

Extraindo Nós Influentes para Difusão de Informação em Redes Sociais

[Trabalho Prático da Disciplina - Projeto e Análise de Algoritmos]

Alcemir Rodrigues Santos
Universidade de Minas Gerais - UFMG
Instituto de Ciências Exatas - ICEx
Minas Gerais - Belo Horizonte - Campus Pampulha - CEP 31270-010
alcemir@dcc.ufmg.br

ABSTRACT

No mundo atual, com a competitividade entre as empresas que buscam mais espaço no mercado, surgem novas formas de fazer a propaganda de sua marca. Com o advento das redes sociais, uma delas tomou grande evidência, o *marketing viral* faz com que a difusão de informação aconteça de forma antes jamais imaginadas. Neste artigo trata-se o problema de maximização de influência de nós (indivíduos) de uma rede social para escolha daquele que tem o maior poder de difundir uma certa informação. As abordagens atuais utilizam métodos de grande complexidade para a aproximação do grau de influência de cada nó, se por ventura adicionado ao conjunto dos difusores da informação. Portanto, este trabalho apresenta um método mais simples para a aproximação através de um algoritmo híbrido baseado no paradigma de *força bruta*.

Categories and Subject Descriptors

F.2 [Análise de algoritmos e complexidade de problemas]: Misto

General Terms

Algoritmos, Experimentação, Desempenho

Keywords

Algoritmos de Aproximação, Marketing Viral, Grau de Influência, Redes Sociais

1. INTRODUÇÃO

Com a surgimento das redes sociais na Internet, os usuários conseguiram o espaço que queriam para debaterem sobre diversos assuntos. Inclusive os que mais os aborreciam, problemas com empresas e/ou produtos e seus atendimentos em *call centers*. Mas o espaço não estava aberto somente para a reclamação dos usuários, as empresas também aprenderam a utilizar-se dela para a sua divulgação mais efetiva.

Falar de uma determinada marca em redes sociais, como o twitter, significa a certeza de ser ouvido por muitas pessoas. Mesmo que elas não estejam em seu ciclo social. Gerando então o que ficou conhecido como “Marketing Viral”. Efeito direto da propaganda “boca-a-boca” e que recebe este nome em analogia ao processo de desenvolvimento de uma epidemia viral por uma cidade.

Obviamente, quanto mais influência um usuário tiver frente aos demais, sua mensagem poderá chegar bem mais longe dentro da rede. Este é o problema abordado aqui. Como encontrar o conjunto de usuários de maior influência e que proporcionam maior *Difusão de Informação em Redes Sociais*. Um exemplo prático deste tipo de avaliação pode ser encontrado no sítio web Twitalyzer (www.twitalyzer.com).

O problema de *seleção do nó mais influente* é NP-difícil [1] e pode ser mapeado para o problema de “*Subconjuntos independentes máximos*”.

2. CONCEITOS PRELIMINARES

Para este trabalho é considerado um grafo $G = (V, E)$ direcionado. Por direcionado entende-se que as arestas do conjunto E deste grafo são direcionadas. Ou seja, que dados dois vértices u e v do conjunto V , uma aresta (u, v) só indica o caminho do vértice u para o vértice v .

Define-se ainda o conjunto de pais de um vértice v , denotado por $\Gamma(v)$ e constituído pelas arestas que liga um vértice u qualquer, ao, por conseguinte, vértice filho v .

O grafo induzido de G é dado por $G' = (V', E')$, onde V' é um subconjunto de V e $E' = E \cap (V' \times V')$.

Um caminho para o v_n partindo do nó v_0 é escrito da seguinte forma: (v_0, \dots, v_n) , sendo que $(v_{i-1}, v_i) \in E$ com $(i = 0, \dots, n)$. Com isso podemos dizer que o vértice v_0 pode alcançar o vértice v_n , ou ainda, que o vértice v_n é alcançável pelo o vértice v_0 .

Para um vértice v do grafo G , define-se $F(v; G)$ como sendo o conjunto de todos os vértices que v pode alcançar e define-se ainda $B(v; G)$ como sendo o conjunto de todos os vértices que podem alcançar v . Para qualquer $A \subset V$, define-se

$$F(A; G) = \bigcup_{v \in A} F(v; G), B(A; G) = \bigcup_{v \in A} B(v; G).$$

Um componente fortemente conectado (SCC, do inglês *strongly connected component*) de G é o subconjunto máximo C de V tal que

$$\{\forall u, v \in C \mid \exists(u, \dots, v)\}.$$

Portanto, o $SCC(v; G)$ define o SCC de G que contém v .

3. PROBLEMA DA MAXIMIZAÇÃO DE INFLUÊNCIA

Kimura *et al.*[3] define o modelo matemático de difusão de informação através de uma rede social, representada aqui por um grafo $G = (V, E)$. Assumimos para os modelos *Independent Cascade* (Seção 3.1.1) e *Linear Threshold* (Seção 3.1.2) os seguintes pontos:

- Os nós que aceitam a informação são chamados ativos;
- O estado de um nó ou é *ativo* ou *inativo*;
- Os nós podem trocar do estado *inativo* para *ativo*, mas não podem fazer a mudança inversa;
- A difusão da informação através da rede G é representada pela difusão de nós ativos em G ;
- Dado um conjunto inicial A de nós ativos, supõe-se que os nós em A primeiro se tornam ativos e os nós restantes continuam inativos no passo $t = 0$.
- O processo de difusão de nós ativos acontece em passos discretos de tempo onde $t \geq 0$.

3.1 Modelos Fundamentais de Difusão de Informação

A representação do problema abordado em redes sociais será dada por um grafo $G = (V, E)$ orientado. Os nós que aceitam a informação serão chamados *ativos*. N discrimina o número de nós em V e L o número de arestas em E . O conjunto de nós-pai de um nó $v \in V$ é denominado por $P(v)$.

O processo de difusão ocorre em passos discretos de tempo t , onde $t > 0$, e considera-se que um nó sempre iniciará com *status inativo* e eventualmente mudará para *ativo*. Não podendo posteriormente retornar ao *status inativo*. Existe um conjunto A com os nós ativos. Ao *passo* 0 o primeiro nó sai do *status inativo* e os demais continuam.

As abordagens para o problema de *Difusão de Informação* encontradas na literatura são basicamente duas e serão utilizadas de acordo com as definições encontradas em [1]. O *Independent Cascade (IC)* e o *Linear Threshold (LT) models* serão apresentados em seguida nas subseções 3.1.1 e 3.1.2.

3.1.1 Modelo de Cascata Independente

Neste modelo, $p_{u,v}$ é uma constante que representa a probabilidade de propagação através de uma aresta (u, v) . Este número indica a probabilidade de um nó u , quando ativo, ativar cada um de seus nós filhos v em um passo t . Isto se dá em chance única e caso consiga, o nó filho ativar-se-á no passo subsequente, $t + 1$. Um nó v somente se tornará ativo no mesmo passo t quando possuir diversos nós-pai que tornaram-se ativos neste mesmo passo. Sendo esta também,

a única chance dos nós-pai u ativar o nó-filho v . É dado fim do processo quando não houver mais ativações possíveis.

Chamamos de “Grau de Influência” de um conjunto inicial A , o número $\sigma(A)$. Ele representa a quantidade de nós ativos no fim do processo.

TEOREMA 1. *O problema de maximização de influência é NP-Difícil para o modelo de Cascata Independente.*

PROOF. Considere uma instância do problema NP - Completo de *Cobertura de Conjuntos*, definido por uma coleção de sub-conjuntos S_1, S_2, \dots, S_m de um conjunto universo $U = u_1, u_2, \dots, u_n$; queremos saber se existe k dos sub-conjuntos dos quais a união é igual a U . Assumindo que $k < n < m$, mostramos que isto pode ser visto como um caso especial do problema de maximização de influência.

Dado uma instância do problema de Cobertura de Conjunto, definimos um grafo bipartido direcionado com $n + m$ nós: existe um nó i correspondente para cada conjunto S_i , um nó j correspondente a cada elemento u_j e uma aresta (i, j) com probabilidade de ativação $p_{i,j} = 1$ quando quer que $u_j \in S_i$. O problema de Cobertura de Conjuntos é equivalente a decidir se existe um conjunto A de k nós no grafo $\sigma(A) \geq n + k$. Perceba que para a instância definimos que ativação é um processo determinístico, como todas as probabilidades são 0 ou 1. Ativando inicialmente os k nós correspondentes aos conjuntos em uma solução para a Cobertura de Conjuntos resultando em ativar todos os nós n correspondentes ao universo U e se qualquer conjunto A de k nós tem $\sigma(A) \geq n + k$, então a Cobertura de Conjuntos deve ser resolvida. \square

3.1.2 Modelo de Limiar Linear

Para cada $v \in V$ neste modelo é necessário especificar um *peso* $w_{u,v} (> 0)$ para os nós-pai u de forma que

$$\sum_{u \in P(v)} w_{u,v} \leq 1 \quad (1)$$

Dado um conjunto A e um *limiar* θ_v para cada nó v é escolhido uniformemente ao acaso dentro do intervalo $[0, 1]$, o processo de difusão acontece da seguinte forma:

1. Um nó ativo u no passo t influencia o nó v de acordo com $w_{u,v}$;
2. $(P_t(v))$ irá denotar o conjunto de nós-pai de v ativos no passo t .)
3. Se o peso total dos nós-pai ativos é pelo menos o limiar θ_v , ou seja,

$$\sum_{u \in P(v)} w_{u,v} \geq \theta_v,$$

então v se tornará ativo no passo $t + 1$.

Então o processo termina se não houverem mais ativações possíveis.

Chamamos de “Grau de Influência” de um conjunto inicial A , o número $\sigma(A)$. Ele representa a quantidade de nós ativos no fim do processo.

TEOREMA 2. *O problema de maximização de influência é NP-Difícil para o modelo de Limiar Linear.*

PROOF. Considere uma instância do problema NP - Completo *Cobertura de Vértice* definido por um grafo não direcionado de n nós $G = (V, E)$ e um inteiro k ; queremos que se existe um conjunto S de k nós em G então toda aresta tem pelo menos uma terminação em S . Mostramos que isto pode ser visto como um caso especial de maximização de influência.

Dado uma instância do problema de Cobertura de Vértice envolvendo um grafo G , defini-se uma instância correspondente de maximização de influência direcionando todas as arestas de G em ambas as direções. Se existe uma cobertura de vértice S de tamanho k em G , então ela pode fazer deterministicamente $\sigma(A) = n$ ao alcançar os nós no conjunto $A = S$; reciprocamente, este é somente o método de encontrar o conjunto A com $\sigma(A) = n$. \square

3.2 Definição do Problema

O problema pode ser definido de forma simples, como segue:

DEFINIÇÃO 1. *Dado um inteiro positivo k , encontre um conjunto $A_k^* \geq \sigma(B)$ para cada conjunto de B de k nós.*

Os algoritmos encontrados a literatura pesquisada são geralmente *Algoritmos Gulosos*. Como o algoritmo de “Escalada de Montanha” que foi utilizado em [2]. Segue uma breve descrição do algoritmo.

Algorithm 1 Escalada de Montanha

```

1: procedure ESCALADADEMONTANHA( $G, k$ )
2:    $A \leftarrow \emptyset$ 
3:   for  $i \leftarrow 1, k$  do
4:     Escolha um nó  $v_i \in V$  maximizando  $\sigma(A \cup \{v\})$ 
        $\triangleright (v \in V \setminus A)$ 
5:     Faça  $A \leftarrow A \cup \{v_i\}$ 
6:   end for
7:   return  $A$   $\triangleright$  O conjunto de nós ativos que
       maximizam o grau de influência
8: end procedure

```

3.2.1 Método Convencional para Estimativa do Grau de Influência

Para que a linha 4 do Algoritmo 1 seja codificada se faz necessário de um método para a estimativa do *grau de influência* de cada vértice v que será inserido no conjunto de nós ativos A . Este método foi definido por Kempe *et al.*[1] e pode ser visto no Algoritmo .

Algorithm 2 Método Convencional para Estimativa do Grau de Influência

```

1: procedure METODOCONVENCIONAL
2:   for  $m \leftarrow 1, M$  do
3:     Calcule  $\varphi(A \cup \{v\})$ .
4:     Faça  $x_m \leftarrow \varphi(A \cup \{v\})$ .
5:   end for
6:   Faça  $\sigma(A \cup \{v\}) \leftarrow (1/M) \sum_{m=1}^M x_m$ .
7: end procedure

```

Onde, cada $\varphi(A \cup \{v\})$ é computado assim:

Algorithm 3

```

1: procedure
2:   Faça  $H_0 \leftarrow A \cup \{v\}$ .
3:   Faça  $t \leftarrow 0$ .
4:   while  $H_t \neq \emptyset$  do
5:     Faça  $H_{t+1} \leftarrow \{\text{nós ativados no passo } t + 1\}$ .
6:     Faça  $t \leftarrow t + 1$ .
7:   end while
8:   Faça  $\varphi(A \cup \{v\}) \leftarrow \sum_{j=0}^{t-1} |H_j|$ 
9: end procedure

```

4. ESTADO-DA-ARTE

No trabalho de Kimura *et al.* tomado como referência foi apresentado um método alternativo ao mostrado na seção 3.2.1. Utilizaram para a elaboração do método a *bond percolation*, a teoria dos grafos. A partir da análise da complexidade computacional do método e a posterior comparação mostrou-se mais eficiente que o Algoritmo 2. Segue o algoritmo proposto por Kimura *et al.* [2] e posteriormente melhor fundamentado [3].

Algorithm 4 Método Proposto por Kimura *et al.*

```

1: procedure
2:   Encontre o subconjunto  $F(A; G_r)$  de  $V$ .
3:   Faça  $|F(A \cup \{v\}; G_r)| \leftarrow |F(A; G_r)|$  para todo  $v \in F(A; G_r) \setminus A$ .
4:   Encontre o subconjunto  $V_r^A = V \setminus F(A; G_r)$  of  $V$ , e o grafo induzido  $G_r^A$  de  $G_r$  a  $V_r^A$ .
5:   Faça  $U \leftarrow \emptyset$ .
6:   while  $V_r^A \setminus U \neq \emptyset$  do
7:     Pegue o vértice  $u \in V_r^A \setminus U$ .
8:     Encontre o subconjunto  $F(u; G_r^A)$  de  $V_r^A$ .
9:     Encontre o subconjunto  $C(u; G_r^A) = B(u; G_r^A) \cap F(u; G_r^A)$  de  $F(u; G_r^A)$ .
10:    Faça  $|F(A \cup v; G_r)| \leftarrow |F(u; G_r^A)| + |F(A; G_r^A)|$ 
       para todos  $v \in C(u; G_r^A)$ .
11:    Faça  $U \leftarrow U \cup C(u; G_r^A)$ .
12:   end while
13: end procedure

```

4.1 Dificuldades encontradas na implementação

Ao término da leitura do trabalho de referência [2], chegou-se a conclusão de inviabilidade de implementação do método apresentado por eles. O algoritmo que é apresentado apenas significa uma parte para a solução do problema, não sendo suficiente para o completo entendimento do mesmo apenas com a da descrição do algoritmo descrito no trabalho.

Após pesquisa sobre o problema, não foram encontradas boas referências, o trabalho de Kempe[1] foi o que mais se aproximou do trabalho de referência. Até encontrar um trabalho do mesmo grupo de Kimura que extendia o trabalho inicial publicado somente 3 anos após o primeiro [3].

Com muito esforço e ajuda do segundo trabalho de Kimura *et al.* começou-se a entender o problema mais claramente e a trabalhar em sua implementação.

5. MÉTODO PROPOSTO

O método proposto por este trabalho é baseado no paradigma de projeto de algoritmos denominado *força bruta* e foi elaborado baseado no método convencional. Para a aproximação do grau de influência exercido por cada indivíduo (denominado *difusor da informação*) na rede social é feita a contabilização de quantas instâncias de indivíduos conectados à esta rede (denominados *nós-alvo*) o difusor consegue alcançar.

Para isto é feito o caminhamento por toda a árvore geradora do grafo que representa as conexões da rede social pelas quais houve de fato a propagação da informação, partindo do indivíduo difusor da informação até chegar-se ao fim das possibilidades de caminhamento. O caminhamento pela árvore é feito por meio de uma busca em largura na lista de adjacência de cada nó avaliado num dado instante.

Algorithm 5 Método Proposto para Estimativa do Grau de Influência

```
1: procedure METODOPROPOSTO
2:   for  $m \leftarrow 1, M$  do
3:     Calcule  $|F(A \cup \{v\}); G_r|, \forall v \in V \setminus A$ .
4:     Calcule  $|F(A); G_r|, \forall v \in A$ 
5:     Faça  $x_{v,m} \leftarrow |F(A \cup \{v\}); G_r|, \forall v \in V \setminus A$ .
6:     Faça  $x_{v,m} \leftarrow |F(A); G_r|, \forall v \in A$ .
7:   end for
8:   Faça  $\sigma(A \cup \{v\}) \leftarrow (1/M) \sum_{m=1}^M x_m$ .
9: end procedure
```

Os nós ativos da rede, ou seja, os nós que iram iniciar a difusão da informação, recebem um tratamento especial. Isto pode ser percebido no algoritmo 5. Isto é proposto pois eles fazem parte do conjunto que maximiza o alcance dentro do grafo e para eles é atribuído o maior valor encontrado para todos os elementos do grupo.

Assim fica descrito o algoritmo proposto para a estimativa do grau de influência de um nó dentro das arestas “ocupadas” do grafo.

5.1 Aspectos técnicos

Ambas as implementações foram feitas com a linguagem JavaSE versão 1.6 e executadas sobre a máquina virtual Java fornecida pela *Oracle*¹. Foi parcialmente testada com o auxílio de ferramentas como JUnit² versão 4 e do hamcrest³, que ajudou nas assertões. Utilizou-se a IDE Eclipse versão 3.6 (Helios) release 2 e a técnica de *Test Driven Development* com para condução do desenvolvimento.

Mais detalhes sobre a configuração do ambiente utilizado na seção de Experimentos (6).

6. EXPERIMENTOS

Os experimentos foram todos conduzidos no ambiente restrito de um computador portátil pessoal de configuração razoável, como segue:

- CPU Intel Pentium Dual Core T2130 de 1.86Hz;
- Memória RAM de aproximadamente 2GB;
- Disco Rígido de aproximadamente 120GB.

O computador opera com sistema operacional de distribuição livre, Ubuntu⁴ versão 10.10, com kernel 2.6.35.28-generic e gerenciador de janelas GNOME 2.32.0.

Os dados referentes ao tempo de execução consideram apenas a execução dos algoritmos para estimar o grau de influência de um nó do grafo. E foram auferidas com o método simples (*System.currentTimeMillis()*) da API da linguagem Java ao início, ao final da execução e em seguida feita a subtração dos dois valores. Gráficos do uso de CPU e de memória Heap foram gerados pela ferramenta de monitoramento de aplicações Java chamada VisualVM⁵

Em cada uma das sub-seções seguintes, são apresentados os dados coletados após a execução dos algoritmos na máquina descrita acima. No momento da execução dos algoritmos, nenhuma outra aplicação era executada em primeiro plano. Isto foi feito para proporcionar que aos algoritmos uma rápida resposta da máquina utilizada quando da requisição de mais recursos.

Em seguida são exibidos os dados coletados com os experimentos.

6.1 Resultados

Nesta seção são apresentados os dados do experimento realizado com os dois algoritmos tratados neste trabalho, o algoritmo de Kempe *et al.* e o método proposto por este trabalho.

Os experimentos foram conduzidos inicialmente utilizando o número de iterações M , definida no algoritmo 2, igual a 1000. O experimento consistiu em executar os algoritmos apresentados por M iterações para ao final calcular a média dos números de nós do conjunto final de nós ativados pelo processo de difusão. Este processo foi reexecutado para pequenos e médios conjuntos de nós ativados de tamanhos iguais a 25, 50, 100, 200, e 500.

O conjunto do nós ativos bem como as arestas do grafo, as quais representavam a difusão da informação, foram gerados aleatoriamente e adicionados a um arquivo texto. O arquivo é lido, monta-se o grafo e em seguida começa-se o processo de estimativa.

6.1.1 Sobre o Tempo de execução

No Gráfico 1 é mostrado o relacionamento do número de nós ativados pelo processo de difusão em relação ao tempo de execução em segundos. Note que para fins de melhor apresentação foi aplicado a função logaritmo utilizando a base 10 ao valor contabilizado para o tempo de execução como aponta o gráfico. Os dados utilizados para a construção do Gráfico 1 são apresentados na Tabela 6.1.1.

¹<http://www.oracle.com>

²<http://www.junit.org>

³<http://code.google.com/p/hamcrest>

⁴<http://www.ubuntu.com>

⁵<http://visualvm.java.net/>

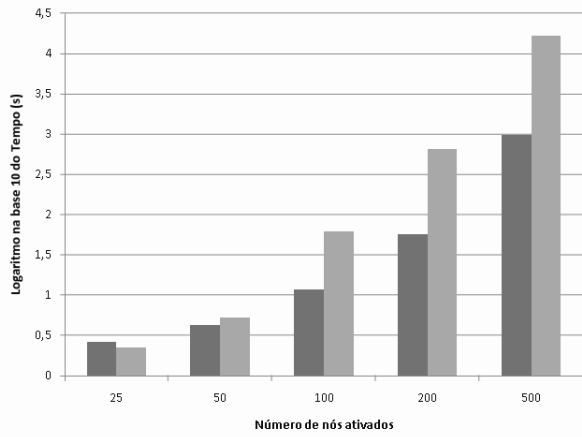


Figure 1: Relacionamento do Número de nós ativados versus \log_{10} (Tempo de execução em segundos).

Inicialmente o método teve um desempenho melhor que o método de Kimura *et al.*. Contudo, percebeu-se que o método proposto não teve um bom desempenho com grafos de tamanhos maiores.

no. de nós	Proposto (s)	Literatura (s)
25	2,214	2,654
50	5,216	4,262
100	62,051	11,815
200	649,582	57,099
500	16442,073	962,940

Table 1: Tempo de execução dos dois algoritmos dado em milissegundos.

O Gráfico 1 mostra claramente o comportamento não polinomial do problema. Com o aumento da quantidade de nós a serem analisados, aumenta de forma semelhante à uma curva exponencial.

6.1.2 Consumo geral de CPU

As Figuras 2 e 3 mostram o uso da CPU e da Heap durante a execução dos algoritmos com o grafo de 200 nós. As grandes oscilações percebidas nos gráficos em instantes próximos ao metade da execução correspondem exatamente ao instante em que encerra-se a execução do algoritmo de referência e passar-se-ia então à execução do algoritmo proposto.

7. ANÁLISE

Nesta seção serão apresentados alguns gráficos e dados produzidos para fins de uma análise estatística.

Os gráficos 4, 5 e 6 mostram variação do tempo em 50 iterações de execução com 25, 50 e 100 nós, respectivamente. A curva de cor mais escura representa o tempo de execução do método proposto e a mais clara o tempo de execução do método de Kimura *et al.*. É perceptível que a medida que aumenta a quantidade de nós a distância entre os tempos de execução também aumenta.

Para uma análise mais apurada, os algoritmos foram sub-

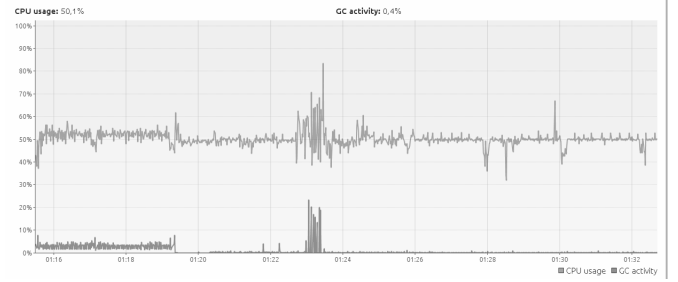


Figure 2: Uso da CPU durante o processamento do grafo de 200 nós.

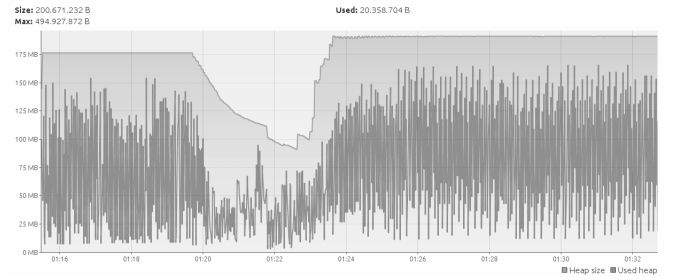


Figure 3: Uso da Heap durante o processamento do grafo de 200 nós.

metidos a execuções repetidas com quantidades de nós diferentes. Coletou-se os dados e em seguida faremos análise dos dados baseada nos resultados da aplicação do Teste T-pareado. Foram executadas 50 repetições de cada execuções e auferidos os tempos de cada uma delas.

no. de nós	T.M.(s)	D.P.(s)	I.C.(95%)
25	181,28	141,04	(167,83;194,73)
50	3080,74	196,27	(3062,02;3099,46)
100	41418,8	1413,93	(41283,93;41553,67)

Table 2: Resultado da aplicação do Teste T pareado.

A Tabela 7 possui os dados da aplicação do Teste T-pareado. A tabela é composta por 4 colunas, cada uma delas indica o número de nós existentes no grafo, a média das diferenças entre os tempos de execução dos algoritmos medido em segundos, o desvio padrão também medido em segundos e o intervalo de confiança com 95% de confiança, respectivamente. Este teste evidenciou a melhor eficiência do algoritmo tomado como referência.

8. CONCLUSÃO

A análise estatística feita para a comparação dos algoritmos deste experimento, mostrou que o método proposto falhou em melhorar o desempenho algoritmo utilizado para comparação sendo encontrado na literatura. Após análise mais íntima entre os dois métodos, o método proposto perde muito e torna-se muito ingênuo frente ao método proposto por Kimura *et al.* ao ignorar a ideia de componentes fortemente conectados.

Frente aos resultados expostos neste estudo é possível concluir que o trabalho de Kimura *et al.* tem desempenho supe-

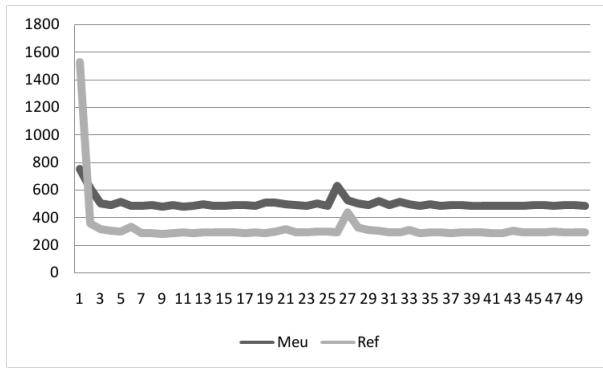


Figure 4: Variação do tempo em 50 iterações de execução com 25 nós.

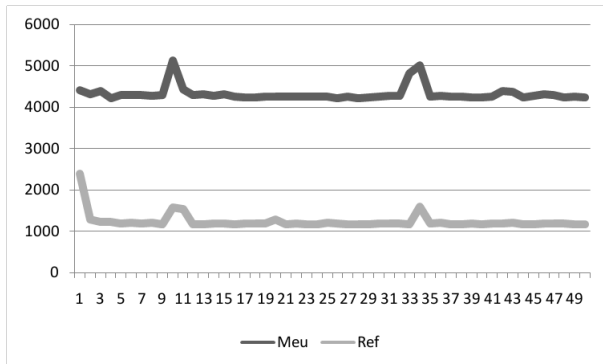


Figure 5: Variação do tempo em 50 iterações de execução com 50 nós.

rior ao método proposto. O fato do método de Kimura *et al.* utiliza-se do conceito de *componentes fortemente conectados* implica na redução do custo computacional do seu algoritmo para ao tempo da estimativa do grau de influência de cada nó que não pertence ao conjunto dos nós ativos.

9. REFERENCES

- [1] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '03*, pages 137–146, New York, NY, USA, 2003. ACM.
- [2] M. Kimura, K. Saito, and R. Nakano. Extracting influential nodes for information diffusion on a social network. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, pages 1371–1376. AAAI Press, 2007.
- [3] M. Kimura, K. Saito, R. Nakano, and H. Motoda. Extracting influential nodes on a social network for information diffusion. *Data Mining Knowledge Discovery*, 20:70–97, January 2010.



Figure 6: Variação do tempo em 50 iterações de execução com 100 nós.