

METAHEURÍSTICAS PARA A DETECÇÃO DE LÍDERES E COMUNIDADES EM REDES SOCIAIS

Francisco Jucelino Matos Junior

fjmjl@de.ufpe.br

Geiza Cristina da Silva

geiza@de.ufpe.br

Raydonal Ospina

raydonal@de.ufpe.br

Universidade Federal do Pernambuco
Av. Prof. Moraes Rego, 1235 - Cidade Universitária, Recife

RESUMO

Deteção 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 deteção de comunidades têm sido propostos na literatura. No entanto, poucos trabalhos tem 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 dois algoritmos baseados na metaheurística GRASP que utilizam adaptações do LCDA na fase de construção. Os resultados experimentais mostram que, a qualidade da solução para a deteção de líderes e comunidades em redes sociais com os algoritmos propostos performam eficientemente e são tão competitivos como os relatados atualmente na literatura.

PALAVRAS CHAVE. Redes sociais, Deteção de Comunidades, Deteção de líderes em comunidades.

Teoria e Algoritmos em Grafos (TAG). Metaheurísticas (MH).

ABSTRACT

Detection of communities is a subject that has fascinated the scientific community by the need to control the consumption and demand of resources and infrastructures in complex systems such as social networks and its massive generation of information in the current *Big Data* paradigm. 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 greater ability to spread information and ideas. Many community detection algorithms have been proposed in the literature. However,

few papers have addressed the problem of selecting the leaders of these communities. In this sense, algorithms such as *Leader Rank*, LCDA 1 and LCDA 2 have been developed for this purpose. In this paper, we propose two algorithms based on the Greedy Randomized Adaptive Search Procedure (GRASP) which use LCDA adaptations in the construction phase. The experimental results show that the solution quality for the detection of leaders and communities in social networks with the algorithms proposed are efficient and are as competitive as those currently reported in the literature.

KEYWORDS. Social networks. Community detection. Leader Detection.

Theory and Algorithms in Graphs (TAG), Metaheuristics (MH)

1. Introdução

Sistemas complexos interagentes geralmente podem ser modelados através de redes complexas. O sistema pode ser de qualquer natureza, como por exemplo, uma rede de computadores ou uma rede social entre um grupo de pessoas. As redes, por sua vez, podem ser representadas por meio de grafos, i.e., uma estrutura geométrica contendo um conjunto de nós e arcos que expressam a conexão entre os nós. No caso das redes sociais, as pessoas participantes do grupo são os nós do grafo e a interação de cada um em relação ao outro pode ser expressa por meio dos arcos.

Considera-se aqui que a rede complexa é representada por um grafo não direcionado e não ponderado, $G = (V, E)$, com V sendo o conjunto nós. Cada nó em G representa um elemento do conjunto de dados e $|V| = n$ representa a quantidade de nós em G . E é o conjunto de arestas com $|E| = m$ representando o número de 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, é considerado que $a_{ij} = 1$, se existir uma relação entre os nós i e j , caso contrário, $a_{ij} = 0$.

Uma comunidade pode ser definida como um conjunto de vértices que possuem propriedades comuns ou desempenham funções similares dentro da rede [Alves et al., 2014]. Problemas de agrupamento (clusterização), uns dos mais explorados na literatura, consistem na identificação de *clusters* 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 clusters 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 et al., 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., 2014].

Na literatura é encontrado que, de forma geral, os algoritmos são utilizados para detectar comunidades e líderes simultaneamente. Em [Ahajjam et al., 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 em [Ahajjam et al., 2018]. 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.

A metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) [Feo e Resende, 1989, 1995] é uma metaheurística iterativa, na qual cada iteração apresenta uma fase de

construção seguida de busca local. Na fase de construção, se constrói uma solução usando um algoritmo adaptativo aleatório guloso que é associado a uma função de custo c . Uma vez obtida uma solução viável, sua vizinhança é investigada até que um mínimo/máximo local seja encontrado durante a fase de busca local. A melhor solução global é retornada como resultado.

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 para a detecção de líderes de comunidades em redes. Uma extensa bateria de testes é realizada em duas das redes sociais mais utilizadas na literatura: *Zachary's Karate Club* e *American College Football*. Os resultados computacionais demonstram que os algoritmos são promissores para obter soluções de alta qualidade para o problema abordado.

O resto do trabalho está organizado da seguinte forma: A seção 2 contém o referencial teórico utilizado neste trabalho, em que são discutidos os algoritmos LCDA 1 e 2 e aspectos gerais da metaheurística GRASP. Na seção 3 são discutidos os métodos propostos neste trabalho. Na seção 4 testes são realizados para avaliar a performance dos algoritmos propostos. Finalmente, na seção 5 são discutidas as conclusões e considerações finais do trabalho.

2. Referencial Teórico

A detecção de líderes e comunidades em redes sociais é fundamental para compreender como ocorre o fluxo da informação dentro da rede. A detecção de líderes implica na determinação dos nós influentes baseada numa métrica de importância dentro da estrutura da comunidade e a detecção das comunidades (*cluster*) revela como as comunidades afetam as decisões e o comportamento de um indivíduo dentro do *cluster*. Existem diversos tipos de algoritmos de detecção de líderes e comunidades.

[Khorasgani et al., 2010] desenvolveram um algoritmo tipo *k-means*, em que k nós são detectados aleatoriamente e os demais nós são aglomerados aos seus líderes mais próximos. Outro algoritmo de detecção de nós e de líderes em redes complexas foi proposto por [Kernighan e Lin, 1970] e foi baseado no particionamento de grafos no qual é feita uma busca nas seções do grafo a fim de minimizar o número de arestas entre as partições, trocando vértices entre essas partições.

Pode-se destacar o trabalho de [Ahajjam et al., 2015] no qual propôs o algoritmo *LeaderRank* que detecta comunidades a partir dos líderes identificados utilizando medidas de vizinhança. Como continuidade do trabalho e extensão do algoritmo *LeaderRank* os mesmos autores [Ahajjam et al., 2018] desenvolveram os algoritmos LCDA 1 e 2. Nestes, medidas de similaridade são utilizadas para aglomerar nós em uma comunidade. Os algoritmos propostos nesse trabalho terão seus resultados comparados com estes algoritmos. Uma descrição mais completa dos métodos pode ser encontrada na seção 2.1.

Conforme discutem Ahajjam et al. [2018], na literatura de ciências da computação, as metodologias utilizadas na análise de influência dos usuários em redes sociais e a 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, nos nós individuais (por exemplo, centralidade de grau). Os métodos de detecção de comunidades também podem ser divididos em duas categorias: particionamento e classificação de grafos. Uma desvantagem ao utilizar o método de particionamento está na necessidade de um conhecimento *a priori* do número de grupos a ser detectado [Newman,

2013; Bader et al., 2013; Fortunato, 2010]. Os métodos de classificação são utilizados, também, para particionar o grafo tomando como base uma medida de similaridade entre as partições [Fortunato, 2010; Yang et al., 2013; Wu et al., 2013]. Entretanto, esses métodos também necessitam de um conhecimento prévio do número de grupos. Segundo [Yang et al., 2013], para contornar esse problema, basta em princípio, selecionar uma métrica que melhore a classificação dos dados.

Alguns desenvolvimentos na área propõem a combinação híbrida de detecção de comunidades e líderes, como por exemplo, o algoritmo *Leader-Follower* [Wu et al., 2013], em que cada comunidade é formada por um líder e pelo menos um “seguidor leal”. Aqui, um membro só tem conexões com membros da mesma comunidade. Entretanto, o algoritmo produz comunidades com poucos membros (dois ou um nó(s)) e a sua remoção causa perda de informação.

[Wu et al., 2013] propuseram um agrupamento para detecção de líderes e comunidades baseado em medidas de centralidade em que o vértice com maior centralidade é escolhido como ponto inicial. A técnica é realizada em três etapas: i) agrupar os vértices do grafo G em grupos, ii) grupos com grande porcentagem de sobreposição são mesclados e iii) são contraídos esses vértices nos mesmos grupos para formar um novo vértice [Zhou et al., 2014].

Trabalhos como os de [Khorasgani et al., 2010; Wang et al., 2011] usam medidas de centralidade globais e locais para identificar efetivamente os líderes em grandes redes sociais.

2.1. Algoritmos LCDA

A detecção de líderes e comunidades em redes é utilizada quando o interesse do pesquisador reside na identificação de algum tipo de comportamento entre os indivíduos em estudo. Segundo [Ahajjam et al., 2018], em redes sociais, apenas um pequeno conjunto de membros de uma sociedade são detentores de características que os destacam dos demais membros pela sua capacidade de transmitir informações. Considerando que a comunidade com seus indivíduos possa ser representada por uma rede complexa a detecção dos nós influentes responsáveis pela difusão da informação torna-se eminente na determinação de comunidades em torno deles.

Segundo [Xia et al., 2014], as medidas de centralidade [Wasserman e 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 et al., 2018] propõem dois algoritmos, *leader-community detection algorithm* 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:

Etapas 1: Detecção de Líderes: Baseando-se em estudos sociométricos, [Ahajjam et al., 2018] afirmam que líderes poderosos são aqueles que foram escolhidos por indivíduos poderosos, enquanto que o líder popular será escolhido por indivíduos mais propensos a serem “comuns”. Desta forma, os autores sugerem o uso de alguma medida de centralidade global para a detecção dos nós líderes em uma rede. Desta forma, na construção dos algoritmos LCDA 1 e 2, [Ahajjam et al., 2018] adotam a medida de centralidade de autovetor. 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:

$$c_{Eig}(v_k) = x_k, \quad (1)$$

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.

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. É utilizada a centralidade de autovetor, sendo então o líder L_1 o nó com maior autovalor na matriz de adjacência A .

Etapla 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 (é utilizado $k \geq 0,5$). Como é suposto que o grafo é não direcionado e não ponderado, a similaridade de borda *Hub Pressed Index* (HPI) foi utilizada (maiores detalhes ver, Ahajjam et al. [2018]) e é definida por:

$$SIM_{HPI}(X, Y) = \frac{|\Gamma(X) \cap \Gamma(Y)|}{\min[\kappa_X, \kappa_Y]}, \quad (2)$$

em que $\Gamma(X)$ denota o conjunto de vizinhos do nó X e κ_X é o grau do nó X .

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.

2.2. Modularidade

Para avaliar a qualidade da detecção de comunidades em grafos, pode-se utilizar a função de modularidade [Clauset et al., 2004], definida por:

$$Q = \frac{1}{2 \times m} \sum_{ij} \left(Adj_{(i,j)} - \frac{k_i \times k_j}{2 \times m} \right) \times \delta(C_i, C_j), \quad (3)$$

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.

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). 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].

2.3. GRASP

Uma metaheurística é entendida como um procedimento heurístico projetado para encontrar, gerar ou selecionar uma heurística (algoritmo de busca parcial) que pode fornecer uma solução suficientemente boa para um problema de otimização, quando há, por exemplo, a ausência de informações completas ou recursos computacionais limitados [Balamurugan et al., 2015; Bianchi et al., 2009]. Entretanto, metaheurísticas, de forma geral para problemas de otimização, não trazem nenhuma garantia de que uma solução global ótima será encontrada [Blum e Roli, 2003].

Diversas metaheurísticas implementam alguma forma de otimização estocástica, de modo que a solução encontrada depende fortemente da geração de variáveis pseudo-aleatórias [Bianchi et al., 2009]. Em otimização combinatória, sobre um grande conjunto de soluções viáveis, metaheurísticas podem frequentemente encontrar boas soluções com menos esforço computacional do que algoritmos de otimização exatos e métodos iterativos [Blum e Roli, 2003]. Por outro lado, segundo [Blum e Roli, 2003], diversos trabalhos sobre metaheurísticas são de natureza experimental o que sugere que é uma ampla linha de pesquisa a ser desenvolvida para encontrar resultados teóricos que permitam avaliar a convergência a ótimos globais.

Dentre as metaheurísticas presentes na literatura, pode-se destacar o GRASP (*Greedy Randomized Adaptive Search Procedure*) [Feo e Resende, 1989, 1995], uma metaheurística iterativa, em que cada iteração consiste de duas fases: i) construção e ii) busca local. Na fase de construção, uma solução é construída por meio de um algoritmo adaptativo aleatório guloso e associado a uma função de custo c . Dada uma solução viável obtida na construção, a busca local visa avaliar um conjunto de vizinhos até que um mínimo/máximo local seja encontrado. A melhor solução global é mantida como resultado. Ou seja, a aplicação do GRASP consiste em criar uma solução inicial e depois efetuar uma busca local para melhorar a qualidade da solução.

Etapa 1: Construção: A estratégia de construção de uma solução GRASP consiste na definição de um critério de avaliação dos elementos que podem ser inseridos na solução para o problema de otimização. O critério adapta-se à solução já construída, de forma que a valoração dos elementos varia 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 formado por todos os elementos do conjunto E que possam ser incorporados na solução parcial a ser construída. A seleção do próximo elemento para incorporação 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: $c(e) \in [c_{\min}, c_{\min} + \alpha (c_{\max} - c_{\min})]$. Note que, se $\alpha = 0$ corresponde a um algoritmo puramente guloso, enquanto se $\alpha = 1$ 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 α para a metaheurística GRASP. Por isso um estudo detalhado do funcionamento deste parâmetro é apresentado na seção 4.

As etapas acima são repetidas até que a solução esteja completa. Cabe destacar, que a construção aleatória gulosa nem sempre é capaz de produzir uma solução viável. Pode ser necessário aplicar um procedimento de “reparo” à solução para obter a sua viabilidade. Exemplos de procedimentos de reparo podem ser encontrados em [Duarte et al., 2007, 2006; Mateus et al., 2011; Nascimento et al., 2008].

Etapas 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. O algoritmo termina quando nenhuma solução melhor é encontrada na vizinhança [Feo e Resende, 1995]. A velocidade e a efetividade do procedimento de busca local depende de vários aspectos, a saber: estrutura de vizinhança, busca de vizinhança, estratégia de avaliação da função custo nos vizinhos e a própria solução inicial. A fase de construção desempenha um papel muito importante em relação a este último aspecto, construindo soluções de partida de alta qualidade para a busca local, em que vizinhanças simples são geralmente utilizadas. A busca local pode ser implementada usando uma estratégia de *melhor vizinho* ou de *primeiro de melhor* [Feo e Resende, 1995].

Na estratégia de melhor vizinho, todos os vizinhos são investigados e o melhor deles é selecionado para compor a solução, que será a solução incumbente na próxima iteração. No caso da estratégia de primeira de melhor, o primeiro vizinho cujo valor de 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 prematura para um mínimo local é mais provável de ocorrer com uma estratégia de melhor vizinho.

3. Algoritmos Propostos

Neste trabalho são propostos dois algoritmos GRASP a partir da adaptação dos algoritmos LCDA 1 e 2 para a fase de construção e um novo algoritmo de busca local. Os algoritmos, que chamaremos de LCDA-GRASP 1 e 2, serão descritos em duas etapas: construção e busca local. Na etapa de construção inicialmente é feita a detecção dos nós líderes e, em seguida, a detecção das comunidades.

Etapa 1: Algoritmo de construção da solução inicial:

1. Detecção de Líderes:

Para detectar os líderes utilizamos como função gulosa a mesma medida de centralidade dos algoritmos LCDA 1 e 2: **a centralidade de autovetor** (Equação 1).

Considere $l(e)$ a medida de centralidade, apresentada na (Equação 1), a LRC é formada pelos nós candidatos à solução e de maiores valores de acordo com $l(e)$, já que nosso objetivo é de encontrar o nó com maior liderança para a comunidade. Assim, os elementos tais que $l(e) \in [l_{\min} + \alpha(l_{\max} - l_{\min}), l_{\max}]$ são inseridos em ordem decrescente na LRC.

Um elemento da LRC é selecionado aleatoriamente para ser o nó líder.

2. Detecção de Comunidades:

A partir do líder determinado, os membros da comunidade são escolhidos. Para tanto, conforme apresentado na descrição dos algoritmos LCDA 1 e 2 (seção 2.1), a medida **similaridade HPI** é utilizada como função gulosa.

Para cada nó candidato, ou seja, ainda não adicionado a alguma comunidade, seja $c(e)$ a medida de similaridade HPI a ser calculada. Os elementos cujos valores de similaridade $c(e)$ tais que $c(e) \in [c_{\min} + \alpha(c_{\max} - c_{\min}), c_{\max}]$ 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.

Os algoritmos LCDA-GRASP 1 e 2 permanecem com a mesma diferença dos algoritmos LCDA 1 e LCDA2 originais: no segundo, a função gulosa de centralidade de nós é recalculada após a construção de cada comunidade com os nós restantes.

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 de primeiro de melhora, conforme definido em 2.3.

Ao utilizar essas metodologias, os algoritmos podem obter soluções em que uma ou mais comunidades podem ter mais de um líder ou, uma ou mais comunidades, 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 os 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 obtiver a maior modularidade.

4. Resultados Computacionais

Nesta seção são apresentados os resultados obtidos pelos algoritmos propostos para as seguintes instâncias.

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;

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 e Newman, 2002]

Os algoritmos propostos foram implementados usando o software estatístico R [R Core Team, 2018] e compilado na versão 8.2.0 utilizando o pacote *igraph* para leitura dos grafos e o pacote *parallel* para habilitar o processamento paralelo da API. Os testes foram realizados em um computador Intel i7-8550 de 1.8GHz 8 núcleos e 16GB de memória RAM rodando Windows 10.

Os resultados apresentados dizem respeito a 10 execuções independentes dos algoritmos LCDA-GRASP 1 e 2 para cada instância utilizando os seguintes parâmetros do GRASP, definidos empiricamente, $\alpha = 0,3$ 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 a quantidade de grupos/comunidades (*ng*) detectadas. Além disso, os resultados obtidos são comparados com os algoritmos da literatura LCDA1 e 2 [Ahajjam et al., 2018], *LeadersRank* 1 e 2 [Ahajjam et al., 2015].

Analisando a Tabela 1 podemos observar que, para a rede social *Zachary's karate club*, as modularidade obtida 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].

Tabela 1: Comparação entre modularidades dos algoritmos propostos e da literatura.

Algoritmos/Redes Sociais	<i>Zachary's karate club</i>	<i>American College football</i>
<i>LeaderRank</i> 1	0,3180	0,3780
LCDA 1	0,3266	0,5162
LCDA-GRASP 1	0,4198	0,6046
<i>LeaderRank</i> 2	0,3014	0,3750
LCDA 2	0,3266	0,4842
LCDA-GRASP 2	0,4198	0,6046

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

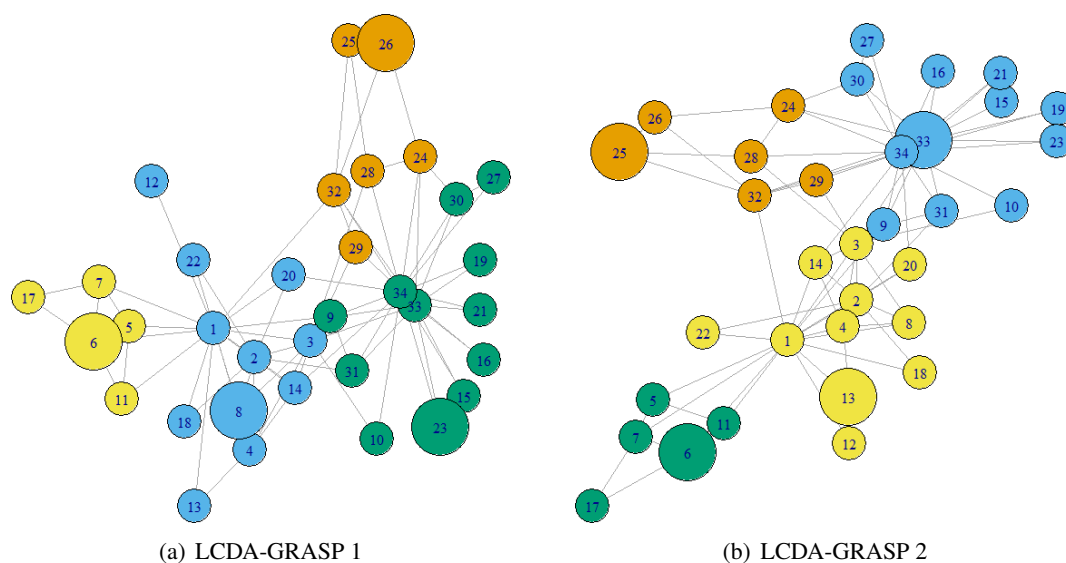


Figura 1: Líderes e comunidades detectadas para a rede social *Zachary's karate club*.

Na Tabela 2 são apresentados os resultados em relação ao tempo de processamento dos algoritmos propostos na literatura LCDA 1 e 2. São apresentados os tempos médios de processamento das 10 execuções para cada instância estudada. Podemos observa-se que o LCDA-GRASP 2 é o algoritmo com maior tempo de execução comparado ao LCDA-GRASP 1. Os tempos de processamento não foram informados [Ahajjam et al., 2018] e portanto não são aqui reproduzidos.

Tabela 2: Tempo de execução para cada rede social (em segundos).

Redes Sociais/Algoritmos	LCDA-GRASP 1	LCDA-GRASP 2
<i>Zachary's karate club</i>	2,7	2,8
<i>American College football</i>	52,5	53,2
Tempo total médio de execução	27,6	28,0

5. Considerações finais

Neste trabalho foram discutidos conceitos e algoritmos utilizados na literatura para a detecção de líderes e/ou comunidades. Dois algoritmos baseados na metaheurística GRASP foram propostos. Os resultados experimentais indicam que os algoritmos proposto apresentam medidas

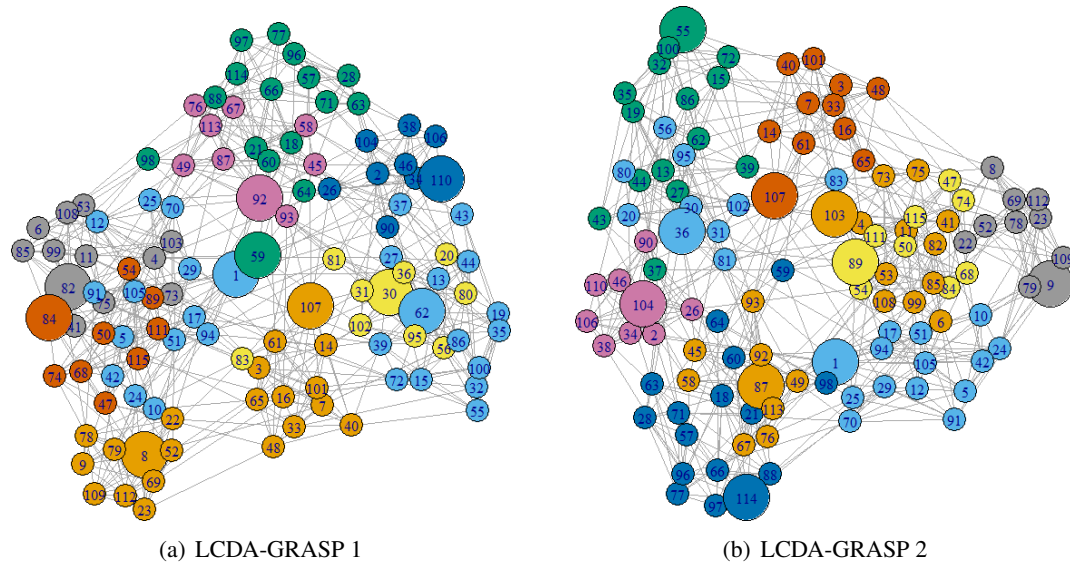


Figura 2: Líderes e comunidades detectadas para a rede social *American college football*.

de modularidade competitivas, quando comparados a métodos discutidos na literatura. Como propostas de trabalhos futuros pode se destacar o refinamento na análise ao inserir métodos estatísticos para comparar os algoritmos, assim como, a qualidade de seus resultados para as diferentes combinações dos parâmetros dos algoritmos.

Referências

- Agarwal, G. e Kempe, D. (2008). Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B*, 66(3):409–418.
- Ahajjam, S., El Haddad, M., e Badir, H. (2015). Leadersrank: Towards a new approach for community detection in social networks. In *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, p. 1–8. IEEE.
- Ahajjam, S., El Haddad, M., e Badir, H. (2018). A new scalable leader-community detection approach for community detection in social networks. *Social Networks*, 54:41–49.
- Almeida, V. A. F. d. (2011). Detecção de comunidades em grafos multicamada muito grandes. Master's thesis, Universidade Federal de Minas Gerais.
- Alves, I., Oliveira, C. S., e de Moura Brito, J. A. (2014). An analysis of the problem of community detection in networks. *Sistemas & Gestão*, 9(4):566–574.
- Atay, Y., Koc, I., Babaoglu, I., e Kodaz, H. (2017). Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms. *Applied Soft Computing*, 50:194–211.

- Bader, D. A., Meyerhenke, H., Sanders, P., e Wagner, D. (2013). *Graph partitioning and graph clustering*, volume 588. American Mathematical Soc.
- Balamurugan, R., Natarajan, A., e Premalatha, K. (2015). Stellar-mass black hole optimization for biclustering microarray gene expression data. *Applied Artificial Intelligence*, 29(4):353–381.
- Bianchi, L., Dorigo, M., Gambardella, L. M., e Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2):239–287.
- Blum, C. e Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3):268–308.
- Bonacich, P. e Lloyd, P. (2001). Eigenvector-like measures of centrality for asymmetric relations. *Social networks*, 23(3):191–201.
- Clauset, A., Newman, M. E., e Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, 70(6):066111.
- Duarte, A. R., Ribeiro, C. C., e Urrutia, S. (2007). A hybrid heuristic to the referee assignment problem with an embedded mip strategy. In *International Workshop on Hybrid Metaheuristics*, p. 82–95. Springer.
- Duarte, A. R., Ribeiro, C. C., Urrutia, S., e Haeusler, E. H. (2006). Referee assignment in sports leagues. In *International Conference on the Practice and Theory of Automated Timetabling*, p. 158–173. Springer.
- Feo, T. A. e Resende, M. G. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, 8(2):67–71.
- Feo, T. A. e Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133.
- Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3-5):75–174.
- Fu, Y.-H., Huang, C.-Y., e Sun, C.-T. (2015). Identifying super-spreader nodes in complex networks. *Mathematical Problems in Engineering*, 2015:1–8.
- Girvan, M. e Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.
- Kernighan, B. W. e Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307.
- Khorasgani, R. R., Chen, J., e Zaiane, O. R. (2010). Top leaders community detection approach in information networks. In *4th SNA-KDD workshop on Social Network mining and Analysis*. Citeseer.

- Mateus, G. R., Resende, M. G., e Silva, R. M. (2011). Grasp with path-relinking for the generalized quadratic assignment problem. *Journal of heuristics*, 17(5):527–565.
- Nascimento, M. C., Resende, M., e Toledo, F. M. (2008). Grasp with path-relinking for the multi-plant capacitated lot sizing problem. *European Journal of Operational Research*.
- Newman, M. E. (2013). Community detection and graph partitioning. *EPL (Europhysics Letters)*, 103(2):28003.
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL <https://www.R-project.org/>.
- Wang, Y., Di, Z., e Fan, Y. (2011). Detecting important nodes to community structure using the spectrum of the graph. *PloS ONE*, 6(11).
- Wasserman, S. e Faust, K. (1994). *Social network analysis: methods and applications*. Cambridge University Press.
- Wu, Q., Qi, X., Fuller, E., e Zhang, C.-Q. (2013). “follow the leader”: A centrality guided clustering and its application to social network analysis. *The Scientific World Journal*, 2013.
- Xia, Y., Ren, X., Peng, Z., Zhang, J., e She, L. (2014). Effectively identifying the influential spreaders in large-scale social networks. *Multimedia Tools and Applications*, 75(15):8829–8841.
- Yang, J., McAuley, J., e Leskovec, J. (2013). Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining*, p. 1151–1156. IEEE.
- Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473.
- Zhou, J., Zhang, Y., e Cheng, J. (2014). Preference-based mining of top-k influential nodes in social networks. *Future Generation Computer Systems*, 31:40–47.