

## **UMA HEURÍSTICA ESPECTRAL BASEADA EM COLÔNIA DE FORMIGAS PARA DETECÇÃO DE COMUNIDADES EM REDES**

**Thaise M. Quitério**

ICT - Instituto de Ciência e Tecnologia – Universidade Federal de São Paulo  
Rua Talim, 330. CEP 12231-280. Phones: +55 12 3309-9595 / +55 12 3921-5717  
São José dos Campos – SP – Brasil  
thaise.quiterio@unifesp.br

**Mariá C. V. Nascimento**

ICT - Instituto de Ciência e Tecnologia – Universidade Federal de São Paulo  
Rua Talim, 330. CEP 12231-280. Phones: +55 12 3309-9595 / +55 12 3921-5717  
São José dos Campos – SP – Brasil  
mcv.nascimento@unifesp.br

### **RESUMO**

Determinar comunidades em redes, grupos bem relacionados de vértices em um grafo, é um problema de alta relevância em diversas áreas de conhecimento. As redes sociais, inspiração do termo comunidades, são exemplos de objetos de aplicações dessa abordagem. A tarefa de detectar comunidades em redes tem sido realizada utilizando diversas abordagens, dentre as quais destaca-se o problema de maximização da modularidade de comunidades. Métodos heurísticos são mais frequentemente usados para abordar esse problema devido à sua natureza combinatorial. Neste artigo, nós propomos um algoritmo de otimização por colônia de formigas baseando-se em uma relaxação espectral do problema de maximização da modularidade. Como resultado dos experimentos computacionais, obtivemos resultados bastante competitivos com o método proposto comparando com os melhores resultados da literatura usando grafos *benchmark*.

**PALAVRAS CHAVE.** Detecção de Comunidades em Redes, Teoria Espectral em Grafos, Otimização por Colônia de Formigas.

**Área Principal:** (indique, em ordem de prioridade a área de conhecimento de seu artigo pois o sistema JEMS coloca em ordem alfabética)

### **ABSTRACT**

Finding communities in networks, i.e., sets of “related” vertices of a graph, is a problem of significant relevance in various research areas. Social networks, from where the nomenclature community arises, are examples of objects of applications of this approach. Nowadays, the task of detecting communities in networks has been performed using a wide range of approaches, specially by the modularity maximization problem. Heuristic methods are mostly used to tackle this problem due to its combinatorial nature. In this paper, we propose an algorithm based on the ant colony optimization hybridized with a spectral relaxation of the modularity maximization problem. As a result of the computational experiments, the proposed method achieved very competitive results comparing with the best results from literature using benchmark graphs.

**KEYWORDS.** Detection Community in Networks, Spectral Graph Theory, Ant Colony Optimization. **Main Area:** (inform by priority the area of the article because JEMS system makes the classification alphabetically)

## 1. Introdução

Redes<sup>1</sup> têm sido objeto de investigação em diferentes áreas de conhecimento devido às suas características topológicas peculiares quando resultantes da modelagem de sistemas reais [24]. Exemplos delas são as redes sociais, que modelam o relacionamento de indivíduos, sendo esses representados pelos vértices da rede e, suas interações, pela existência de arestas entre pares de vértices. Dentre as principais características dessas redes, podemos destacar sua estrutura com tendência de agrupamento, ou seja, grupos de vértices (ou *comunidades*) densamente conectados entre si com poucas conexões com vértices de outros grupos. Para estudar essas redes, modelos probabilísticos foram propostos para modelá-las, como, por exemplo, o modelo de redes de pequeno mundo [29].

Em particular, para redes sociais, determinar suas comunidades permite explicar o comportamento de grupos de indivíduos baseando-se em suas interações com outros indivíduos, [14]. O estudo do problema de *detecção de comunidades em redes* tornou-se foco de intenso estudo de pesquisadores das áreas de física, matemática e computação desde a proposta de [15]. Nela, os autores desenvolvem uma formalização do conceito de comunidades com a introdução de uma medida baseada em mecânica estatística, conhecida por *modularidade*. O problema de detecção de comunidades em redes (também conhecido como agrupamento em grafos).

As pesquisas desenvolvidas para o problema de detecção de comunidades em redes envolvem, principalmente, novos métodos, em sua maioria, heurísticos, para a solução do problema de encontrar comunidades (ou partições do conjunto de vértices de uma rede). Métodos que lidem com redes de grande porte representam um dos maiores desafios desse tema. Além disso, de acordo com [24], existe a necessidade de encontrar formalizações matemáticas a respeito da estrutura das comunidades encontradas por meio das medidas existentes para esse fim para que se tenham maiores informações a respeito do tipo de comunidades geradas e limitações dessas.

Neste artigo, desenvolvemos uma nova metodologia para o problema de maximização da modularidade. Consistindo de uma meta-heurística baseada na otimização por colônia de formigas, [9], aqui chamada de ACO, abordamos uma hibridização com uma heurística espectral. Esse tipo de metodologia visa uma otimização da ACO com a hibridização por um método eficiente para lidar com redes de grande porte.

## 2. O problema de maximização de modularidade

Seja  $G = (V, E)$  um grafo, em que  $V(G)$  é o seu conjunto de vértices, e  $E(G)$  de arestas. O conjunto de vértices será representado por números inteiros de 1 a  $|V(G)|$ . Cada aresta  $e \in E(G)$  é definida por uma tupla  $e := (i, j)$ , em que  $i$  e  $j$  pertencem ao conjunto  $V(G)$  e são conhecidas como pontas de  $e$ . O grau de um vértice  $i$  ( $d_G(i)$ ) é o número de arestas que tem  $i$  como ponta.

A modularidade é uma medida que avalia a qualidade de uma dada partição  $\pi = \{C_1, C_2, \dots, C_k\}$  de  $V(G)$ , em que  $k \leq n$  é o número de grupos da partição<sup>2</sup>. Essa qualidade visa a tendência de agrupamento da partição avaliada e, dada uma partição  $\pi$ , pode ser formulada por:

$$q(\pi) := \frac{1}{2m} \sum_{C_k \in \pi} \sum_{i, j \in C_k} (a_{ij} - p_{ij}) \quad (1)$$

em que  $n = |V(G)|$ ,  $m = E(G)$ ,  $a_{ij}$  é o número de arestas entre os vértices  $i$  e  $j$ , e  $p_{ij}$  é o número esperado de arestas entre os vértices  $i$  e  $j$  em um grafo aleatório. Para calcular  $p_{ij}$ , considere que um vértice  $i$  de  $G$  tem  $d_G(i)$  arestas diferentes na qual ele é ponta. Portanto, a soma  $\sum_{i=1}^n d_G(i)$  irá resultar em  $2m$ . Logo, podemos construir um grafo aleatório  $H$  tal que  $V(H) = V(G)$  e que tenha

<sup>1</sup>Neste artigo, redes são definidas como estruturas matemáticas idênticas aos grafos.

<sup>2</sup>A definição de partição usada neste artigo é a usual, na qual dado um conjunto  $D$  e  $k$  subconjuntos  $B_i \subset D$ , com  $k \leq |D|$ ,  $\bigcap_{i=1}^k B_i = \emptyset$  e  $\bigcup_{i=1}^k B_i = D$ .

a mesma sequência de graus de vértices que  $G$ . A ligação entre pares de vértices é definida por

$$\begin{aligned} P[(i, j) \in E(H)] &:= P[\text{escolher aresta incidente a } i] \cdot P[\text{escolher aresta incidente a } j] \\ &= \frac{d_G(i)}{2m} \cdot \frac{d_G(j)}{2m}. \end{aligned}$$

Dada esta probabilidade de existência de aresta ente os vértices  $i$  e  $j$  em  $H$ , o número esperado de arestas entre os vértices  $i$  e  $j$  é

$$p_{ij} = 2P[(i, j) \in E(H)] = \frac{d_G(i)d_G(j)}{2m}. \quad (2)$$

Portanto a equação (1) se torna

$$q(\pi) := \frac{1}{2m} \sum_{C_k \in \pi} \sum_{i, j \in C_k} \left( a_{ij} - \frac{d_G(i)d_G(j)}{2m} \right). \quad (3)$$

Para determinar comunidades em redes por meio dessa medida, os trabalhos da literatura visam o problema de maximização da modularidade de redes, que consiste em encontrar comunidades com a maior modularidade. Segundo [5] um problema de decisão para ele é da classe NP-difícil. Por esse motivo, os algoritmos existentes na literatura para a sua solução são, em sua maioria, heurísticas, dentre as quais destacam-se, metaheurísticas [17, 25, 1, 21], heurísticas [22, 8, 27, 30, 3], algoritmos multi-níveis [23], dentre outros. Embora os mais recentes algoritmos sejam bastante eficientes, métodos mais robustos são necessários para encontrar melhores soluções para problemas de grande porte. Métodos exatos não têm sido muito explorados, [2], principalmente devido às grandes dimensões das redes testadas por algoritmos de agrupamento.

Neste artigo, comparamos o algoritmo proposto, baseado na maximização da modularidade, com os melhores resultados encontrados na literatura.

### 3. Metodologia

Como método heurístico para o problema de maximização da modularidade, nós propomos, neste artigo, uma heurística baseada na otimização por colônia de formigas e em uma relaxação espectral de [22]. O algoritmo proposto é composto por uma inicial relaxação espectral, que é um problema de ordenação linear [7], para a qual é desenvolvida uma solução heurística por uma estratégia de otimização por colônia de formigas. Após isso, uma partição é gerada a partir da ordenação encontrada pela estratégia anterior por meio de um algoritmo de programação dinâmica adaptado de [28]. A heurística proposta é apresentada no Algoritmo 1.

No Algoritmo 1, considere as funções *Modularidade*, *Jacobi*, *Escale* e *Inicialize*, respectivamente, uma função para calcular a matriz de modularidade  $B$ , uma função para computar os autovetores e autovalores de uma matriz, por meio do método de Jacobi [6], uma função que escala a matriz de autovetores, explicada na Seção 3.1, e uma função de inicialização da matriz de feromônio  $\tau = [\tau_{ij}]_{n \times n}$  com valor zero, e da matriz dependente do valor da modularidade  $\eta = [\eta_{ij}]_{n \times n}$ , que inicialmente recebe o valor de uma matriz unitária de mesma dimensão. Além disso, as constantes *MaxIteracoes* e *n<sub>formigas</sub>*, são valores estabelecidos após testes computacionais. As funções *Construa Solução*, *Busca Local* e *Atualize Feromônio* são explicadas, respectivamente, nas Seções 3.1, 3.2 e 3.3.

#### 3.1. Estratégia de Construção da Solução

Para se ter as vantagens de algoritmos espectrais, isto é, sua rapidez, elegância e boa qualidade de soluções obtidas por meio deles, nós desenvolvemos uma heurística de construção de solução baseada na heurística espectral de [22]. Inicialmente, a heurística gulosa de [22] para criar a solução baseada em uma matriz de autovetores normalizados da matriz de modularidades de um dado grafo foi alterada para uma heurística baseada em otimização por colônia de formigas.

---

**Algoritmo 1** SpectralACO

---

```
1: Entrada: Um grafo  $G = (V, E)$ 
2:  $B \leftarrow \text{Modularidade}(G)$ ;
3:  $(U, \lambda) \leftarrow \text{Jacobi}(B)$ ;
4:  $R \leftarrow \text{Escale}(U)$ ;
5: Inicialize  $(\tau, \eta)$ ;
6:  $S_{\text{melhor}} \leftarrow \emptyset$ ;
7:  $iter \leftarrow 0$ ;
8: Enquanto ( $iter < \text{MaxIteracoes}$ ) faça
9:    $\Omega \leftarrow \emptyset$ ;
10:  Para ( $j = 1, \dots, n_{\text{formigas}}$ ) faça
11:     $S \leftarrow \text{Construa Solução}(\tau, \eta, R, \lambda)$ ;
12:     $S \leftarrow \text{Busca Local}(S)$ ;
13:    Se ( $q(S) > q(S_{\text{melhor}})$  ou  $S_{\text{melhor}} = \emptyset$ ) então
14:       $S_{\text{melhor}} \leftarrow S$ ;
15:    Fim Se
16:     $\Omega \leftarrow \Omega \cup \{S\}$ ;
17:  Fim Para
18:  Atualize Feromônio  $(\tau, \sigma_{iter}, S_{\text{melhor}})$ 
19:   $iter \leftarrow iter + 1$ ;
20: Fim Enquanto
21: Saída: A partição  $S_{\text{melhor}}$  de  $V(G)$ 
```

---

A razão disso é promover a diversidade nas soluções geradas que serão melhoradas em próximas iterações do algoritmo.

Para a compreensão da fase de construção da solução da heurística proposta, é importante o entendimento da relaxação espectral proposta por [22]. Para isso, considere a matriz de modularidade  $B = [b_{ij}]_{n \times n}$ , em que  $b_{ij} = a_{ij} - p_{ij}$ , de um grafo  $G$  e  $U = [u_{ij}]_{n \times n}$ , a matriz que representa em sua  $j$ -ésima coluna, o autovetor associado ao autovalor  $\lambda_j$  de  $B$ . Os autovalores  $\lambda_j$ , com  $1 \leq j \leq n$ , estão organizados em ordem decrescente de valor.

Seja  $R = [r_{ij}]_{n \times k}$  uma matriz de autovetores normalizados de  $U$ . O elemento de sua  $i$ -ésima linha e  $j$ -ésima coluna dessa matriz é definida por  $r_{ij} = u_{ij} \sqrt{\lambda_j - \gamma}$ , em que  $\gamma = \frac{1}{n-c} \sum_{i=c+1}^n \lambda_i$  (ajustado depois de um estudo de minimização de erros de uma aproximação de  $q(\pi^k)$ ). Cada vetor  $\vec{r}_i$ , para  $1 \leq i \leq n$ , denota um vetor  $k$ -dimensional que corresponde à  $i$ -ésima linha de  $R$ . No Algoritmo 1, esse cálculo é realizado na função `Escale`.

Nessa matriz  $R$ , a  $i$ -ésima linha corresponde ao vértice  $v_i$  de um grafo  $G$ . Eliminando os autovalores negativos da equação (pois eles contribuem negativamente na função de maximização), supondo que eles correspondem aos últimos  $n - c$  autovalores,  $q(\pi^k)$  pode ser aproximado por:  $q(\pi^k) = n\theta + \sum_{j=1}^k \|R_j\|^2$ , sendo  $\theta$ , uma constante. Como  $c \geq k$ , pois ele é um limitante superior de  $k$ , considere  $R_j = \sum_{v_i \in C_j} \vec{r}_i$ , o problema pode ser abordado pelo problema de particionamento de um vetor [22].

Então, a partir dos autovetores normalizados encontrados, uma sequência de vértices foi definida a partir do caminho definido por uma determinada formiga. A probabilidade de transição de um vértice para outro foi calculada pela equação (4), que, segundo [12], é a probabilidade mais usada para esse fim. Ao fim da construção do caminho pela formiga, ele será usado para definir a partição final.

$$p(j \in S_p | S_u) = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{k,l \in S_u} (\tau_{kl})^\alpha (\eta_{kl})^\beta} \quad (4)$$

em que  $\alpha$  e  $\beta$  são parâmetros ajustados experimentalmente e assumem valores maiores do que 1;  $S_u \subset \{1, 2, \dots, n\}$ ;  $S_p = \{1, 2, \dots, n\} \setminus S_u$  é um conjunto ordenado de vértices cuja relação de ordem é definida pela posição do caminho construído pela formiga no qual determinado vértice pertence; e  $i$  é o último vértice do conjunto ordenado  $S_p$ . Um algoritmo sumarizando os passos envolvidos na construção da solução por nossa estratégia é apresentado no Algoritmo 2.

---

**Algoritmo 2** Construa Solução

---

- 1: **Entrada:** Uma matriz de feromônio,  $\tau$ , a matriz auxiliar  $\eta$ , o vetor  $\lambda$  e a matriz  $R$  com, respectivamente, os autovalores e os autovetores normalizados da matriz de modularidade do grafo  $G = (V, E)$
  - 2:  $S_p \leftarrow \emptyset$ ;
  - 3:  $S_u \leftarrow 1, 2, 3, \dots, n$ ;
  - 4:  $j \leftarrow 1$
  - 5: **Enquanto** ( $S_u \neq \emptyset$ ) **faça**
  - 6:  $i \leftarrow \text{Sorteia}(ic, S_p, \tau, S_u, p, R)$ ;
  - 7:  $S_u \leftarrow S_u \setminus ic$ ;
  - 8:  $ic \leftarrow i$ ;
  - 9:  $S_{p_j} \leftarrow ic$ ;
  - 10:  $j \leftarrow j + 1$ ;
  - 11: **Fim Enquanto**
  - 12:  $S \leftarrow \text{Encontra Particao}(S_p)$ ;
  - 13: **Saída:** A partição  $S$  de  $V(G)$
- 

No Algoritmo 2, a função *Sorteia*, escolhe o próximo vértice a ser adicionado no conjunto ordenado  $S_p$  de acordo com as probabilidades calculadas segundo a equação (4) e na solução parcial da relaxação espectral, que é atualizada na matriz  $\eta$  até o momento.

Após a definição de uma ordenação dos vértices, então um algoritmo de programação dinâmica adaptado da versão de [28] foi utilizado para resolver heurísticamente o problema de particionamento do vetor  $S$  do Algoritmo 1.

Portanto, ao fim desse processo, teremos uma partição definida segundo o caminho construído por uma determinada formiga.

### 3.2. Busca Local

A estratégia de busca local usada neste trabalho foi a mesma apresentada em [21]. Ela consiste na busca na 1-vizinhança de uma dada partição por partições de melhor qualidade. Para mais detalhes a respeito desta estratégia, recomendamos o leitor interessado a referência [21], que tem uma versão resumida no sítio do DIMACS Challenge de 2012<sup>3</sup>.

### 3.3. Atualização do Feromônio

Neste artigo, é proposta uma estratégia de atualização de feromônio que, segundo [10], foi a primeira proposta usada para atualização em sistemas de colônia de formigas [11]. Para tal, todas as soluções da iteração são armazenadas em um conjunto  $\Omega$ , e proporcional à qualidade e à frequência de parte das soluções desse conjunto e da taxa de evaporação de feromônio,  $\rho \in (0, 1]$ , atualizamos a matriz de feromônio  $\tau$ .

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho \sum_{k=1}^{n_{formigas}} \delta_{ij}^k \quad (5)$$

em que  $\delta_{ij}^k = q(\pi^k)$ , sendo que  $\pi^k$  é a  $k$ -ésima solução do conjunto  $\Omega$ . Tentamos outra estratégia de atualização de feromônio, considerando a melhor solução até o momento, entretanto, essa maneira atualmente é a mais bem sucedida.

---

<sup>3</sup><http://www.cc.gatech.edu/dimacs10/papers/%5B12%5D-NascimentoandPitsoulis.v3.pdf>

#### 4. Experimento Computacional

Os experimentos para este trabalho foram todos realizados em um microcomputador Intel Core I5 com 6GB de RAM e 2.6GHZ com o sistema operacional Windows 7. Os valores de parâmetros utilizados foram obtidos experimentalmente, por meio de uma estratégia que considera a razão entre frequência e tempo da melhor solução encontrada em determinados intervalos de valores dos parâmetros. Os melhores valores obtidos e usados nos experimentos para os parâmetros  $MaxIteracoes$ ,  $n_{formigas}$ ,  $\alpha$ ,  $\beta$  e  $\rho$  foram, respectivamente, 40, 5, 2, 1 e 0,5. Os grafos usados neste experimento são os grafos *benchmark*: *karate* [31] com 34 vértices e 78 arestas; *dolphins* [20] com 62 vértices e 159 arestas, *lesmis* [19] com 77 vértices e 254 arestas; *adjnoun* [22] com 112 vértices e 425 arestas; *afootball* [15] com 115 vértices e 613 arestas; *jazz* [16] com 198 vértices e 2742 arestas; *celegans\_metabolic* [13] com 453 vértices e 2025 arestas; e *email* [4] com 1133 vértices e 5451 arestas. Além disso, para cada uma dessas bases, coletamos os melhores resultados encontrados na literatura, para efeitos de comparação da nossa heurística. Os resultados obtidos neste experimento estão sumarizados na Tabela 1.

Tabela 1: Resultados dos experimentos computacionais.

Grafo	Melhor Solução		Solução Mediana		Pior Solução		Melhor Solução	
	Valor	Tempo	Valor	Tempo	Valor	Tempo	Valor	Tempo
<i>karate</i>	0,41979	0,12	0,41979	0,10	0,41978	0,11	0,41979	0,00
<i>dolphins</i>	0,52852	0,51	0,52680	0,44	0,52761	0,43	0,5285	4
<i>lesmis</i>	0,55786	0,65	0,55207	0,67	0,55618	0,65	0,56008	4,00
<i>adjnoun</i>	0,30406	5,49	0,29363	5,28	0,29811	5,55	0,311	1
<i>afootball</i>	0,60457	2,79	0,59778	2,82	0,60306	2,74	0,60457	–
<i>jazz</i>	0,44514	7,24	0,44503	6,80	0,44514	6,79	0,44514	–
<i>celegans</i>	0,42685	78,37	0,41852	73,36	0,42274	75,55	0,452	98
<i>email</i>	0,55449	1516,21	0,54536	1482,99	0,54916	1367,74	0,581	–

Como o algoritmo proposto tem uma componente aleatória, para melhor avaliação da meta-heurística proposta, o executamos 10 vezes. Na Tabela 1, apresentamos o melhor resultado, o resultado mediano e o pior resultado (com relação ao valor da modularidade) dentre essas 10 execuções, com seus respectivos tempos computacionais. Além disso, na última coluna, para fins de comparação, reportamos os melhores valores de solução encontrados na literatura e seus respectivos tempos computacionais, quando disponíveis. Na ordem em que os grafos aparecem na tabela, suas melhores soluções foram encontradas nas referências [1, 26, 21]; [1]; [1]; [21]; [1, 26]; [13]; [21] e [26].

Pelos resultados apresentados na Tabela 1, podemos observar que a atual versão do algoritmo proposto é bem competitivo, entretanto, ainda não supera os melhores resultados da literatura (considerando as soluções medianas). Os tempos computacionais foram razoáveis e ligeiramente inferiores aos dos tempos disponíveis nos melhores resultados da literatura.

Por meio de uma análise dos resultados obtidos de acordo com as comunidades geradas pela heurística proposta, observamos o comportamento de suas soluções com uma análise gráfica. Tomemos como exemplo a Figura 1. Para obter esta figura, que apresenta a representação do grafo *dolphins*, utilizamos um algoritmo para representação gráfica de grafos de [18]. Nesta figura, os vértices dos grafos, representados pelos círculos, estão preenchidos de acordo com a comunidade encontrada pelo algoritmo de otimização por colônia de formigas proposto. Por exemplo, na Melhor Solução, temos a melhor solução encontrada pelo algoritmo nas 10 execuções do algoritmo, sendo que os vértices em vermelho pertencem a uma primeira comunidade, em preto a uma segunda comunidade, em amarelo, a uma terceira comunidade, em azul, a uma quarta comunidade e cinza, a uma quinta comunidade.

Note que, na Figura 1, as diferenças significativas entre as três partições são os grupos



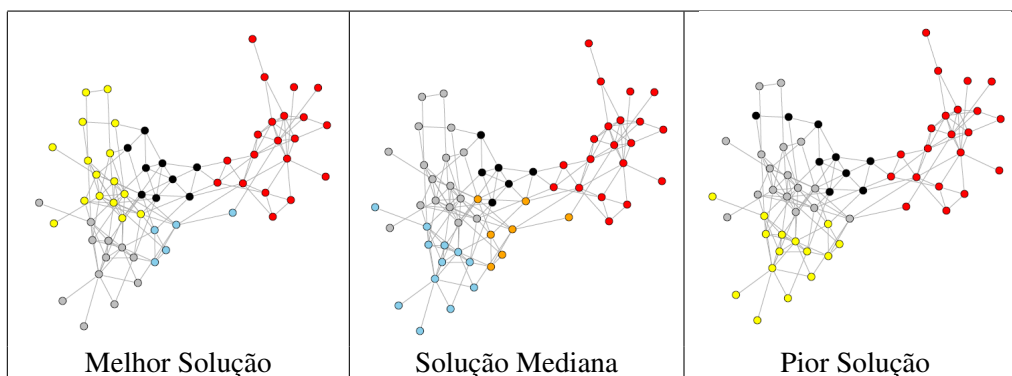


Figura 1: Figuras com, respectivamente, a melhor solução, a solução mediana e a pior solução encontradas para o grafo *dolphins*.

com menor número de vértices. A comunidade que, na melhor solução tem rótulo azul, tem apenas 5 vértices e nas duas outras partições, essa comunidade, que parece ser de fronteira, encontra-se “diluída” na Pior Solução, e com elementos de uma outra comunidade, na Solução Mediana. Uma tentativa de melhoria desse algoritmo, que, a partir da Solução Mediana não convergiu para a Melhor Solução, é considerar outra estratégia de busca local usando a 2-vizinhança.

## 5. Conclusões e Pesquisas Futuras

Neste artigo, apresentamos uma heurística para o problema de maximização de modularidade baseado em uma estratégia espectral hibridizada com a meta-heurística de otimização por colônia de formigas. Para os experimentos foram considerados grafos *benchmark* e algoritmos da literatura que, atualmente, constituem nas melhores estratégias encontradas. Como resultado, observamos que a meta-heurística proposta se mostrou bem competitiva, entretanto, não superou os resultados da literatura.

O método proposto foi desenvolvido com uma estrutura de dados simples, que, entretanto, está em fase de melhoramento para maior intensificação da estratégia. Realizamos um estudo das partições geradas, os seus padrões, e temos perspectivas de melhoramento da heurística de acordo com os resultados observados até o fim do semestre, quando acaba o projeto da aluna.

## 6. Agradecimentos

Agradecemos à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) seu apoio financeiro para o desenvolvimento deste projeto.

## Referências Bibliográficas

- [1] G. Agarwal and D. Kempe. Modularity-maximizing graph communities via mathematical programming. *European Physical Journal B*, 66:409–418, 2008.
- [2] D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, L. Liberti, and S. Perron. Column generation algorithms for exact modularity maximization in networks. *Physical Review E*, 82(4):046112, 2010.
- [3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large network. *Journal of Statistical Mechanics: Theory and Experiment*, page P10008, 2008.
- [4] M. Boguñá, R. Pastor-Satorras, and A. Arenas. Models of social networks based on social distance attachment. *Phys. Rev. E*, 70:056122, 2004.
- [5] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hofer, Z. Nikolosk, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20:172–188, 2008.

- [6] Richard L. Burden and J. Douglas Faires. *Anse Numca*. Cengage Learning, 2008.
- [7] I. Charon and O. Hudry. An updated survey on the linear ordering problem for weighted or unweighted tournaments. *Annals of Operations Research*, 175:107–158, 2010.
- [8] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111–[6 pages], Dec 2004.
- [9] M. Dorigo. *Optimization, learning and natural algorithms*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992. (in Italian).
- [10] M. Dorigo, M. Birattari, and T. Sttzle. Ant Colony Optimization: Artificial ants as a computational intelligence technique. IRIDIA 2006-023, Universitbre de Bruxelles, 2006.
- [11] M. Dorigo, V. Maniezzo, and A. Colorni. Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 26:29–41, 1996.
- [12] M. Dorigo and T. Sttzle. Ant colony optimization: Overview and recent advances. In M. Gendreau and Y. Potvin, editors, *International Series in Operations Research & Management Science*, volume 146 of 2nd edition. Handbook of Metaheuristics, Springer, Verlag, 2010.
- [13] J. Duch and A. Arenas. Community identification using extremal optimization. *Physical Review E*, 72:027104, 2005.
- [14] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.
- [15] M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99:7821–7826, 2002.
- [16] P. Gleiser and L. Danon. Community structure in jazz. *Advances in Complex Systems*, 6:565–573, 2003.
- [17] R. Guimerá, M. Sales-Pardo, and L.A.N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101, 2004.
- [18] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15, 1988.
- [19] D. E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. Addison-Wesley, Reading, MA, 1993.
- [20] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54:396–405, 2003.
- [21] M. C. V. Nascimento and L. O. Pitsoulis. Community detection by modularity maximization using grasp with path-relinking. Submitted to *Computers & Operations Research*, 2011.
- [22] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036–104, 2006.
- [23] A. Noack and R. Rotta. Multi-level algorithms for modularity clustering. In *Proceedings of the 8th International Symposium on Experimental Algorithms (SEA 2009)*, volume 5526 of *Lecture Notes in Computer Science*, pages 257–268. Springer, 2009.
- [24] M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the AMS*, 56:1082–1097, 2009.





- [25] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74:016110–[14 pages], 2006.
- [26] Randolph Rotta. *A Multi-Level Algorithm for Modularity Graph Clustering*. PhD thesis, Brandenburgische Technische Universität Cottbus, 2008.
- [27] P. Schuetz and A. Caffish. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E*, 77:046112, May 2008.
- [28] H. M. Wagner and T. M. Whitin. Dynamic version of the economic lot size mode. *Management Science*, 5:89–96, 1958.
- [29] D. Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.
- [30] Z. Ye, S. Hu, and J. Yu. Adaptive clustering algorithm for community detection in complex networks. *Physical Review E*, 78:046115, 2008.
- [31] W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.