

# Heurística da Colônia de Formigas para Detecção de Comunidades

Eduardo Souza Santos, Rafael de Santiago

Laboratório de Inteligência Aplicada – Universidade do Vale do Itajaí (UNIVALI)  
Campus Itajaí – 88302-901 – Itajaí – SC – Brasil

eduardo.souza@edu.univali.br, rsantiago@univali.br

**Abstract.** *Modularity maximization is a model used to detect communities. Exact algorithms can not solve the problem of maximizing modularity in polynomial time, which justifies the use of heuristic search. Ant Colony Optimization (ACO) was the meta-heuristic used in networks real world and random in order to test its feasibility. Two combinations were responsible for generating better quality solutions in a polynomial computational time, they were:  $\alpha = 0.3$ , greedy factor = 0.3, population = 5, iterations = 15, pheromone  $\in \{0.3, 0.7\}$ .*

**Resumo.** *Maximização da modularidade é um modelo utilizado para detecção de comunidades. Algoritmos exatos podem não resolver o problema da maximização da modularidade em tempo polinomial, o que justifica o uso de busca heurística. Otimização por Colônia de Formigas (OCF) foi a meta-heurística utilizada em redes do mundo real e aleatórias, a fim de testar sua viabilidade. Duas combinações foram responsáveis por gerar as soluções de melhor qualidade em um tempo computacional polinomial, elas foram:  $\alpha = 0.3$ , fator guloso = 0.3, população = 5, iterações = 15, feromonio  $\in \{0.3, 0.7\}$ .*

## 1. Introdução

Um grafo, ou rede, consiste de vértices que geralmente correspondem às entidades, e arestas, que ligam os vértices. O relacionamento entre as entidades pode carregar alguma informação, qualitativa ou quantitativa, indo além da relação entidade-entidade. Como exemplo dessa representação, Guimerà e Amaral (2005) representam modelos de redes metabólicas com enzimas, metabólitos e suas respectivas reações.

Uma propriedade que parece ser comum em muitas redes é a estrutura de comunidade: a divisão dos vértices da rede em grupos, onde entre as entidades do grupo as conexões são densas e as conexões entre grupos são esparsas. A capacidade de encontrar e analisar esses grupos pode fornecer ajuda na compreensão e visualização da estrutura de redes [Newman e Girvan 2004].

Nas mais diversas áreas de pesquisa pode-se encontrar e analisar estruturas de comunidades em redes, das quais são encontrados na literatura alguns exemplos, tais como: (i) detectar organizações criminosas em redes de telefonia móvel [Ferrara et al. 2014]; (ii) detectar e agrupar complexos de proteínas, o que permite analisar suas funções e interações [Nepusz, Yu e Paccanaro 2012]; e (iii) analisar a estrutura neurológica [Meunier et al. 2014].

Na literatura diversas abordagens são utilizadas para detecção de comunidades, porém o presente trabalho utilizará uma abordagem conhecida como *Modularity Clustering*, proposta por Newman e Girvan (2004) e Newman (2004), cujo método é baseado em um problema de otimização, onde o objetivo é maximizar a modularidade, uma medida de qualidade sobre partição em grafos.

A detecção de comunidades através de modularidade envolve condições de decisão que nem sempre podem ser resolvíveis em tempo polinomial. Brandes, Delling e Gaertler (2008) demonstraram que a localização de comunidades através da modularidade é um problema que pertence à classe *NP-Hard*, o que justifica a utilização de heurísticas.

O objetivo deste trabalho foi implementar a meta-heurística Otimização por Colônia de Formigas para o problema de detecção de comunidades através da maximização da modularidade.

Nas próximas seções serão abordados os seguintes temas: (i) a maximização da modularidade; (ii) a meta-heurística Otimização por Colônia de Formigas; (iii) o desenvolvimento do algoritmo; (iv) os experimentos realizados; (v) os resultados obtidos nos experimentos; (vi) as conclusões do trabalho.

## 2. Modularidade

A fim de avaliar a qualidade das comunidades encontradas, Newman e Girvan (2004) desenvolveram a métrica da modularidade, conforme descrito na Equação 1 para a solução  $C$ :

$$Q(C) := \frac{1}{2|E|} \sum_{u,v \in V} \left( a_{u,v} - \frac{d_u d_v}{2|E|} \right) \quad \text{Equação 1}$$

Agarwal e Kempe (2008) definem que para uma rede dada como um grafo não dirigido  $G = (V, E)$ , onde  $V$  representa o conjunto de vértices e  $E$  representa o conjunto de arestas, a matriz de adjacência de  $G$  é denotada por  $A = (a_{u,v})$ : assim,  $a_{u,v} = a_{v,u} = 1$  se  $u$  e  $v$  compartilham uma aresta, e  $a_{u,v} = 0$  caso contrário. O grau de uma aresta  $v$  é denotada por  $d_v$ , assim como o grau de uma aresta  $u$  é denotada por  $d_u$ . O número de arestas presentes no grafo é representado por  $2|E|$  [Agarwal e Kempe 2008].

A otimização da modularidade tem uma desvantagem, o limite de resolução, a impossibilidade de encontrar comunidades de tamanho pequeno, sob certas circunstâncias, dependendo da topologia da rede [Fortunato e Barthélemy 2007].

Brandes et al. (2008) demonstraram que a versão de decisão do problema da maximização da modularidade pertencente à classe de problemas *NP-Hard*. Isto significa que algoritmos eficientes para agrupamento ótimo, em tempo polinomial no tamanho do grafo para todos os grafos, são improváveis de existir. Busca-se então, desenvolver heurísticas produzindo agrupamentos com a qualidade máxima possível em tempo viável [Agarwal e Kempe 2008].

## 3. Otimização por Colônia de Formigas

Otimização por Colônia de Formigas (OCF) é uma meta-heurística para resolver problemas combinatoriais difíceis. A fonte de inspiração de OCF é o comportamento de formigas reais em depositar e seguir a trilha de feromônio – um hormônio secretado no

ambiente que permite o reconhecimento e interação entre as formigas, usado como um meio de comunicação – no ambiente [Dorigo e Stützle 2004].

Formigas artificiais usadas em OCF são procedimentos de construção de soluções estocásticas que probabilisticamente constroem uma solução iterativamente adicionando componentes a uma solução parcial, levando em conta: (i) informação heurística sobre a instância do problema a ser resolvido, se possível, e (ii) trilhas de feromônio artificiais que mudam dinamicamente em tempo de execução para refletir a experiência na busca adquirida pelos agentes [Dorigo e Stützle 2004].

Segundo Dorigo e Stützle (2004) OCF apresenta as seguintes vantagens em sua utilização: (i) um componente estocástico permite que formigas construam uma grande variedade de diferentes soluções e explorem um número maior de soluções; (ii) a experiência que as formigas obtêm na busca pode ser usada para influenciar, de um modo que lembra o reforço da aprendizagem, na construção de soluções futuras nas próximas iterações do algoritmo; e (iii) o uso de uma colônia de formigas pode dar ao algoritmo um aumento na robustez, onde o uso da coletividade de uma população de agentes tende a resolver eficazmente um problema.

#### 4. Desenvolvimento

O algoritmo OCF, foi remodelado a fim de conter uma etapa de busca gulosa e uma etapa de cálculo da modularidade. Um percentual de construção por busca gulosa foi necessário a fim de evitar que formigas gerassem apenas subótimos locais e acabassem por não explorar as melhores soluções possíveis. A busca gulosa utilizou o parâmetro  $\alpha$  que selecionou um entre os  $\alpha\%$  melhores elementos da solução.

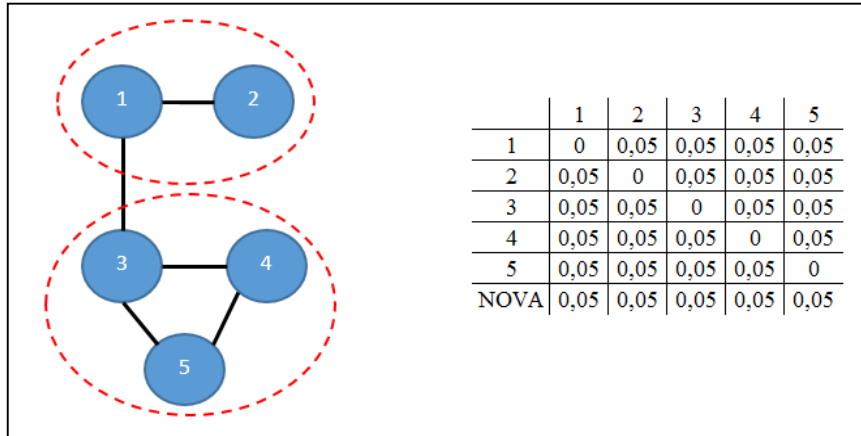
A construção por formigas utilizou um método de seleção por roleta para guiar sua construção através da matriz de feromônio, onde os maiores valores de feromônio estiveram mais propensos a serem seguidos, pois indicavam um caminho promissor.

$$\frac{1}{V^2 - d_p} \quad \text{Equação 2}$$

O feromônio depositado na matriz para a primeira iteração, foi distribuído uniformemente conforme descrito no cálculo da Equação 2. Nesta Equação o valor 1 divide o quadrado da quantidade de vértices ( $V$ ), por se tratar de um grafo não dirigido. Na diagonal principal ( $d_p$ ) este cálculo não se aplica, pois elas representam o mesmo vértice.

A matriz de feromônio, diferente da matriz de adjacência, não é uma matriz quadrada, pois contém uma linha adicional (NOVA), a qual foi responsável por indicar ao algoritmo de seleção que o vértice selecionado naquela linha, é um elemento formador de uma nova comunidade. A Figura 1 demonstra um grafo de cinco vértices com estrutura de duas comunidades juntamente com a representação da matriz de feromônio que apresenta a distribuição de feromônio na fase inicial.

A matriz de feromônio é atualizada a cada iteração para a melhor solução encontrada, até atingir o seu critério de parada. A atualização de feromônio foi a etapa responsável por armazenar o caminho das melhores soluções.



**Figura 1. Feromônio inicial para um grafo de 5 vértices**

A fim de atender ao problema proposto a Equação 3 foi criada para a etapa de atualização do feromônio, conforme segue:

$$\tau_{i,j} = \tau_{i,j} \cdot \Delta_{\tau_{i,j}} + \left( (1 - \Delta_{\tau_{i,j}}) \cdot \left( \frac{Q(s) + 1}{2} \right) \right) \quad \text{Equação 3}$$

Onde  $\tau_{i,j}$  é o feromônio do vértice  $i, j$ , a taxa de evaporação do feromônio é denotada por  $\Delta_{\tau_{i,j}}$ ,  $Q(s)$  é o valor obtido através do cálculo da modularidade sendo  $s$  a representação da melhor solução gerada.

A Figura 2 apresenta 5 iterações e demonstra passo a passo como ocorre a construção da solução para o grafo da Figura 1. O método de seleção por roleta é o responsável pela escolha dos vértices e seu agrupamento em comunidades, no exemplo não leva-se em conta etapas de construção por busca gulosa, sendo a construção exclusivamente por formigas.

Para a construção da solução consideram-se as colunas como indicadores dos vértices e as linhas os indicadores de comunidades. Na primeira iteração o vértice 1 foi selecionado aleatoriamente e indicado como sendo membro de uma nova comunidade, representado pela cor vermelha.

Conforme os vértices vão sendo adicionados à solução, as linhas que os representam vão sendo habilitadas. Deste modo, cada célula é o cruzamento de vértice que já está na solução com um vértice candidato, onde o valor deste cruzamento representa a propensão de um candidato entrar em uma mesma comunidade de um vértice já adicionado na solução.

Uma busca local monótona *Best Improvement* foi utilizada com o fim de intensificar a qualidade da solução obtida nas etapas construtivas. Utilizou-se a estratégia de 1-Vizinhança, que atua na modificação da comunidade de um vértice em relação à solução original.

A Figura 3 demonstra a etapa de atualização de feromônio para uma solução  $S = [[1, 2], [3, 4, 5]]$  com taxa de 30% de evaporação, utilizando como base os exemplos das Figuras 1 e 2.

	1	2	3	4	5	1º iteração: solucao_parcial = {[1]}
NOVA	<b>0,05</b>	0,05	0,05	0,05	0,05	
	1	2	3	4	5	2º iteração: solucao_parcial = {[1], [3]}
1	0	0,05	0,05	0,05	0,05	
NOVA	0,05	0,05	<b>0,05</b>	0,05	0,05	
	1	2	3	4	5	3º iteração: solucao_parcial = {[1], [3, 4]}
1	0	0,05	0,05	0,05	0,05	
3	0,05	0,05	0	<b>0,05</b>	0,05	
NOVA	0,05	0,05	0,05	0,05	0,05	
	1	2	3	4	5	4º iteração: solucao_parcial = {[1, 2], [3, 4]}
1	0	<b>0,05</b>	0,05	0,05	0,05	
3	0,05	0,05	0	0,05	0,05	
4	0,05	0,05	0,05	0	0,05	
NOVA	0,05	0,05	0,05	0,05	0,05	
	1	2	3	4	5	5º iteração: solucao_parcial = {[1, 2], [3, 4, 5]}
1	0	0,05	0,05	0,05	0,05	
2	0,05	0	0,05	0,05	0,05	
3	0,05	0,05	0	0,05	<b>0,05</b>	
4	0,05	0,05	0,05	0	0,05	
NOVA	0,05	0,05	0,05	0,05	0,05	

**Figura 2. Construção de uma solução por formigas**

Para um valor de modularidade 0,4 para  $S$ , o valor a ser depositado conforme cálculo da Equação 3 é de 0,505 para a evaporação de feromônio, onde as células que não participaram da solução e apenas sofreram evaporação, passaram a ter o valor 0,015. Os vértices 1 e 3 foram os primeiros vértices a ingressar em suas respectivas comunidades então foram eleitos como vértices formadores de comunidades e sofreram atualização na linha adicional da matriz, responsável pelas novas comunidades. Com a matriz de feromônio atualizada, a próxima iteração utilizou a matriz como base para a construção de uma nova solução.

	1	2	3	4	5
1	0	0,505	0,015	0,015	0,015
2	0,505	0	0,015	0,015	0,015
3	0,015	0,015	0	0,505	0,505
4	0,015	0,015	0,505	0	0,505
5	0,015	0,015	0,505	0,505	0
NOVA	0,505	0,015	0,505	0,015	0,015

**Figura 3. Atualização de feromônio**

## 5. Experimentos

Para a realização dos experimentos utilizaram-se 30 replicações para cada configuração de parâmetros de entrada, a fim de encontrar a melhor combinação para tempo e qualidade.

A configuração dos parâmetros de entrada teve sua distribuição da seguinte forma: foram utilizados para a taxa de evaporação de feromônio os valores de  $\{0.3, 0.7\}$ . Para a construção por busca gulosa, o fator guloso teve os valores de  $\{0.0, 0.3\}$ , sendo 0.0 o parâmetro responsável pela construção somente por formigas, e para o alfa necessário à construção por busca gulosa os valores utilizados foram  $\{0.1, 0.3\}$ . Utilizaram-se os valores  $\{1, 5\}$  para representar a população de formigas e como critério de parada utilizaram-se 15 iterações.

Utilizou-se instâncias consideradas como clássicas pela literatura, juntamente com as instâncias aleatórias geradas a partir do método desenvolvido por Lancichinetti, Fortunato e Radicchi (2008).

A fim de analisar com que frequência se atingiu as melhores soluções conhecidas foram utilizados testes de hipótese. Os testes foram aplicados à duas hipóteses:  $H_0$  e  $H_1$ . O valor de significância utilizado para os testes de hipótese foi de 0.01, sobre o qual utilizou-se o critério de *Bonferroni* [Dunn 1961], que é a atualização do *p-value* para seu valor, dividido pelos 240 testes. Com esta atualização o *p-value* foi de 0.00004166, que significa que os testes tentam rejeitar a hipótese nula, mantendo a alternativa em mais de 99% das amostras.

Com o resultado do teste de hipóteses, foi possível identificar o número de vezes que determinada configuração de parâmetros rejeitou a hipótese nula, tanto para qualidade de soluções como para o tempo de execução, assim permitindo a criação de dois *rankings*.

Para o tempo computacional as hipóteses foram divididas em: (i)  $Ht_0$  quando aplicados os parâmetros  $\alpha$ , a demanda de tempo é a mesma que aplicando os parâmetros  $\beta$ ; (ii)  $Ht_1$  com os parâmetros  $\alpha$  a heurística demanda menos tempo computacional que com os parâmetros  $\beta$ .

Quanto à qualidade (modularidade) as hipóteses foram divididas em: (i)  $HQ_0$  quando aplicados os parâmetros  $\alpha$ , os resultados possuem a mesma qualidade de que aplicando os parâmetros  $\beta$ ; (ii)  $HQ_1$  com os parâmetros  $\alpha$  a heurística gera soluções com mais qualidade do que com os parâmetros  $\beta$ .

O algoritmo foi desenvolvido na linguagem de programação *Python* e para a realização dos experimentos foi utilizado um computador com processador *Intel Core i7 CPU 950@ 3.06GHz*, com 6GB de memória e ambiente *Linux 64 bits*.

## 6. Resultados

A partir dos experimentos realizados, buscou-se identificar o tempo, a qualidade das soluções encontradas, a complexidade amortizada da meta-heurística OCF, e uma análise qualitativa da melhor configuração de parâmetros (qualidade).

## 6.1. Análise de Qualidade e Tempo

A Tabela 1 apresenta o *ranking* dos melhores resultados obtidos do teste de hipótese quanto à qualidade e tempo para as instâncias clássicas e aleatórias. Para a representação da posição obtida pelos parâmetros utilizaram-se as, as colunas *PQ* para qualidade e *Pt* para tempo – quanto menor sua posição, melhor sua colocação no *ranking* – e, para representar o número de vezes em que a configuração rejeitou  $H_0$  quando comparada às demais configurações, utilizaram-se as colunas *HQ* para qualidade e *Ht* para tempo. O caractere (-) indica que a hipótese foi rejeitada em todos os testes.

**Tabela 1. Teste de hipótese – Qualidade e Tempo**

<i>Alpha</i>	<i>Fator Guloso</i>	<i>População</i>	<i>Iterações</i>	<i>Feromônio</i>	<i>Complexidade</i>	<i>Pt</i>	<i>Ht</i>	<i>PQ</i>	<i>HQ</i>
0.1	0	1	15	0.3	$n^{2.61\pm 0.15}$	1	14	-	-
0.3	0	1	15	0.3	$n^{2.60\pm 0.15}$	1	14	-	-
0.1	0	1	15	0.7	$n^{2.69\pm 0.16}$	2	12	-	-
0.3	0	1	15	0.7	$n^{2.62\pm 0.15}$	2	12	-	-
0.1	0.3	1	15	0.3	$n^{2.69\pm 0.21}$	3	11	6	4
0.1	0.3	1	15	0.7	$n^{2.70\pm 0.21}$	4	10	6	4
0.3	0.3	1	15	0.3	$n^{2.68\pm 0.21}$	5	8	5	6
0.3	0.3	1	15	0.7	$n^{2.64\pm 0.21}$	5	8	5	6
0.3	0	5	15	0.3	$n^{2.65\pm 0.14}$	6	6	5	6
0.1	0	5	15	0.3	$n^{2.65\pm 0.14}$	6	6	5	6
0.3	0	5	15	0.7	$n^{2.70\pm 0.17}$	7	5	3	8
0.1	0	5	15	0.7	$n^{2.70\pm 0.16}$	8	4	4	7
0.1	0.3	5	15	0.3	$n^{2.70\pm 0.20}$	9	2	2	11
0.3	0.3	5	15	0.3	$n^{2.48\pm 0.22}$	10	2	1	14
0.3	0.3	5	15	0.7	$n^{2.45\pm 0.22}$	-	-	1	14
0.1	0.3	5	15	0.7	$n^{2.68\pm 0.20}$	-	-	2	11

Como pode ser observado, a população de formigas influencia diretamente na qualidade das soluções obtidas, pois com mais formigas a matriz acaba recebendo um reforço maior de feromônio, resultando em melhores soluções. Pode-se observar também a importância de um percentual de busca gulosa na qualidade da solução, onde uma busca construída somente por formigas pode vir a permanecer presa em um subótimo local e acabar não explorando melhores soluções no espaço de busca.

A população de formigas representa também influência no tempo, uma vez que uma formiga itera quinze vezes, cinco formigas irão iterar setenta e cinco vezes. A construção exclusivamente por formigas é outro fator influente, pois demanda um tempo menor de execução que a construção em conjunto com a busca gulosa.

A função de complexidade amortizada presente na coluna *Complexidade* demonstra um comportamento semelhante para quase todos os parâmetros. Isto indica que a diferença do tamanho das instâncias possui demanda semelhante em número de operações para todos os parâmetros.

A Tabela 1 demonstra que a população de formigas e o fator guloso são os parâmetros mais influentes da meta-heurística OCF, onde observa-se que as configurações que obtiveram um alto *ranking* são inversamente proporcionais (as configurações melhores em tempo são piores em qualidade).

Os dados apresentados na Figura 4 reforçam os argumentos já discutidos na Tabela 1, de que as melhores configurações de parâmetro obtidas para qualidade apresentam os piores tempos de execução quando comparados às demais configurações.

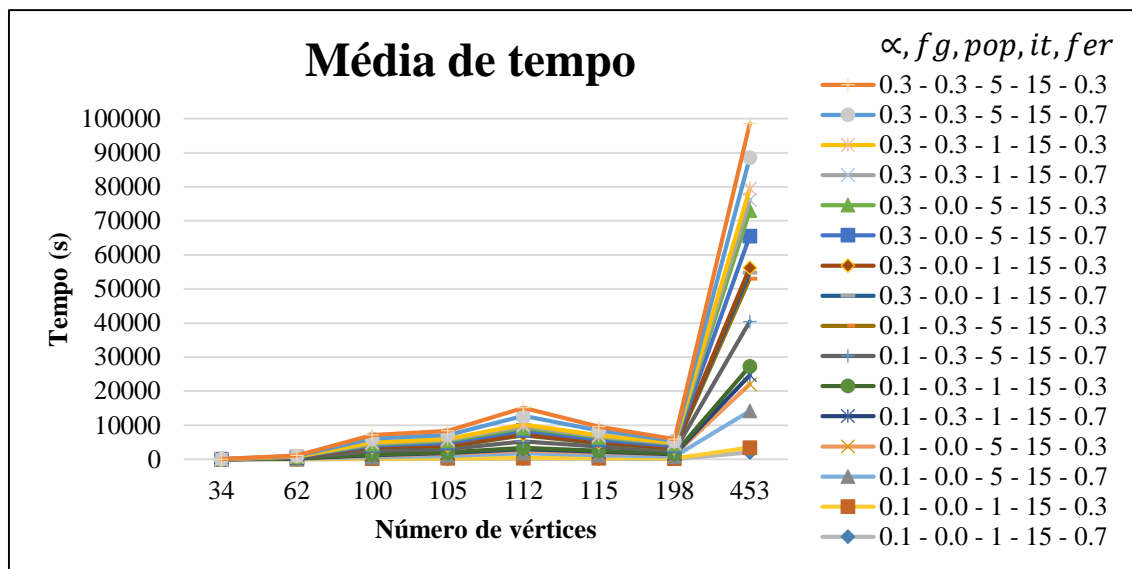


Figura 4. Média de tempo em relação a quantidade de vértices

### 6.3 Análise Qualitativa

Para realizar uma avaliação qualitativa, a meta-heurística OCF foi comparada com a heurística GRASP com religamento de caminho de Nascimento e Pitsoulis (2013) e a meta-heurística Busca em Vizinhança Variada (VNDS) de Aloise et al. (2012), sendo estes os maiores valores relatados para modularidade na literatura, conforme demonstrado na Tabela 2. O caractere (-) indica que aquela instância não foi testada pelo algoritmo em questão.

A coluna  $n$  apresenta o número de vértices,  $m$  o número de arestas,  $|C|$  o número de comunidades conhecidas. A coluna  $Q^*$  foi gerada através da solução exata de Brandes, Delling e Gaertler (2008), utilizando o problema de Programação Linear Inteira Mista no IBM-CPLEX exceto para a instância *Adjectives and nouns*, que foi obtida através da referência Aloise et al. (2012). A distância em percentual que os resultados de OCF alcançaram em relação a  $Q^*$  são relatados pela coluna Distância e, a coluna Origem demonstra qual trabalho obteve o maior valor de modularidade para a determinada instância, sendo [1] para Brandes, Delling e Gaertler (2008) e [2] para Aloise et al. (2012).

Os valores destacados demonstram a quantidade de vezes em que a meta-heurística OCF não obteve o valor máximo em relação aos melhores resultados relatados na literatura, no entanto, para as instâncias *Adjectives and nouns* e *100\_6* chegou-se a um valor bem aproximado. Os valores de distância do resultado ótimo indicam que OCF pode ser utilizado em instâncias onde não se conhece uma solução ótima, pois tende a ficar próximo ao valor ótimo com uma margem de erro de aproximadamente 7,5% para o pior caso.



**Tabela 2. Análise qualitativa**

<i>Instância</i>	<i>n</i>	<i>m</i>	<i> C </i>	<i>VNDS</i>	<i>GRASP</i>	<i>OCF</i>	<i>Q*</i>	<i>Origem</i>	<i>Distância (OCF/Q*)</i>
<i>Adjectives and nouns</i>	112	425	7	0,312	0,311	<b>0,305</b>	0,313	[2]	2,56 %
<i>Celegans metabolic</i>	453	2025	9	0,453	0,452	<b>0,419</b>	0,453	[1]	7,51 %
<i>College football</i>	115	613	10	0,605	0,605	<b>0,575</b>	0,605	[1]	4,96 %
<i>Dolphins</i>	62	159	5	0,529	0,529	0,529	0,529	[1]	0 %
<i>Jazz musicians</i>	198	2742	4	0,445	0,445	0,445	0,445	[1]	0 %
<i>Karate club</i>	34	78	4	0,420	0,420	0,420	0,420	[1]	0 %
<i>Politics books</i>	105	441	5	0,527	0,527	0,527	0,527	[1]	0 %
<i>100_1</i>	100	95	-	-	0,445	0,445	0,445	[1]	0 %
<i>100_2</i>	100	98	-	-	0,528	0,528	0,528	[1]	0 %
<i>100_3</i>	100	99	-	-	0,459	0,459	0,459	[1]	0 %
<i>100_4</i>	100	97	-	-	0,407	0,407	0,407	[1]	0 %
<i>100_5</i>	100	97	-	-	0,462	0,462	0,462	[1]	0 %
<i>100_6</i>	100	97	-	-	0,497	<b>0,494</b>	0,497	[1]	0,6 %
<i>100_7</i>	100	95	-	-	0,410	0,410	0,410	[1]	0 %
<i>100_8</i>	100	96	-	-	0,424	0,424	0,424	[1]	0 %
<i>100_9</i>	100	96	-	-	0,415	0,415	0,415	[1]	0 %
<i>100_10</i>	100	96	-	-	0,463	0,463	0,463	[1]	0 %

## 7. Conclusão

A partir dos experimentos realizados pôde-se encontrar as melhores configurações de parâmetros para tempo e qualidade. Como resultado mais positivo, percebeu-se que a meta-heurística OCF obtém as melhores soluções conhecidas para o problema da maximização da modularidade, pois encontrou na maioria das vezes resultados ótimos ou próximos a ótimos conforme literatura.

Como esta é uma meta-heurística composta por mais de uma estratégia, como buscas locais e buscas construtivas, sugere-se que os parâmetros mais indicados não estariam relacionados a tempo, e sim, a qualidade, pois OCF obtém resultados com qualidade para o contexto de maximização da modularidade. Os parâmetros  $\alpha = 0.3$ ,  $fator\ guloso = 0.3$ ,  $populacao = 5$ ,  $iteracoes = 15$ ,  $feromonio \in \{0.3, 0.7\}$  representaram as configurações que resultaram nos melhores resultados quanto à qualidade.

A meta-heurística OCF demanda tempo polinomial, no entanto insuficiente para executar instâncias com milhares de vértices. A complexidade amortizada revelou ordem próxima a cúbica de complexidade. Esta demanda de tempo está relacionada ao reforço de feromônio na matriz.

Para trabalhos futuros, é possível citar: (i) dar continuidade aos estudos de heurísticas de busca local, visando uma melhoria nas soluções encontradas pela meta-heurística OCF; (ii) dar continuidade aos estudos de heurísticas de busca gulosa, visando melhorias no momento da construção da solução; e (iii) a melhoria das estruturas de dados utilizadas, visando um acesso mais eficiente aos dados armazenados.

## Referências

- Aloise, D.; et al. (2012). Modularity maximization in networks by variable neighborhood search. *Graph Partitioning and Graph Clustering*, v. 588, p. 113-127.
- Agarwal, G. and Kempe, D. (27 nov 2008). Modularity-maximizing graph communities via mathematical programming. *The European Physical Journal B-Condensed ...*, v. 66, n. 3, p. 409–418.
- Brandes, U.; Delling, D.; Gaertler, M. (2008). On finding graph clusterings with maximum modularity. *Graph-Theoretic ...*, n. 001907, p. 1–19.
- Dorigo, M.; Stützle, T. (2004). *Ant colony optimization*. London: Massachusetts Institute of Technology.
- Dunn, O. J. (mar 1961). Multiple comparisons among means. *Journal of the American Statistical Association*, v. 56, n. 293, p. 52-64.
- Ferrara, E.; et al. (2014). Detecting criminal organizations in mobile phone networks. *Expert Systems With Applications*, v. 41, n. 13, p. 5733-5750.
- Fortunato, S.; Barthélemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*, v. 104, n. 1, p. 36-41.
- Guimerà, R.; Amaral, L. A. N. (24 fev 2005). Functional cartography of complex metabolic networks. *Nature*, v. 433, p. 895-900.
- Lancichinetti, A.; Fortunato, S.; Radicchi, F. (30 out 2008). Benchmark graphs for testing community detection algorithms. *Physical Review E*, v. 78, n. 046110.
- Meunier, D.; Fonlupt, P.; Saive, A. L.; Plailly, J.; Ravel, N.; Royet, J. P. (jul 2014). Modular structure of functional networks in olfactory memory. *NeuroImage*, v. 95, p. 264-275.
- Nascimento, M. C. V.; Pitsoulis, L. (dec 2013). Community detection by modularity maximization using GRASP with path relinking. *Computers and Operations Research*, v. 40, n. 12, p. 3121-3131.
- Nepusz, T.; Yu, H.; Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. *Nature Methods*, v. 9, p. 471-472.
- Newman, M. E. J.; Girvan, M. (fev 2004). Finding and evaluating community structure in networks. *Physical Review E*, v. 69, n. 026113.
- Newman, M. E. J. (18 jun 2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, v. 69, n. 066133.