmos LCDA-GR 1 e 2 para diferentes escolhas das funções de centralidade e similaridade. Ressalta-se que os resultados obtidos pelos algoritmos LCDA-GR 1 e 2, para instância em questão, são superiores que os obtidos e apresentados por Ahajjam, Haddad e Badir (2018), em que a modularidade máxima obtida por seus algoritmos é de 0,4062.

Observando a Tabela 44, temos que, em ambos os algoritmos, as combinações de centralidade e similaridade em estudo, possuem máximos próximos, não diferindo mais que 0,0001. A

Tabela 45 apresenta o tempo médio de execução dos algoritmos propostos para diferentes combinações das funções de centralidade e similaridade para a rede social *Political Blogs*. Pode-se observar que a função de similaridade Dice combinada a qualquer função de centralidade apresentam os menores tempo de execução médio dentre as demais similaridades, para ambos os algoritmos. Em relação a função de centralidade, podemos destacar, para o algoritmo LCDA-GR 1, a centralidade de intermediação apresenta os menores tempos de execução médio combinado com qualquer função de similaridade, o mesmo comportamento pode ser observado para a centralidade de autovalor para o algoritmo LCDA-GR 2. Pode-se destacar a combinação da similaridade Dice com a centralidade de intermediação, para o algoritmo LCDA-GR 1, e centralidade de autovalor, para o algoritmo LCDA-GR 2, apresentam os menores tempos computacionais médias para as demais combinações em estudo.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GR 1	Intermediação	0,4266	0,4269	0,4266
	Proximidade	0,4266	0,4267	0,4266
	Autovalor	0,4266	0,4270	0,4264
LCDA-GR 2	Intermediação	0,4266	0,4269	0,4265
	Proximidade	0,4266	0,4267	0,4266
	Autovalor	0,4266	0,4268	0,4266

Tabela 44 – Modularidade máxima obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Political Blogs*.

Method	Similaridade			
	Centralidade	Dice	HPI	Jaccard
LCDA-GR 1	Intermediação	424,31	305,04	479,39
	Proximidade	830,73	824,63	976,87
	Autovalor	601,67	587,23	713,70
LCDA-GR 2	Intermediação	475,57	308,09	553,63
	Proximidade	855,45	1016,07	1039,13
	Autovalor	531,61	601,08	640,85

Tabela 45 – Tempo médio de execução (em segundos) dos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Political Blogs*.

Na Tabela 46, letras maiúsculas iguais ou letras minúsculas iguais significam médias estatisticamente iguais ao nível de 5% de significância. Em relação as médias, apresentadas na Tabela 46, tem-se que a similaridade HPI, em ambos os algoritmos, proporciona as maiores modularidades médias. Por outro lado, as centralidades de proximidade e autovalor, em ambos os algoritmos, apresentam as maiores modularidades médias, em relação as demais centralidades.

Method	Similaridade					Média	Decision
	Centralidade	Dice	HPI	Jaccard			
LCDA-GR 1	Intermediação	0,4260	0,4266	0,4260	0,4260	0,4262	a
	Proximidade	0,4262	0,4263	0,4260	0,4260	0,4262	b
	Autovalor	0,4260	0,4265	0,4258	0,4258	0,4261	c

	<i>Média</i>	0,4261	0,4265	0,4259	0,4262	
	Decision	A	A	B		
LCDA-GR 2	Intermediação	0,4261	0,4265	0,4260	0,4262	a
	Proximidade	0,4261	0,4262	0,4260	0,4261	b
	Autovalor	0,4259	0,4265	0,4259	0,4261	b
	<i>Média</i>	0,4260	0,4264	0,4259	0,4261	
	Decision	A	B	B		

Tabela 46 – Modularidade média obtida pelos algoritmos propostos para cada combinação das funções de centralidade e similaridade para a rede *Political Blogs*.

Portanto, para os algoritmos LCDA-GR 1 e 2, o uso da função de similaridade HPI com a centralidade de autovalor proporcionam, dentre as demais métricas, os melhores valores de modularidades para a rede *Political Blogs* e os menores tempos de execução dos algoritmos.

A Tabela 47 apresenta uma comparação entre as modularidades obtidas pelos algoritmos LCDA-GR 1 e 2 e os algoritmos da literatura LCDA 1 e 2 (AHAJJAM; HADDAD; BADIR, 2018), *LeadersRank* 1 e 2 (AHAJJAM; HADDAD; BADIR, 2015) ao serem aplicadas nas instâncias trabalhadas nesta Seção. Analisando a Tabela 47 podemos observar que, para a rede social *Zachary's karate club*, as modularidades obtidas pelos algoritmos LCDA-GR 1 e 2 são maiores que as obtidas pelos algoritmos LCDA 1 e 2 e *LeaderRank* 1 e 2. O mesmo pode-se afirmar para a rede social *American college football*, em que a modularidade obtida por ambos os algoritmos propostos superam as modularidades obtidas pelos algoritmos LCDA 1 e 2 e *LeaderRank* 1 e 2. De fato, as modularidades obtidas para ambas as instâncias pelos algoritmos propostos corresponde ao valor ótimo reportado na literatura, conforme pode ser visto em Agarwal e Kempe (2008).

<b>Algoritmos/Redes Sociais</b>	<i>Zachary's karate club</i>	<i>American College football</i>	<i>Dolphin social network</i>	<i>Books about US politics</i>	<i>Political Blogs</i>
<i>LeaderRank 1</i>	0,3180	0,3780	-	-	0,2359
LCDA 1	0,3266	0,5162	-	-	0,3977
LCDA-GR 1	<b>0,4198</b>	<b>0,6046</b>	<b>0,5285</b>	<b>0,5272</b>	<b>0,4270</b>
<i>LeaderRank 2</i>	0,3014	0,3750	-	-	0,2388
LCDA 2	0,3266	0,4842	-	-	0,4062
LCDA-GR 2	<b>0,4198</b>	<b>0,6046</b>	<b>0,5285</b>	<b>0,5272</b>	<b>0,4269</b>

Tabela 47 – Comparação entre modularidades dos algoritmos propostos e da literatura.

#### 4 BENCHMARK EM ALGORITMOS QUE DETECTAM LÍDERES E COMUNIDADES

Testar um algoritmo significa essencialmente analisar uma rede com uma estrutura comunitária e de liderança bem definidas e recuperar suas comunidades e líderes. Idealmente, gostaria-se de ter muitas instâncias de redes reais cujas comunidades e líderes são precisamente conhecidos, mas esse geralmente não é o caso. Portanto, os testes mais abrangentes são realizados em redes geradas por computador, com uma estrutura de liderança e comunidade integrada. Em geral, um algoritmo deve ser testado em um conjunto de instâncias em que as comunidades tenham tamanhos e grau médio de vértices variáveis, pois esses parâmetros podem afetar seriamente o resultado do método e revelar seus limites. Esta questão é tão crucial quanto desenvolver novos algoritmos para a detecção de líderes e comunidades.

O conjunto de instâncias testes mais famoso, destinado à detecção de comunidades apenas, é uma classe de redes introduzida por Girvan e Newman (GN) (GIRVAN; NEWMAN, 2002). Cada rede possui 128 nós, divididos em quatro grupos com 32 nós. O grau médio da rede é 16 e os nós têm aproximadamente o mesmo grau, como em um grafo aleatório. Dado um parâmetro  $k_{out}$  (indicando o número esperado de arestas que une cada nó a nós de grupos diferentes ou grau externo) menor que 8, cada nó compartilha mais arestas com outros nós do seu grupo do que com o restante da rede. Nesse caso, os quatro grupos são comunidades bem definidas e um bom algoritmo deve ser capaz de identificá-las. Esse *benchmark* é usado regularmente para testar algoritmos. É necessário ressaltar, no entanto, duas características que o impede de se aproximar de um conjunto de redes reais: (i) todos os nós da rede têm essencialmente o mesmo grau; (ii) as comunidades são do mesmo tamanho.

Redes reais são caracterizadas por distribuições heterogêneas de grau de nós consideradas responsáveis por uma série de atributos notáveis de redes reais (ALBERT; JEONG; BARABÁSI, 2000; COHEN et al., 2000; PASTOR-SATORRAS; VESPIGNANI, 2001). Portanto, é desejável que um *benchmark* tenha uma distribuição de graus heterogênea. Da mesma forma, seria desejável poder supor tamanhos diferentes de comunidades, já que a distribuição dos tamanhos das comunidades de redes reais é ampla (PALLA et al., 2005; GUIMERA et al., 2003; CLAUSET; NEWMAN; MOORE, 2004).

Este capítulo tem por objetivo propor um algoritmo para a geração de instâncias testes composto de grafos simulando redes sociais. Nesta proposta, o número de comunidades e o número de indivíduos em cada uma delas é um parâmetro do método, sendo definido pelo usuário. O

grau dos vértices é variável e, para cada comunidade, um líder é determinado. Além disso, a metodologia proposta permite inserir falsos-líderes, aqui considerados indivíduos quase tão bem conectados quanto os líderes, mas que não são tão bons em disseminar a informação quanto estes. O diferencial dessa proposta é que os grafos gerados possibilitam a testagem da detecção de comunidades e de líderes das comunidades, conjuntamente. Assim, permite estudar a acurácia com que os algoritmos detectam não somente as comunidades mas também seus líderes.

#### 4.1 GERADOR DE GRAFOS COM LIDERANÇA-COMUNIDADE (LCGG)

Seja um grafo  $G = (V, E)$  não direcionado, com  $V$  sendo o conjunto nós. Cada nó em  $G$  é associado a um elemento do conjunto de dados com  $|V| = n$ , a quantidade de nós em  $G$ .  $E$  é o conjunto de  $|E| = m$  arestas. Cada aresta representa um relacionamento (interação) entre um par de nós. A estrutura da rede é representada por uma matriz de adjacência  $A = \{a_{ij}\}$ , com  $a_{ij} \in \mathbb{R}$ . Neste trabalho, considera-se  $a_{ij} = 1$ , se existir uma relação entre os nós  $i$  e  $j$ , caso contrário,  $a_{ij} = 0$ .

Consideramos os seguintes parâmetros:

- $K$ , o número de comunidades;
- $n_k$ , o número de indivíduos em cada comunidade, com  $k = 1, \dots, K$ ;
- $\alpha$ , a proporção de arestas entre nós de comunidades distintos.

Definimos  $\alpha \in (0; 0.5)$ , uma vez que, para haver uma estrutura de comunidade bem definida devemos ter mais conexões entre nós do mesmo grupo do que entre nós de grupos diferentes.

Além disso, as seguintes probabilidades são pré-definidas:

- $p_{leader}$ , a probabilidade de conexão de um *nó líder* com outro nó da rede, onde  $p_{leader} \in (0, 1]$ ;
- $p_{follower}$ , a probabilidade de um *nó não líder* se conectar com outro nó da rede, onde  $p_{follower} \in (0, 1]$ ;
- $p_{prototypical\_leader}$ , probabilidade de um *falso-líder* se conectar a outro nó da rede, onde  $p_{prototypical\_leader} \in (0, 1]$ .

Um nó falso-líder aqui é considerado um indivíduo quase tão bem conectado quanto o líder, mas que não é tão bom em disseminar a informação quanto esse.

Como estratégia de centralizar os líderes na rede e dentro dos seus grupos, assumimos que os nós não líderes possuem probabilidade de conexão menor que os nós líderes tal que  $p_{leader} > p_{follower}$ . Caso seja de interesse inserir falsos-líderes, devemos ter  $p_{leader} > p_{prototypical\_leader} > p_{follower}$ .

O LCGG segue as seguintes etapas para a geração de um grafo  $G$  com  $K$  grupos, composto de  $G_k$  subgrafos.

1. Para cada comunidade  $k = 1, \dots, K$ , seleciona-se aleatoriamente  $n_k$  nós que constituirão a comunidade  $K$ .
2. Para cada grupo  $k$  seleciona-se aleatoriamente, dentre os nós do grupo, um para ser o nó líder. Caso seja de interesse incluir falsos-líderes, novamente é selecionado um nó de cada grupo aleatoriamente.
3. Para cada grupo  $k$ , constrói-se uma matriz  $\mathbb{P}_k$  de conexão intra-grupo, tal que:

$$\mathbb{P}_k(i, j) = \begin{cases} p_{leader} & \text{se } i \text{ ou } j \text{ é um nó líder;} \\ p_{prototypical\_leader} & \text{se } i \text{ é um nó falso-líder e } j \text{ é um nó não-líder e vice-versa;} \\ p_{follower} & \text{se } i \text{ ou } j \text{ é um nó não-líder.} \end{cases}$$

A partir da matriz  $P_k$ , gera-se um sub-grafo  $G_k$ , com  $k = 1, \dots, K$ , através do método gerador de um modelo de bloco estocástico (FAUST; WASSERMAN, 1992);

4. Dado o número  $m_{intra}$ , indicando todas as arestas criadas com os nós dentro de cada grupo e o parâmetro  $\alpha$ , calcula-se  $m_{inter} = \left(\frac{\alpha}{1-\alpha}\right) m_{intra}$ , representando todas arestas criadas entre nós de diferentes grupos. Em seguida, os sub-grafos  $G_k$  são unidos da seguinte forma:

- a) Para cada par de grupos, sejam eles  $f$  e  $g$ , são realizadas  $\frac{m_{inter}}{\left(\frac{K(K-1)}{2}\right)}$  conexões entre os indivíduos, aleatoriamente. Em seguida, todos os líderes são conectados dois a dois. No caso em que há falsos-líderes, para cada grupo, seleciona-se aleatoriamente o nó líder ou falso-líder a partir de suas probabilidades de conexão  $p_{leader}$  e  $p_{prototypical\_leader}$ , respectivamente;

- b) Caso  $\max \left[ m_{inter} - \left( \frac{K(K-1)}{2} \right), 0 \right] > 0$ , para cada par ( $f$  e  $g$ ) de grupos, um nó do grupo  $f$  é selecionado a partir das probabilidades  $p_i$  dos indivíduos do grupo  $f$  e um nó do grupo  $g$  é selecionado a partir das probabilidades  $p_i$  dos indivíduos do grupo  $g$  e uma conexão entre esses nós é estabelecida;
- c) Esse processo é repetido até que todas as  $\frac{m_{inter}}{\left( \frac{K(K-1)}{2} \right)}$  conexões sejam realizadas para cada par de grupo.
5. Ao final, caso tenha sido gerado um grafo desconexo, as ligações entre os nós desconectados e o restante do grafo são geradas da seguinte forma:
- (a) No caso do grafo sem falsos-líderes, estabelece-se uma conexão entre cada nó desconexo e o líder de seu grupo;
- (b) No caso em que o grafo possua falsos-líderes, seleciona-se um entre o nó líder e o falso-líder do grupo ao qual o nó desconectado pertence, a partir de suas probabilidades de conexão  $p_{leader}$  e  $p_{prototypical\_leader}$ , respectivamente, para realizar a conexão.

## 4.2 TESTES E RESULTADOS COMPUTACIONAIS

Nesta seção são apresentados os resultados de dois estudos realizados a partir do conjunto de instâncias testes proposto. A primeira bateria de testes tem o intuito estudar as características dos grafos gerados e a segunda, comparar o desempenho dos algoritmos LCDA, LCDA-GRASP e LCDA-GR, discutidos nos Capítulos 2 e 3, em relação as suas qualidades de detecção de líderes e comunidades. Esses algoritmos foram aqui implementados usando o software estatístico R (R Core Team, 2018) e compilado na versão 8.2.0. Os testes foram realizados em um computador Intel i7-8550 com 8 núcleos de 1.8GHz e 16GB de memória RAM rodando Ubuntu 20.04.

Os grafos utilizados nessa Seção foram gerados a partir dos seguintes parâmetros:  $n = 40$ ,  $k = 4$  e  $n = 100$ ,  $k = 8$ . Além disso, para cada par destes utilizamos  $\alpha = (0.1; 0.2; 0.3; 0.4; 0.5; 0.6)$ . As probabilidades de conexão dos nós líderes e nós não líderes assumidos foram  $(p_{leader}, p_{follower}) = \{(0.5; 0.3), (0.7; 0.2)\}$ . Para cada par  $(p_{leader}, p_{follower})$  também simulamos o caso em que há ou não falsos-líderes, em que as probabilidade de conexão dos mesmos são dadas por  $p_{prototypical\_leader} = (0.3; 0.4; 0.2; 0.5)$ . Ao final, totalizamos 24 diferentes tipos de instâncias,

conforme demonstrado na Tabela 48. Para cada instâncias foram realizadas  $B = 300$  réplicas para a análise.

Tabela 48 – Tipos de instâncias a partir das combinações dos parâmetros  $\alpha$ ,  $p_{leader}$ ,  $p_{follower}$  e  $p_{prototypical\_leader}$ .

Inst.	$\alpha$	$p_{leader}$	$p_{follower}$	$p_{prototypical\_leader}$
1	0.1	0.5	0.3	0.3
2	0.1	0.5	0.3	0.4
3	0.1	0.7	0.2	0.2
4	0.1	0.7	0.2	0.5
5	0.2	0.5	0.3	0.3
6	0.2	0.5	0.3	0.4
7	0.2	0.7	0.2	0.2
8	0.2	0.7	0.2	0.5
9	0.3	0.5	0.3	0.3
10	0.3	0.5	0.3	0.4
11	0.3	0.7	0.2	0.2
12	0.3	0.7	0.2	0.5
13	0.4	0.5	0.3	0.3
14	0.4	0.5	0.3	0.4
15	0.4	0.7	0.2	0.2
16	0.4	0.7	0.2	0.5
17	0.5	0.5	0.3	0.3
18	0.5	0.5	0.3	0.4
19	0.5	0.7	0.2	0.2
20	0.5	0.7	0.2	0.5
21	0.6	0.5	0.3	0.3
22	0.6	0.5	0.3	0.4
23	0.6	0.7	0.2	0.2
24	0.6	0.7	0.2	0.5

Para avaliar a qualidade da detecção de comunidades em grafos, adotou-se o coeficiente de modularidade (CLAUSET; NEWMAN; MOORE, 2004), apresentada em 1. A Figura 6 apresenta dois grafos gerados através do algoritmo LCGG para  $n = 40$  e  $K = 4$  e  $n = 100$  e  $K = 8$ , respectivamente.

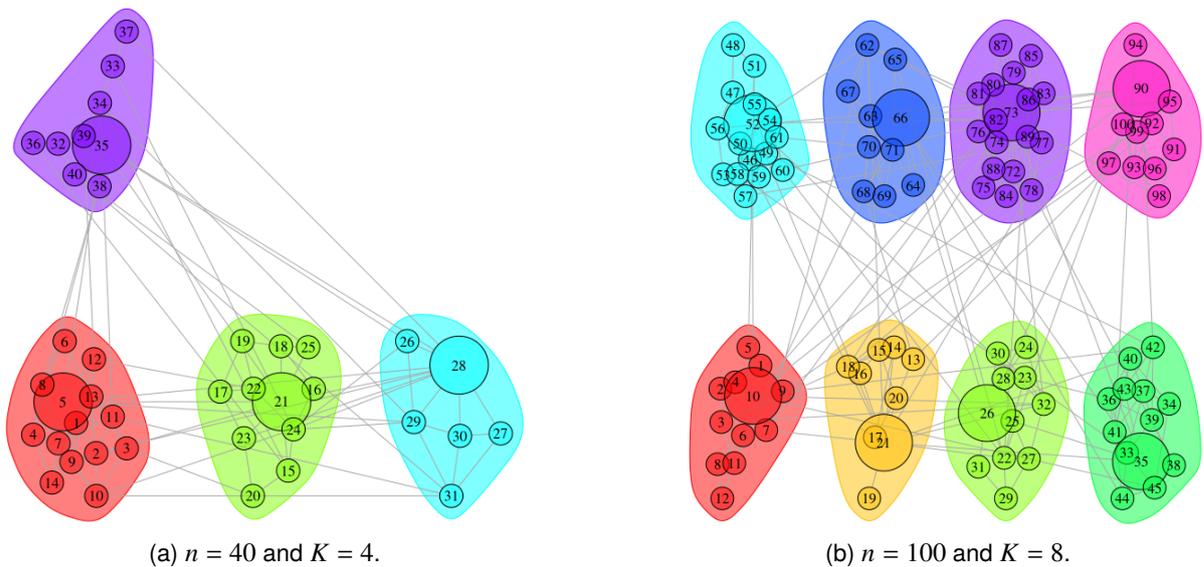


Figura 6 – Grafos gerados através do LCGG para  $n = 40$  e  $K = 4$  e  $n = 100$  e  $K = 8$ , respectivamente.

#### 4.2.1 Estudo de Simulação 1: Características dos Grafos Gerados

Os resultados aqui apresentados dizem respeito a 300 execuções independentes do algoritmo proposto para os tipos de Instâncias apresentados na Tabela 48. Na avaliação dos grafos gerados foram computadas as seguintes medidas: grau dos nós, número de arestas geradas, coeficiente de modularidade  $Q$  (Eq. 1) e a centralidade de autovetor dos nós líderes e dos falsos-líderes. Para os casos em que  $p_{\text{prototypical\_leader}} = p_{\text{follower}}$ , representando a não presença de falsos-líderes, as suas centralidades não foram computadas.

As Tabelas 49 e 50 apresentam os resultados para  $n = 40$  e  $K = 4$  e  $n = 100$  e  $K = 8$ , respectivamente. As colunas 1 e 2 apresentam o grau médio e o número de arestas ( $m$ ) médio dos grafos. A coluna 3 apresenta a média do valor calculado do coeficiente de modularidade ( $Q$ ). As colunas 4 e 5 apresentam a estrutura de liderança com a média e, em parênteses, o desvio padrão dos valores calculados da centralidade de autovetor dos nós líderes e *prototypical leaders*. A última coluna apresenta a média do tempo de execução do algoritmo em segundos.

Pode-se observar que, conforme aumentamos o valor do parâmetro  $\alpha$ , maior é o grau mé-

dio e o número de arestas. Daí se conclui que, quanto maior for o parâmetro  $\alpha$ , maior será a conectividade do grafo gerado.

Analisando-se as medidas de centralidade dos nós líderes e prototypical leaders, verifica-se que conforme ocorre o aumento do parâmetro  $\alpha$  há aumento na centralidade dos nós líderes e prototypical leaders. O desvio-padrão, no entanto, diminui com o aumento de  $\alpha$ . Ou seja, conforme aumentamos as conexões entre os nós de grupos diferentes, mais centrais se tornam os nós líderes e prototypical leaders. Observe que, as centralidades dos nós líderes é sempre, em média, maior que a dos nós prototypical leaders.

O parâmetro  $\alpha$  tem grande influência também na modularidade  $Q$  dos grafos gerados. Quanto maior  $\alpha$ , menor se torna a modularidade dos grafos. Isso ocorre, pois conforme  $\alpha$  aumenta, mais conexões entre grupos diferentes são geradas. Observando-se a Equação 1 podemos verificar que para essas conexões a função indicadora  $\delta(C_i, C_j) = 0$ . Com isso a modularidade tende a diminuir.

O nosso conjunto de instâncias começa ter uma redução na modularidade, mesmo quando há 80% das conexões intra-grupos ( $\alpha$  pequeno). Isso se deve ao fato de que a otimização da modularidade possui um limite de resolução intrínseco (FORTUNATO; BARTHELEMY, 2007). De fato é esperado que, conforme o número de conexões inter-grupos se aproxima do número de conexões intra-grupos, a modularidade do grafo diminua.

Há indicativos, a partir desses resultados, que os grafos gerados pelo algoritmo proposto revela uma relação entre as estruturas de comunidade e liderança. Conforme aumentamos a modularidade do grafo gerado, ou seja, quanto mais intra-conectados são os grupos e menos inter-conectados eles são, menor é a centralidade dos líderes. Por outro lado, conforme tornamos os líderes mais centrais, menor se torna a modularidade do grafo.

As Figuras 7, 8 e 9 mostram o aumento do parâmetro  $\alpha$  implicam no incremento da média do grau e do número de arestas dos grafos gerados e no aumento da variabilidade, como pode ser visto pela altura dos violins. Na modularidade, o aumento de  $\alpha$  causa o efeito inverso, a modularidade tende a diminuir e a sua variabilidade também, o que pode ser observado pelos violins decrescendo e com as alturas diminuindo.

Tabela 49 – Resultados do estudo de simulação para as Instâncias com  $n = 40$  e  $K = 4$ .

Inst.	Grau Médio	$m$	$Q$	Centralidade do Leader (Std. Deviation)	Centralidade do Prototypical Leader (Std. Deviation)	Tempo (s)
1	4.25	84.95	0.4996	0.5828 (0.1810)	-	0.49
2	4.44	88.71	0.5076	0.4379 (0.1527)	0.3613 (0.1339)	0.52
3	3.57	71.35	0.5011	0.6470 (0.1669)	-	0.53
4	4.15	82.96	0.5207	0.5060 (0.1555)	0.4026 (0.1389)	0.51
5	4.60	92.08	0.4819	0.6110 (0.1733)	-	0.52
6	4.76	95.25	0.4869	0.4734 (0.1470)	0.3939 (0.1450)	0.52
7	3.85	76.94	0.4784	0.6729 (0.1572)	-	0.51
8	4.45	88.91	0.4972	0.5553 (0.1514)	0.4282 (0.1360)	0.51
9	5.09	101.79	0.4534	0.6577 (0.1646)	-	0.50
10	5.39	107.88	0.4567	0.5201 (0.1580)	0.4416 (0.1416)	0.49
11	4.29	85.80	0.4526	0.7115 (0.1546)	-	0.50
12	4.94	98.75	0.4588	0.6136 (0.1418)	0.4831 (0.1335)	0.50
13	5.77	115.49	0.4201	0.6921 (0.1562)	-	0.53
14	6.02	120.31	0.4205	0.5822 (0.1463)	0.4966 (0.1403)	0.50
15	4.78	95.63	0.4145	0.7562 (0.1257)	-	0.50
16	5.59	111.73	0.4588	0.6849 (0.1315)	0.5469 (0.1267)	0.51
17	6.74	134.71	0.3836	0.7484 (0.1266)	-	0.52
18	6.98	139.69	0.3830	0.6706 (0.1261)	0.5659 (0.1315)	0.48
19	5.50	110.02	0.3815	0.7896 (0.1086)	-	0.52
20	6.40	128.08	0.4198	0.7625 (0.1030)	0.6228 (0.1074)	0.65
21	8.01	160.19	0.3364	0.8180 (0.0912)	-	0.61
22	8.31	166.21	0.3386	0.7508 (0.0983)	0.6535 (0.1121)	0.64
23	6.61	132.19	0.3368	0.8526 (0.0754)	-	0.65
24	7.71	154.26	0.3368	0.8283 (0.0814)	0.6838 (0.0927)	0.58

Tabela 50 – Resultados do estudo de simulação para as Instâncias com  $n = 100$  e  $K = 8$ .

Inst.	Grau Médio	$m$	$Q$	Centralidade do Leader (Std. Deviation)	Centralidade do Prototypical Leader (Std. Deviation)	Tempo (s)
1	14.93	298.60	0.6167	0.5788 (0.2381)	-	3.11
2	15.35	306.54	0.6231	0.2984 (0.1474)	0.2381 (0.1230)	3.39
3	11.86	237.26	0.6111	0.6608 (0.2158)	-	3.52
4	13.23	264.52	0.6371	0.3759 (0.1759)	0.2768 (0.1334)	3.36
5	16.26	325.25	0.5942	0.6058 (0.2419)	-	3.22
6	16.82	336.38	0.5968	0.3321 (0.1664)	0.2698 (0.1271)	3.64
7	13.10	261.91	0.5862	0.6785 (0.2121)	-	3.44
8	14.66	293.18	0.6080	0.4237 (0.1856)	0.3179 (0.1438)	3.17
9	18.31	366.17	0.5643	0.6148 (0.2362)	-	3.25
10	18.89	377.90	0.5677	0.3719 (0.1692)	0.3069 (0.1467)	3.26
11	14.41	288.15	0.5625	0.6990 (0.1960)	-	3.14
12	16.19	323.76	0.5713	0.4927 (0.1893)	0.3739 (0.1453)	3.52
13	20.54	410.89	0.5363	0.6323 (0.2252)	-	3.47
14	21.23	424.52	0.5369	0.4301 (0.1838)	0.3588 (0.1533)	4.12
15	16.40	328.00	0.5310	0.7307 (0.1778)	-	4.05
16	18.20	323.76	0.5368	0.5552 (0.1696)	0.4349 (0.1373)	3.81
17	23.62	472.39	0.5002	0.6798 (0.2042)	-	3.66
18	24.31	486.16	0.5012	0.5232 (0.1813)	0.3588 (0.1533)	3.93
19	18.91	378.12	0.4964	0.7803 (0.1425)	-	3.90
20	20.93	418.59	0.6532	0.6772 (0.1519)	0.5271 (0.1254)	3.20
21	28.21	564.15	0.4580	0.7785 (0.1407)	-	3.49
22	29.06	581.13	0.4562	0.6506 (0.1513)	0.5528 (0.1363)	3.39
23	22.35	447.04	0.4546	0.8508 (0.0845)	-	3.34
24	25.04	500.79	0.4481	0.7851 (0.0812)	0.6246 (0.0825)	3.20

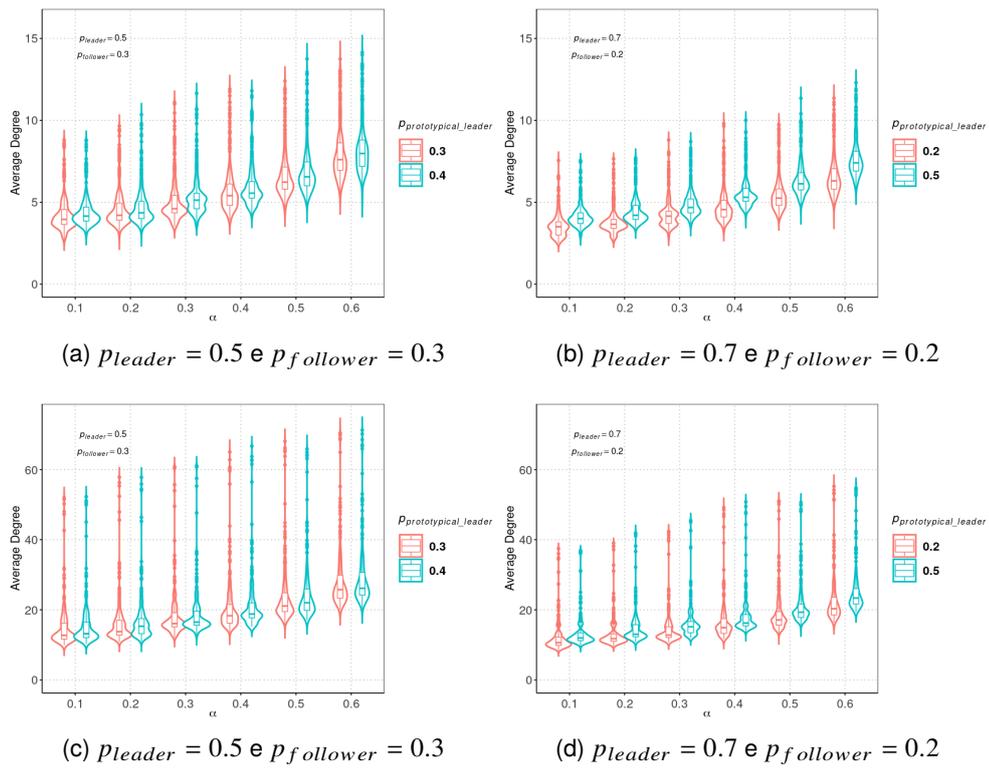


Figura 7 – Grau médio obtido para as Instâncias com: (top)  $n = 40$  e  $K = 4$  e (bottom)  $n = 100$  e  $K = 8$ .

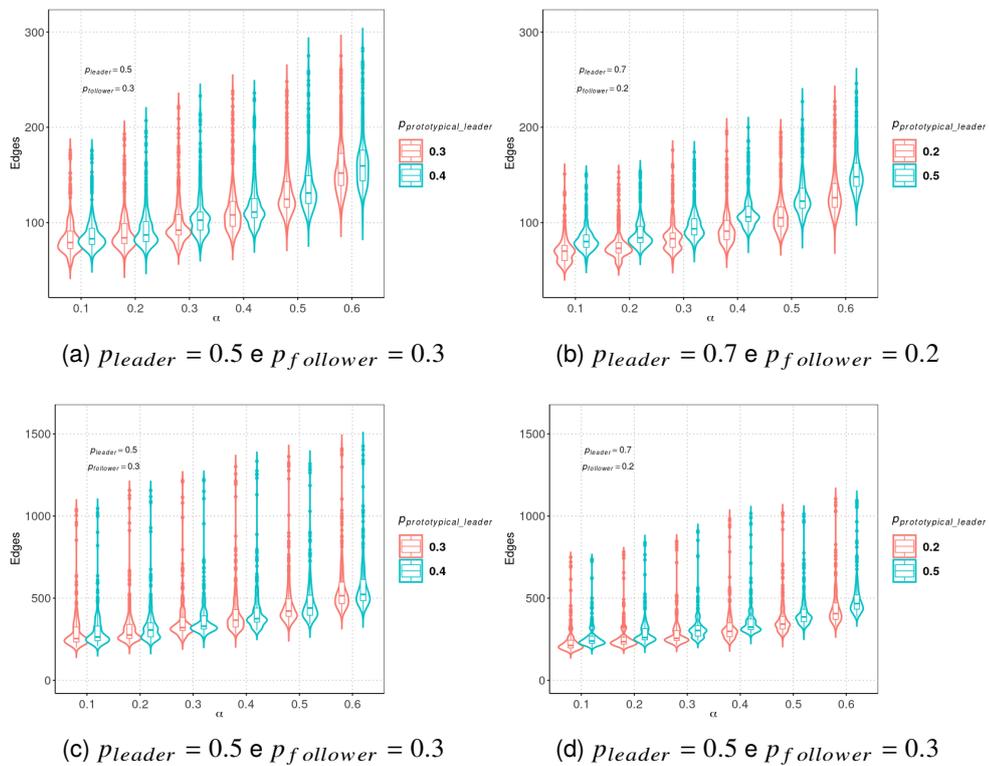


Figura 8 – Número de arestas obtido para as Instâncias com: (top)  $n = 40$  e  $K = 4$  e (bottom)  $n = 100$  e  $K = 8$ .

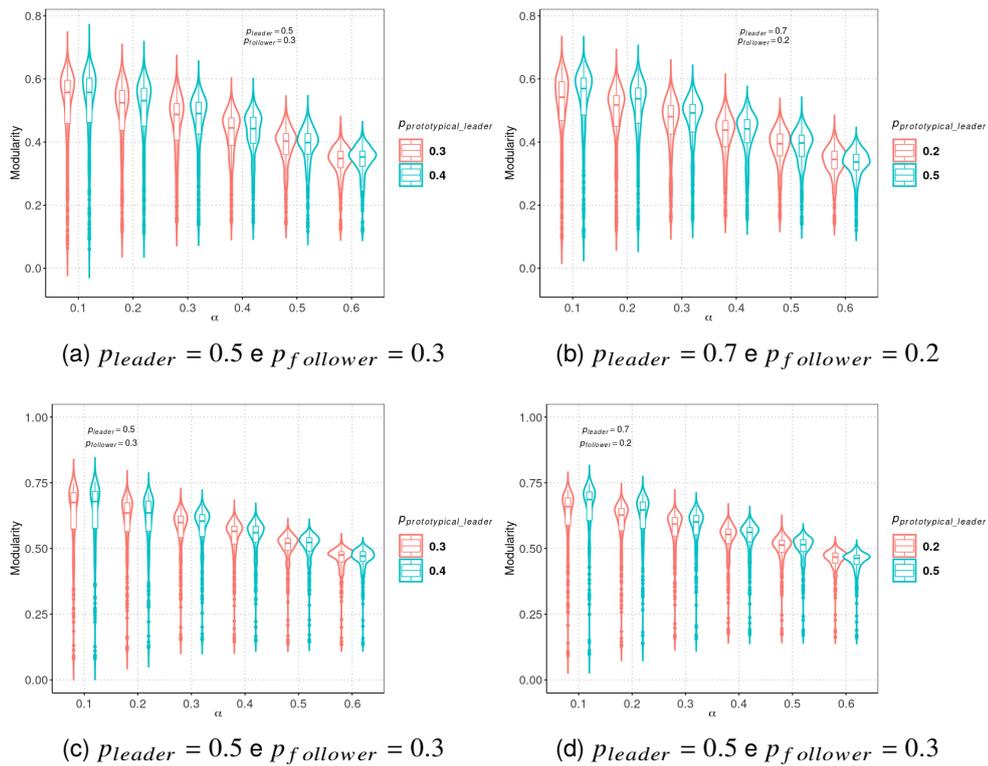


Figura 9 – Modularidades obtidas para as Instâncias com: (top)  $n = 40$  e  $K = 4$  e (bottom)  $n = 100$  e  $K = 8$ .

#### 4.2.2 Estudo de Simulação 2: Comparação de desempenho dos algoritmos LCDA 1 e 2, LCDA-GRASP 1 e 2 e LCDA-GR 1 e 2

Os resultados apresentados dizem respeito a 300 execuções independentes do algoritmo proposto para cada uma das combinações dos parâmetros definida na Tabela 48. Para cada grafo gerado são executados os algoritmos LCDA-GRASP 1 e 2 e LCDA-GR 1 e 2.

A fim de avaliar as soluções obtidas pelos algoritmos computamos o erro quadrático médio ( $EQM$ ) levando-se em consideração a modularidade do grafo gerado e a modularidade obtida pelos algoritmos, pela expressão:

$$EQM = \sum_{i=1}^s \frac{(Q_{i_{LCDA}} - Q_i)^2}{s},$$

onde  $s$  é o número de soluções geradas;  $Q_{i_{LCDA}}$ , a modularidade retornada pelo algoritmo LCDA 1 ou 2 e  $Q_i$ , a modularidade do grafo.

Além do  $EQM$ , calculamos a proporção média de acerto dos líderes dos grupos através da expressão:

$$PMA = \sum_{i=1}^s \frac{\#acertos_i}{K},$$

em que  $\#acertos_i$  é o número de líderes detectados corretamente pelo algoritmo e  $K$  é o número de líderes na solução.

As Tabelas 51, 52, 53 e 54 apresentam os valores calculados de  $EQM$  e  $PMA$  a partir dos testes realizados com os algoritmos LCDA 1 e 2, LCDA-GRASP 1 e 2 e LCDA-GR 1 e 2 para  $n = 40$  e  $K = 4$  e  $n = 100$  e  $K = 8$ , respectivamente. O objetivo é comparar os resultados dos algoritmos e avaliar suas detecções de líderes e comunidades. As instâncias para a geração dos grafos sintéticos são as mesmas presentes na Tabela 48.

Observando as Tabelas 51 e 52 podemos afirmar que, conforme o tamanho do grafo aumenta, menor é a capacidade dos algoritmos LCDA 1 e 2 de detecção da estrutura comunidade. Por outro lado, os algoritmos LCDA-GRASP 1 e 2 e LCDA-GR 1 e 2 possuem uma redução na medida  $EQM$ , indicando uma melhor capacidade de detecção das comunidades dos grafos gerados. Outra característica observada é que, o  $EQM$  dos algoritmos LCDA 1 e 2 são os maiores observados, em ambas as Tabelas, quando comparados aos demais algoritmos, indicando que os algoritmos propostos possuem uma melhor acurácia de detecção de comunidades que os algoritmos LCDA 1 e 2. Além disto, os algoritmos LCDA-GRASP 1 e 2 e LCDA-GR 1 e 2 possuem resultados bastantes similares entre si.

Ao observarmos as Tabelas 53 e 54 afirma-se que o aumento no tamanho do grafo reduz a capacidade de detecção de líderes de todos os algoritmos, uma vez que, a medida de avaliação  $PMA$  reduz conforme aumentamos o tamanho de  $n$  e, também, a quantidade de comunidades  $k$ . De modo geral, temos que conforma o valor de  $\alpha$  do gerador diminui, melhores são as detecções realizadas pelos algoritmos LCDA-GRASP 1 e 2 e LCDA-GR 1 e 2 em relação aos algoritmos LCDA 1 e 2. Em particular, os algoritmos LCDA-GRASP 1 e LCDA-GR 1 se saem melhor entre os demais algoritmos, em grande parte das instâncias. Outra característica favorável aos algoritmos LCDA-GRASP 1 e 2 e LCDA-GR 1 e 2, está nos seus resultados superiores quando a diferença entre as probabilidades de conexão  $p_{leader}$  e  $p_{follower}$  é menor, em relação aos algoritmos LCDA 1 e 2. Porém, quando  $n$  e  $k$  aumentam, nossas propostas detectam melhores os líderes na grande maioria das instâncias em estudo, em particular, o algoritmo LCDA-GRASP 1 é o que detecta melhor os líderes entre as instâncias em estudo.

Tabela 51 –  $EQM$  obtidos pelos algoritmos LCDA 1 e 2 das instâncias com  $n = 40$  e  $k = 4$ 

Inst.	$EQM$					
	LCDA 1	LCDA 2	LCDA-GRASP 1	LCDA-GRASP 2	LCDA-GR 1	LCDA-GR 2
1	0.2536	0.2359	0.0039	0.0038	0.0038	0.0038
2	0.2064	0.1884	0.0037	0.0035	0.0034	0.0035
3	0.2178	0.2130	0.0048	0.0045	0.0046	0.0047
4	0.1590	0.1447	0.0033	0.0033	0.0034	0.0034
5	0.1994	0.1843	0.0037	0.0038	0.0039	0.0039
6	0.1749	0.1602	0.0026	0.0026	0.0027	0.0027
7	0.2086	0.2057	0.0038	0.0038	0.0038	0.0038
8	0.1377	0.1229	0.0031	0.0028	0.0031	0.0030
9	0.1856	0.1734	0.0029	0.0029	0.0029	0.0029
10	0.1436	0.1345	0.0027	0.0025	0.0026	0.0026
11	0.1946	0.1958	0.0040	0.0041	0.0041	0.0041
12	0.1249	0.1175	0.0023	0.0023	0.0023	0.0023
13	0.1529	0.1414	0.0022	0.0023	0.0023	0.0023
14	0.1434	0.1357	0.0023	0.0023	0.0023	0.0023
15	0.1513	0.1486	0.0039	0.0039	0.0039	0.0039
16	0.1219	0.1130	0.0023	0.0023	0.0024	0.0023
17	0.1268	0.1162	0.0025	0.0026	0.0025	0.0027
18	0.1161	0.1061	0.0023	0.0023	0.0023	0.0023
19	0.1343	0.1321	0.0031	0.0031	0.0032	0.0032
20	0.0918	0.0832	0.0022	0.0022	0.0022	0.0022
21	0.0781	0.0744	0.0028	0.0028	0.0027	0.0028
22	0.0939	0.0827	0.0020	0.0021	0.0021	0.0020
23	0.1018	0.0961	0.0032	0.0032	0.0032	0.0033
24	0.0860	0.0808	0.0023	0.0023	0.0023	0.0023

Tabela 52 – *EQM* obtidos pelos algoritmos LCDA 1 e 2 das instâncias com  $n = 100$  e  $k = 8$ 

Inst.	<i>EQM</i>					
	LCDA 1	LCDA 2	LCDA-GR 1	LCDA-GR 2	LCDA-GRASP 1	LCDA-GRASP 2
1	0.2304	0.2152	0.0002	0.0001	0.0002	0.0002
2	0.2577	0.2384	0.0001	0.0001	0.0001	0.0001
3	0.1982	0.2003	0.0002	0.0002	0.0003	0.0003
4	0.1870	0.1673	0.0001	0.0001	0.0001	0.0001
5	0.2340	0.2183	0.0002	0.0001	0.0002	0.0002
6	0.2275	0.2184	0.0001	0.0001	0.0002	0.0002
7	0.1777	0.1796	0.0002	0.0002	0.0003	0.0002
8	0.1580	0.1431	0.0001	0.0001	0.0001	0.0001
9	0.2073	0.1954	0.0003	0.0003	0.0003	0.0003
10	0.1993	0.1859	0.0002	0.0002	0.0003	0.0003
11	0.1585	0.1564	0.0003	0.0003	0.0004	0.0004
12	0.1464	0.1412	0.0002	0.0002	0.0002	0.0002
13	0.1915	0.1867	0.0003	0.0003	0.0004	0.0004
14	0.1765	0.1673	0.0004	0.0004	0.0004	0.0004
15	0.1507	0.1459	0.0003	0.0003	0.0003	0.0004
16	0.1136	0.1090	0.0002	0.0002	0.0002	0.0002
17	0.1529	0.1476	0.0004	0.0004	0.0005	0.0005
18	0.1724	0.1669	0.0004	0.0004	0.0004	0.0004
19	0.1170	0.1167	0.0005	0.0004	0.0005	0.0005
20	0.0980	0.0958	0.0004	0.0004	0.0004	0.0004
21	0.1093	0.1077	0.0006	0.0006	0.0006	0.0006
22	0.1191	0.1144	0.0006	0.0006	0.0006	0.0006
23	0.0857	0.0823	0.0006	0.0006	0.0006	0.0006
24	0.0752	0.0661	0.0005	0.0005	0.0005	0.0005

Tabela 53 – *PMA* obtidos pelos algoritmos LCDA 1 e 2 das instâncias com  $n = 40$  e  $k = 4$ 

Inst.	<i>PMA</i>					
	LCDA 1	LCDA 2	LCDA-GRASP 1	LCDA-GRASP 2	LCDA-GR 1	LCDA-GR 2
1	0.3175	0.2756	0.3494	0.2650	0.3483	0.2278
2	0.2725	0.2123	0.3256	0.2761	0.2650	0.2778
3	0.4754	0.3882	0.4811	0.3443	0.2894	0.3089
4	0.3997	0.3287	0.4372	0.3311	0.4200	0.3400
5	0.3439	0.2781	0.3928	0.2878	0.3306	0.2389
6	0.2517	0.2172	0.2878	0.2689	0.3683	0.2194
7	0.4361	0.3373	0.3881	0.3478	0.3533	0.2361
8	0.3468	0.3235	0.3683	0.4361	0.2478	0.3283
9	0.3073	0.2460	0.3639	0.2556	0.3733	0.2478
10	0.2766	0.2331	0.2628	0.2300	0.2400	0.2511
11	0.4808	0.3996	0.3337	0.3556	0.3067	0.3122
12	0.3998	0.3537	0.3856	0.3328	0.3694	0.3794
13	0.2848	0.2491	0.3122	0.2617	0.2244	0.2250
14	0.2588	0.2153	0.2872	0.2817	0.3356	0.2206
15	0.4739	0.3983	0.3589	0.2678	0.3339	0.3050
16	0.4784	0.3978	0.3678	0.3833	0.3444	0.3228
17	0.3281	0.2364	0.4056	0.3550	0.3556	0.3183
18	0.2746	0.2314	0.3839	0.2417	0.2806	0.2278
19	0.5010	0.4119	0.3839	0.3194	0.2833	0.2294
20	0.5173	0.4001	0.3611	0.3200	0.3328	0.3089
21	0.3262	0.2387	0.3194	0.2472	0.3361	0.3500
22	0.2872	0.2362	0.2844	0.2639	0.2861	0.2361
23	0.5163	0.4284	0.3033	0.3467	0.3411	0.3472
24	0.5006	0.4566	0.3939	0.3317	0.2917	0.2528

Tabela 54 – *PMA* obtidos pelos algoritmos LCDA 1 e 2 das instâncias com  $n = 100$  e  $k = 8$ 

Inst.	<i>PMA</i>					
	LCDA 1	LCDA 2	LCDA-GR 1	LCDA-GR 2	LCDA-GRASP 1	LCDA-GRASP 2
1	0.1345	0.1124	0.2195	0.1566	0.2115	0.1524
2	0.2573	0.2360	0.3487	0.3134	0.2710	0.1595
3	0.2483	0.2249	0.2035	0.1273	0.1619	0.1287
4	0.3561	0.3277	0.3270	0.3684	0.3047	0.1870
5	0.1473	0.1266	0.1882	0.1440	0.1775	0.1383
6	0.2520	0.2255	0.2880	0.2958	0.2379	0.1967
7	0.2838	0.2509	0.2269	0.1640	0.1940	0.1821
8	0.3479	0.3218	0.4160	0.3753	0.2344	0.2412
9	0.1473	0.1276	0.2091	0.1413	0.1907	0.1214
10	0.2423	0.2167	0.3440	0.2568	0.2577	0.1508
11	0.2726	0.2270	0.2231	0.1526	0.1538	0.1386
12	0.3442	0.3024	0.3949	0.3618	0.3478	0.2592
13	0.1627	0.1408	0.2119	0.1745	0.1937	0.1298
14	0.2308	0.2108	0.3437	0.2856	0.2592	0.2041
15	0.2678	0.2372	0.1863	0.1780	0.1684	0.1530
16	0.3439	0.3084	0.4536	0.3963	0.3681	0.2818
17	0.1715	0.1453	0.1898	0.1657	0.1557	0.1427
18	0.2011	0.1728	0.3198	0.2592	0.2352	0.2712
19	0.3097	0.2583	0.1924	0.1834	0.1482	0.1482
20	0.3550	0.3105	0.3946	0.3101	0.2369	0.1968
21	0.1728	0.1501	0.1942	0.1813	0.1735	0.1641
22	0.2076	0.1769	0.3433	0.2481	0.2091	0.1808
23	0.3217	0.2846	0.2269	0.2085	0.2027	0.1175
24	0.3648	0.3355	0.3801	0.3275	0.2225	0.2339

## 5 CONCLUSÃO

Neste trabalho foram discutidos conceitos e algoritmos utilizados na literatura para a detecção de líderes e/ou comunidades, além de geradores de redes sintéticas com estruturas conjuntas de comunidade e liderança.

Os resultados experimentais mostram que os algoritmos LCDA-GRASP 1 e 2 propostos apresentam melhores desempenhos em relação aos algoritmos originais LCDA 1 e 2. Além disso, podemos definir regiões dos parâmetros dos algoritmos que fornecem ótimas soluções para diferentes redes.

Nossos experimentos mostram que o uso de diversas métricas de similaridade e centralidade proporciona uma maior flexibilidade aos algoritmos ao se depararem com diferentes tipos de redes/sistemas complexos, tornando-os capazes de obterem bons desempenhos em diferentes tipos de topologia.

Neste mesmo capítulo, também apresentou dois novos algoritmos baseados na heurística GRASP Reativo para resolver a detecção de líderes e comunidades. O principal objetivo deste trabalho foi estender os algoritmos desenvolvidos no Capítulo 3 de tal forma que, as novas propostas fossem capazes de auto-selecionar os parâmetros  $\alpha_c$  e  $\alpha_s$  e, desta forma, reduzir o esforço computacional utilizado para a sua seleção, principalmente, ao trabalharmos com grafos com  $n > 100$ .

Além disto, pode-se comparar ambas as heurísticas GRASP e GRASP Reativo ao serem utilizadas conjuntamente com os algoritmos LCDA 1 e 2 na fase de construção. Nossos resultados experimentais mostram que os algoritmos LCDA-GR 1 e 2 propostos também apresentam melhores desempenhos em relação aos algoritmos originais LCDA 1 e 2. Além disto, os mesmos são capazes de fazer o auto-ajuste dos parâmetros  $\alpha_c$  e  $\alpha_s$  de tal forma a que fornecerem ótimas soluções para diferentes redes.

Nossos estudos experimentais também mostram que o uso de diversas métricas de similaridade e centralidade proporciona uma maior flexibilidade aos algoritmos ao se depararem com diferentes tipos de redes/sistemas complexos, tornando-os capazes de obterem bons desempenhos em diferentes tipos de topologia.

O segundo trabalho apresentamos uma proposta de algoritmo para a geração de grafos representando redes sociais com estrutura conjunta de comunidades e liderança visando a avaliação de algoritmos que detectam líderes e comunidades conjuntamente, visto que, de nosso

conhecimento, não há esse tipo de instâncias testes na literatura.

Foram apresentadas medidas descritivas dos grafos gerado, tais como grau médio e número de arestas, demonstrando como a estrutura de conexões dos nós muda conforme utiliza-se os diferentes parâmetros do nosso algoritmo. Além disto, outras medidas relacionadas as estruturas de comunidades e lideranças foram apresentadas como a modularidade e a centralidade de nós.

Dois conjuntos de instâncias de pequeno e médio porte foram gerados e utilizados para avaliar a qualidade de detecção das estruturas de liderança e comunidades dos algoritmos da literatura LCDA e com as nossas propostas LCDA-GRASP 1 e 2 e LCDA-GR 1 e 2. Concluimos que nossos algoritmos apresentam boa capacidade de detecção de comunidade para diferentes estruturas apresentadas e, principalmente quando o tamanho da rede aumenta, apresentam uma melhor qualidade de detecção de líderes do que os algoritmos LCDA 1 e 2.

Pode-se notar que, conforme mudamos a estrutura de comunidade dos grafos através dos parâmetros há um efeito na detecção dos líderes de ambos algoritmos, da mesma forma que, ao mudarmos os valores dos parâmetros referentes à liderança, afetamos a qualidade de detecção da estrutura de comunidade. Desta forma, nota-se que há uma relação de causa-efeito entre as estruturas de liderança e comunidades no que diz respeito a sua detecção por esses algoritmos.

Por fim, concluimos que o algoritmo LCGG proposto é capaz de gerar grafos adequados para análise e avaliação de algoritmos que detectam estruturas de comunidades e liderança conjuntamente, podendo revelar as limitações e vantagens de algoritmos de detecção.

Como proposta de trabalhos futuros podemos citar:

1. o desenvolvimento de novos algoritmos genéticos utilizando como base os algoritmos LCDA 1 e 2;
2. uma generalização dos algoritmos LCDA-GRASP e LCDA-GR para grafos com pesos na matriz de adjacência;
3. um gerador de grafos aleatórios utilizando os conceitos de modelos difusores para os líderes.

## REFERÊNCIAS

- AGARWAL, G.; KEMPE, D. Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B*, Springer, v. 66, n. 3, p. 409–418, 2008.
- AHAJJAM, S.; HADDAD, M. E.; BADIR, H. Leadersrank: Towards a new approach for community detection in social networks. In: IEEE. *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*. [S.l.], 2015. p. 1–8.
- AHAJJAM, S.; HADDAD, M. E.; BADIR, H. A new scalable leader-community detection approach for community detection in social networks. *Social Networks*, Elsevier, v. 54, p. 41–49, 2018.
- ALBERT, R.; JEONG, H.; BARABÁSI, A.-L. Error and attack tolerance of complex networks. *nature*, Nature Publishing Group, v. 406, n. 6794, p. 378–382, 2000.
- ALMEIDA, V. A. F. d. *Detecção de comunidades em grafos multicamada muito grandes*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, 2011.
- ALVES, I.; OLIVEIRA, C. S.; BRITO, J. A. de M. An analysis of the problem of community detection in networks. *Sistemas & Gestão*, Universidade Federal Fluminense, Escola de Engenharia, Departamento de Engenharia de Produção, v. 9, n. 4, p. 566–574, 2014.
- ATAY, Y.; KOC, I.; BABAOGLU, I.; KODAZ, H. Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms. *Applied Soft Computing*, Elsevier, v. 50, p. 194–211, 2017.
- BACK, K. *EXCHANGE AND POWER IN SOCIAL-LIFE-BLAU, PM*. [S.l.]: UNIV NORTH CAROLINA PRESS BOX 2288, CHAPEL HILL, NC 27515-2288, 1965.
- BADER, D. A.; MEYERHENKE, H.; SANDERS, P.; WAGNER, D. *Graph partitioning and graph clustering*. [S.l.]: American Mathematical Soc., 2013. v. 588.
- BAUMES, J.; GOLDBERG, M.; MAGDON-ISMAIL, M.; WALLACE, W. A. Discovering hidden groups in communication networks. In: SPRINGER. *International Conference on Intelligence and Security Informatics*. [S.l.], 2004. p. 378–389.
- BONACICH, P.; LLOYD, P. Eigenvector-like measures of centrality for asymmetric relations. *Social networks*, Elsevier, v. 23, n. 3, p. 191–201, 2001.
- CLAUSET, A.; NEWMAN, M. E.; MOORE, C. Finding community structure in very large networks. *Physical review E*, APS, v. 70, n. 6, p. 066111, 2004.
- COHEN, R.; EREZ, K.; BEN-AVRAHAM, D.; HAVLIN, S. Resilience of the internet to random breakdowns. *Physical review letters*, APS, v. 85, n. 21, p. 4626, 2000.
- DUARTE, A. R.; RIBEIRO, C. C.; URRUTIA, S. A hybrid ils heuristic to the referee assignment problem with an embedded mip strategy. In: SPRINGER. *International Workshop on Hybrid Metaheuristics*. [S.l.], 2007. p. 82–95.
- DUARTE, A. R.; RIBEIRO, C. C.; URRUTIA, S.; HAEUSLER, E. H. Referee assignment in sports leagues. In: SPRINGER. *International Conference on the Practice and Theory of Automated Timetabling*. [S.l.], 2006. p. 158–173.

- DUCZMAL, L.; BESSEGATO, L.; SANTOS, M.; NETO, S. F. Introdução às pilhas e testes de permutação. *Relatório Técnico RTE-03/2003 Série Ensino. Departamento de Estatística, UFMG*, 2003.
- EDGINGTON, E. S. *Randomization tests*. New York: M. [S.I.]: Dekker, 1995.
- FAUST, K.; WASSERMAN, S. Blockmodels: Interpretation and evaluation. *Social networks*, Elsevier, v. 14, n. 1-2, p. 5–61, 1992.
- FEO, T. A.; RESENDE, M. G. A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, Elsevier, v. 8, n. 2, p. 67–71, 1989.
- FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. *Journal of global optimization*, Springer, v. 6, n. 2, p. 109–133, 1995.
- FORTUNATO, S. Community detection in graphs. *Physics reports*, Elsevier, v. 486, n. 3-5, p. 75–174, 2010.
- FORTUNATO, S.; BARTHELEMY, M. Resolution limit in community detection. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 104, n. 1, p. 36–41, 2007.
- FREEMAN, L. C. A set of measures of centrality based on betweenness. *Sociometry*, JSTOR, p. 35–41, 1977.
- FU, Y.-H.; HUANG, C.-Y.; SUN, C.-T. Identifying super-spreader nodes in complex networks. *Mathematical Problems in Engineering*, v. 2015, p. 1–8, 06 2015.
- GETOOR, L.; DIEHL, C. P. Link mining: a survey. *Acm Sigkdd Explorations Newsletter*, ACM New York, NY, USA, v. 7, n. 2, p. 3–12, 2005.
- 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.
- GOOD, P. *Permutation tests: a practical guide to resampling methods for testing hypotheses*. [S.I.]: Springer Science & Business Media, 2013.
- GUIMERA, R.; DANON, L.; DIAZ-GUILERA, A.; GIRALT, F.; ARENAS, A. Self-similar community structure in a network of human interactions. *Physical review E*, APS, v. 68, n. 6, p. 065103, 2003.
- HE, X.; FENG, J.; PLANT, C. Automatically spotting information-rich nodes in graphs. In: IEEE. *2011 IEEE 11th International Conference on Data Mining Workshops*. [S.I.], 2011. p. 941–948.
- KERNIGHAN, B. W.; LIN, S. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, Wiley Online Library, v. 49, n. 2, p. 291–307, 1970.
- KHORASGANI, R. R.; CHEN, J.; ZAÏANE, O. R. Top leaders community detection approach in information networks. In: CITESEER. *4th SNA-KDD workshop on Social Network mining and Analysis*. [S.I.], 2010. p. 2319–7323.
- KREBS, V. *Books about US politics*. 2013. <<http://www-personal.umich.edu/~mejn/netdata/>>. [Último acesso em 21-Maio-2019].

- KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, Taylor & Francis Group, v. 47, n. 260, p. 583–621, 1952.
- LI, W.; SCHUURMANS, D. Modular community detection in networks. In: *Twenty-Second International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2011. p. 1366–1371.
- LUSSEAU, D.; SCHNEIDER, K.; BOISSEAU, O. J.; HAASE, P.; SLOOTEN, E.; DAWSON, S. M. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, Springer, v. 54, n. 4, p. 396–405, 2003.
- MATEUS, G. R.; RESENDE, M. G.; SILVA, R. M. Grasp with path-relinking for the generalized quadratic assignment problem. *Journal of heuristics*, Springer, v. 17, n. 5, p. 527–565, 2011.
- NASCIMENTO, M. C.; RESENDE, M.; TOLEDO, F. M. Grasp with path-relinking for the multi-plant capacitated lot sizing problem. *European Journal of Operational Research*, Citeseer, 2008.
- NEWMAN, M. E. A measure of betweenness centrality based on random walks. *Social networks*, Elsevier, v. 27, n. 1, p. 39–54, 2005.
- NEWMAN, M. E. Community detection and graph partitioning. *EPL (Europhysics Letters)*, IOP Publishing, v. 103, n. 2, p. 28003, 2013.
- PALLA, G.; DERÉNYI, I.; FARKAS, I.; VICSEK, T. Uncovering the overlapping community structure of complex networks in nature and society. *nature*, Nature Publishing Group, v. 435, n. 7043, p. 814–818, 2005.
- PASTOR-SATORRAS, R.; VESPIGNANI, A. Epidemic dynamics and endemic states in complex networks. *Physical Review E*, APS, v. 63, n. 6, p. 066117, 2001.
- PRAIS, M.; RIBEIRO, C. C. Reactive grasp: An application to a matrix decomposition problem in tdma traffic assignment. *INFORMS Journal on Computing*, INFORMS, v. 12, n. 3, p. 164–176, 2000.
- R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2018. Disponível em: <<https://www.R-project.org/>> .
- RIOS, E.; OCHI, L. S.; BOERES, C.; COELHO, I. M.; COELHO, V. N.; MLADENOVIĆ, N. A performance study on multi improvement neighborhood search strategy. *Electronic Notes in Discrete Mathematics*, Elsevier, v. 58, p. 199–206, 2017.
- SABIDUSSI, G. The centrality index of a graph. *Psychometrika*, Springer, v. 31, n. 4, p. 581–603, 1966.
- SUN, J.; FALOUTSOS, C.; PAPADIMITRIOU, S.; YU, P. S. Graphscope: parameter-free mining of large time-evolving graphs. In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2007. p. 687–696.
- TANG, L.; LIU, H. Community detection and mining in social media. *Synthesis lectures on data mining and knowledge discovery*, Morgan & Claypool Publishers, v. 2, n. 1, p. 1–137, 2010.
- WANG, Y.; DI, Z.; FAN, Y. Detecting important nodes to community structure using the spectrum of the graph. *PLoS ONE*, v. 6, n. 11, 2011.

---

WASSERMAN, S.; FAUST, K. *Social network analysis: methods and applications*. [S.l.]: Cambridge University Press, 1994.

WU, Q.; QI, X.; FULLER, E.; ZHANG, C.-Q. "follow the leader": A centrality guided clustering and its application to social network analysis. *The Scientific World Journal*, Hindawi, v. 2013, 2013.

XIA, Y.; REN, X.; PENG, Z.; ZHANG, J.; SHE, L. Effectively identifying the influential spreaders in large-scale social networks. *Multimedia Tools and Applications*, Springer, v. 75, n. 15, p. 8829–8841, 2016.

XIE, J.; KELLEY, S.; SZYMANSKI, B. K. Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*, ACM, v. 45, n. 4, p. 43, 2013.

YANG, J.; MCAULEY, J.; LESKOVEC, J. Community detection in networks with node attributes. In: IEEE. *2013 IEEE 13th International Conference on Data Mining*. [S.l.], 2013. p. 1151–1156.

ZACHARY, W. W. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, University of New Mexico, v. 33, n. 4, p. 452–473, 1977.