Laboratório Nacional de Computação Científica

Relatório de Atividades PIBIC

Processo CNPq 153399/2015-5

Dados Gerais

Este documento é uma descrição das pesquisas feitas pelo bolsista **Oscar Neiva Eulálio Neto** (CPF 140.409.757-07) para o Programa de Bolsas de Iniciação Científica (PIBIC) do Laboratório Nacional de Computação Científica (LNCC). O trabalho possui como título: **Controle e Simulação de Sistemas Sujeitos a Saltos Markovianos** e os estudos foram supervisionados pelo professor Marcos Garcia Todorov, pesquisador da instituição.

Introdução

A crescente quantidade de informação na *Web* nos últimos anos, fez das ferramentas de busca algo indispensável na coleta de informação. O algoritmo *PageRank* é a técnica chave do Google, proposta inicialmente em [2] no ano de 1998. O Google é um dos mais bem sucedidos buscadores e o *PageRank* é com certeza um dos fatores deste sucesso.

Para o cálculo do *PageRank* algumas considerações devem ser feitas e um fator crítico a ser levado em conta é o tamanho da estrutura da *Web*. A *Web* até 2010 era composta por pouco mais de 8 bilhões de páginas [5], lembrando ainda que esse número está em constante crescimento. Atualmente a estimação do *Ranking* das páginas é feita de forma centralizada no Google, onde toda a coleta de dados da *Web* é feita por rastreadores, ou *Crawlers*, que a acessam de forma autônoma.

O algoritmo do *PageRank* está associado a conceitos da área de sistemas e controle. Seu cálculo dá-se por uma equação de diferença [3], cuja a matriz de transição é do tipo estocástica, devido a possibilidade do conjunto de páginas da *Internet* poderem ser modeladas como uma cadeia de Markov de estados discreto [4].

O trabalho tratou-se de uma revisão dos modelos propostos para o algoritmo do *PageRank* em [6]. Assim, inicialmente foi necessário um estudo dos conceitos por trás dos modelos matemáticos do algoritmo, para que em seguida pudessem ser realizadas as simulações e serem feitas conclusões a partir delas. Foram realizadas simulações desde os modelos mais simples a aqueles que estão associados a situações mais mal comportadas. Por fim, foram analisados os resultados e comparados os modelos e recursos usados nas simulações.

O Algoritmo *PageRank*

O *PageRank* é o principal algoritmo por trás das engrenagens de busca do *Google*, criado em 1998 por Lawrence Page e Sergey Brin (figura 1) para atribuir pesos a páginas da *Web*. Ele tem

como função atribuir um valor numérico para cada elemento em um conjunto de documentos, em que na *Internet* trataríam-se de páginas *Web*. Tendo como principal objetivo a identificação das páginas mais importantes aos usuários da rede, de forma a atribuir valores maiores as páginas mais importantes.

Figure 1: Lawrence Page e Sergey Brin criadores do *PageRank* e fundadores do Google.

A respeito da atribuição de importância às páginas, a página mais importante será a mais visitada e a página mais visitada será aquela que está associada a um maior número de *links*. De forma mais precisa o número de visitas de uma página vai estar associado aos *links* de saída que ela possui. Contudo, uma página também pode ser considerada importante por estar próxima a uma página de alto *PageRank*.

O Problema do *PageRank*

O *PageRank* pode ser pensado como um modelo que simula o comportamento de um usuário que age de forma aleatória. Considerar que existe um navegador que vai acessando páginas aleatoriamente, de forma a clicar somente em *links* na página em que se encontra, consiste no modelo mais simples de cálculo do *PageRank*.

Outra importante situação que precisa ser levada em consideração é a possibilidade de um usuário acessar uma página sem links de saída, como por exemplo, um link que o direcione para um documento PDF, como ilustrado na figura 2. Neste caso o usuário poderia usar o botão de voltar do *Browser* para sair deste tipo de buraco negro na rede, o que simplifica possíveis problemas no cálculo do *PageRank*.

Um outro problema surge ao considerar-se que o usuário pode saltar de uma página para outra sem seguir a estrutura de *links*, por ficar entediado, ou por um outro motivo qualquer. Um exemplo de como isso poderia ocorrer, é o caso do usuário estar na página do Laboratório Nacional

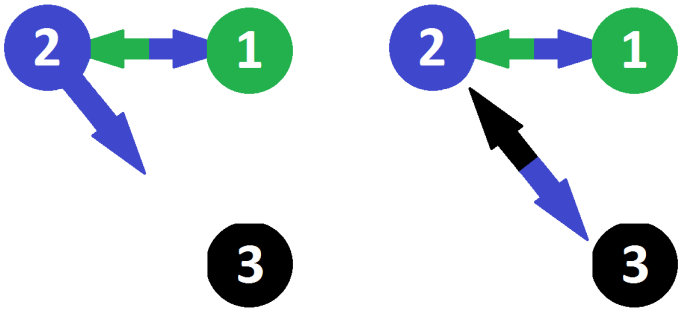

Figure 2: O buraco negro da *Web*.

de Computação Científica - LNCC e seguindo a estrutura de *Hyperlink*, clicar em um *link* que o leve à página do governo federal. No entanto, o usuário de repente altera o *Uniform Resource Locator - URL* e vai para um *site* de notícias que não está diretamente ligado aos outros dois *sites*. Este exemplo é ilustrado na figura 3.

Figure 3: Navegação entre páginas indiretamente conectadas.

A massividade da *Web* é uma das principais questões por trás do cálculo do *PageRank*, considerando que a dimensão da *Web* pode tornar o ranqueamento de páginas impossível. No entanto, a solução para este problema é o uso de algoritmos distribuídos, sobre os quais a cada ano novas publicações são feitas [7]. Neste contexto, a fim de melhorar a performance do processamento, o cálculo a partir do modelo distribuído é realizado com a ajuda de vários conputadores num *cluster*, como o da figura 4.

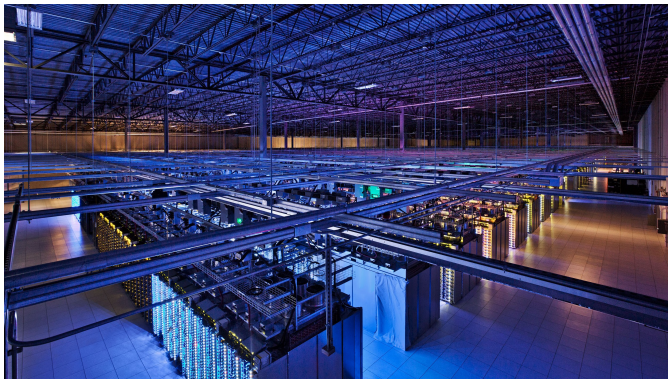

Figure 4: Cluster do Google.

Definição dos Modelos

A Matriz *Hyperlink*

A construção da matriz *hyperlink* é feita com base na estrutura da *Web*, ou ainda do grafo que a representa. Assim, considerando que o grafo da figura 5 representa um conjunto de páginas interligadas, seus elementos são definidos da seguinte forma:

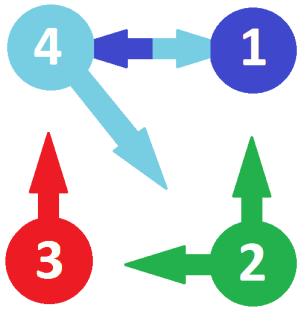

Figure 5: Grafo representando *links* entre páginas *Web*.

- n: como os nós, que representam as páginas,
- $\mathcal{E}$: como as arestas, que representam os *links* entre as páginas.

Para representar o *link* entre duas páginas específicas usa-se i e j, de forma que o i esteja associado à página de saída e j com a de entrada. Considerando então o caso de um vértice i estar conectado a um j, tem-se $(i, j) \in \mathcal{E}$.

A matriz de transição que é construída a partir do grafo tem a seguinte propriedade para cada um de seus elementos:

$$a_{ij} = \begin{cases} \dfrac{1}{n_i}, & \text{caso } (i,j) \in \mathcal{E}, \\ 0, & \text{caso contrário.} \end{cases} \tag{1}$$

Cada elemento a_{ij} é formado a partir das relações entre os nós do grafo. Para o nó 1 por exemplo, constata-se que ele só possui relações com o nó 4, por tanto, para um navegador que encontra-se em 1 tem-se 100%, de certeza de que seu próximo passo será para 4. Já no caso do nó 2 pode-se observar que o navegador pode ir para 1 ou 3, ou seja, ele tem 50% de chance de ir para cada um desses dois nós.

$$A = \begin{pmatrix} 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \tag{2}$$

E assim constroi-se a matriz de transição (2) de forma que todas as suas colunas respeitem a propriedade:

$$\sum_{i=1}^{n} x_i = 1. \tag{3}$$

O *Power Method*

Este é o modelo mais simples para a estimação do *PageRank*. Esta abordagem constitui-se de uma equação de diferença, a qual possui uma matriz de transição coluna estocástica e um vetor estocástico. O vetor estocástico é formado pelos valores das variáveis aleatórias da cadeia de Markov, ou seja os valores das páginas. O valor de *PageRank* de um conjunto de n páginas da *Web* é definido como o vetor $x \in \mathbb{R}^{n\times 1}$ que satisfaz as seguintes equações:

$$x = Ax, \quad x \geq 0, \quad \sum_{i=1}^{n} x_i = 1, \tag{4}$$

onde $A \in \mathbb{R}^{n\times n}$ é a transposta da matriz de transição da cadeia de Markov.

Assim, o método usual para obtenção do *PageRank* é o chamado *Power Method* [6], que alcança o *PageRank* através da seguinte iteração:

$$x(k+1) = Ax(k), \quad k \geq 0, \quad \text{com} \quad x(0) = x_0, \tag{5}$$

onde $x_0 \in \mathbb{R}^{n\times 1}$ é uma condição inicial positiva de soma igual a um.

Questões de Convergência do *Power Method*

Após um número considerável de iterações e em um número considerável de casos, o *Power Method* atinge a distribuição limite com os valores de *PageRank* de cada página. Além disso, independente da condição inicial, ao final das iterações os valores de *PageRank* tendem a ser os mesmos.

Isso ocorre devido a matriz de transição ser irredutível, o que reflete no navegador aleatório sempre conseguir chegar a uma certa página partindo de uma outra qualquer, mesmo que essas duas páginas não estejam diretamente conectadas. Observando o grafo da figura 5, pode-se observar que não existe um *link* direto entre 1 e 3, para chegar-se em 3 a partir de 1 o usuário teria de ir de 1 para 4, de 4 para 2 e de 2 para 3.

O *Teleportation Model*

O *Teleportation Model* é uma estratégia reconhecida para que, através de uma pequena modificação na matriz A, o método convirja globalmente para o PageRank. Essa modificação na matriz A é representada como uma combinação convexa de duas matrizes, em que a matriz de transição agora passa a ser M, dada por:

$$M = (1 - m)A + \frac{m}{n}\mathbf{1}\mathbf{1}^T, \tag{6}$$

onde $M \in \mathbb{R}^{n \times n}$, $m \in (0, 1)$ e $\mathbf{1} \in \mathbb{R}^{N \times 1}$. Em que $\mathbf{1}\mathbf{1}^T$ é uma matriz preenchida de 1's, resultante do produto entre um vetor de 1's $\mathbf{1}$ e um vetor de 1's transposto $\mathbf{1}^T$. Vale lembrar também que n é o número de nós do grafo, ou número de páginas, e que A continua sendo a mesma matriz de transição, apenas é substituida por M na equação apresentada no *Power Method*.

O Modelo dos Algoritmos Distribuídos

A fim de tornar o cálculo do *PageRank* menos custoso, e melhor explorar os recursos computacionais dos servidores disponíveis na web, uma alternativa é o emprego de algoritmos distribuídos. A agregação de tais experimentos aleatórios faz então com que o *power method* se torne um sistema linear com saltos Markovianos do seguinte tipo:

$$x(k + 1) = A_{\theta(k)}x(k), \quad k \geq 0, \quad \text{com} \quad x(0) = x_0. \tag{7}$$

Em que a matriz de transição A agora passa a ser várias matrizes *links* distribuídas $A_i \in \mathbb{R}^{n \times n}$, $i = 1, 2, ..., n$. Sendo cada uma das matrizes distribuídas definidas da seguinte forma:

- a i-ésima coluna de A_i coincide com a i-ésima coluna de A,
- a j-ésima entrada diagonal de A_i é igual a um,
- para $j = 1, ..., n$, $j \neq i$, todas as outras entradas a_{ij} são iguais a zero.

A ideia agora é de que cada página fará o seu cálculo de *PageRank*, por isso são criadas as matrizes distribuídas. Em que as matrizes são construídas da seguinte forma.

$$A_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad A_2 = \begin{pmatrix} 1 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{8}$$

$$A_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad A_4 = \begin{pmatrix} 1 & 0 & 0 & 1/2 \\ 0 & 1 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{9}$$

Acoplando agora o modelo das matrizes distribuídas ao modelo do *Teleportation* tem-se (10),

$$x(k+1) = (1 - \hat{m})A_{\theta(k)}x(k) + \frac{\hat{m}}{N}\mathbf{1}\mathbf{1}^T, \quad k \geq 0, \quad \text{com} \quad x(0) = x_0, \tag{10}$$

onde $\hat{m}$ é definido como:

$$\hat{m} = \frac{2m}{n - m(n-2)}, \tag{11}$$

e $m = 0.15$, é um parâmetro que foi obtido através de testes em [8].

Questões de Convergência do Modelo Distribuído

De forma similar ao que ocorre no *Power Method*, este método apresenta problemas de convergência, pois as matrizes A_1, A_2, A_3 e A_4 não são irredutíveis. Isso ocorre devido a inserção de 1's nas diagonais, o que na prática implicaria no nó 4 só ter saída e chegada nele mesmo. Dessa forma, usando-se da matriz A_1, por exemplo, um navegador aleatório que partisse de 4, nunca sairia de lá.

Para isso é usada uma média no tempo, em que a cada iteração é feita uma média de todos os estados anteriores. Assim $y(k)$ é a média do conjunto de amostras $x(0), ..., x(k)$ definida como (12),

$$y(k) = \frac{1}{k+1} \sum_{l=0}^{k} x(l). \tag{12}$$

De tal forma que o algoritmo converge no sentido da média quadrada, ou seja,

$$\lim_{k \to \infty} \mathbb{E}[\| \, y(k) - x^* \, \|^2] = 0. \tag{13}$$

Neste contexto, quanto maior for o número de iterações k, mais o valor de $y(k)$ se aproxima da distribuição limite, o que representa o valor de *PageRank*.

A fim de minimizar custos computacionais, um formato recursivo da média $y(k)$ foi usado nas simulações, que possui o seguinte formato:

$$y(k+1) = \frac{(k+1)}{(k+2)} y(k) + \frac{1}{(k+2)} x(k+1). \tag{14}$$

Simulações

As simulações foram feitas a partir dos modelos descritos anteriormente e a partir do grafo da figura 6 com o objetivo de observar algumas considerações feitas na literatura.

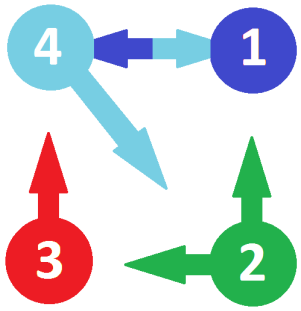

Figure 6: Grafo usado nas simulações.

Simulação do *Power Method*

Na simulação do modelo mais simples do cálculo do *PageRank*, o *Power Method*, observa-se uma rápida convergência para cada um dos valores do vetor de estados. Isso ocorre devido a dimensão do problema usado na simulação. As cores das linhas dos gráficos, que representam os resultados das simulações, seguem o mesmo esquema das cores do grafo na figura 6. Desta forma, as linhas do gráfico estão associadas às páginas da seguinte forma:

- página 1: azul escuro,

- página 2: verde,
- página 3: vermelho,
- página 4: azul claro.

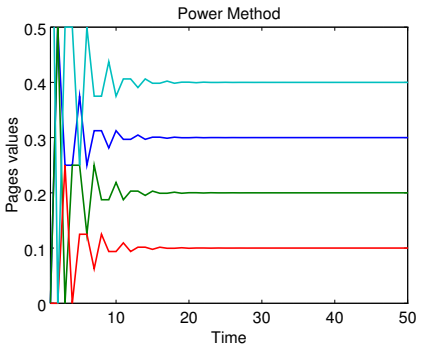

Figure 7: Resultado das simulações do *Power Method*.

Assim pode-se observar que a página de número 4 foi a que recebeu um maior *PageRank*, dentro deste conjunto de páginas. Ao mesmo que 4 é a página que está associada a um maior número de links, tanto de entrada quanto de saída, o que define seu alto valor de *PageRank*.

Entretanto, um resultado em primeira vista parece não ser conveniente ao grafo. Ao analisar-se o valor final de 1 observa-se que ele está na frente da página 2, e 2 possui mais *links* de saída que 1. Fazendo-se uma análise mais cuidadosa nota-se que 1 possui *links* de entrada e saída para 4, ou seja por estar ligada a 4 e por 4 possuir alto *PageRank*, o número de acessos a 1 foi maior por estar próximo a uma página supostamente importante.

Simulação do *Teleportation Model*

Na simulação do *Teleportation Model*, figura 8, também observa-se uma rápida convergência como nos resultados da simulação do *Power Method*.

Além disso uma grande semelhança é observada entre as duas simulações, como pode-se constatar na figura 9. Tanto a semelhança quanto a rápida convergência se devem ao fato do exemplo simulado não possuir um enorme número de estados n e por ser bem comportado.

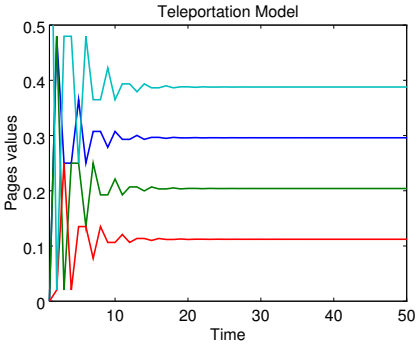

Figure 8: Resultado das simulações do *Teleportation Model*.

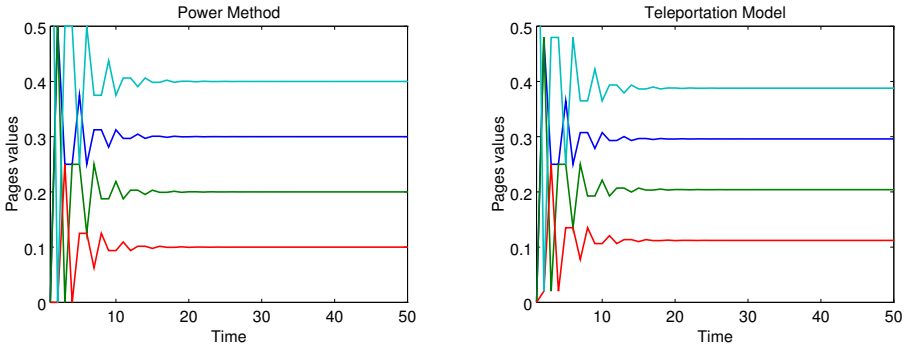

Figure 9: Comparação entre o *Power Method* e o *Teleportation Model*.

Simulações do Modelo Distribuído

No modelo distribuído sem a média no tempo observa-se uma oscilação dos estados sem que os valores das variáveis aleatórias atinjam a distribuição limite, ou seja, neste modelo sem a média no tempo não foi possível encontrar um valor de *PageRank* mesmo depois de 500 iterações. Os resultados dessa simulação estão apresentados na figura 10.

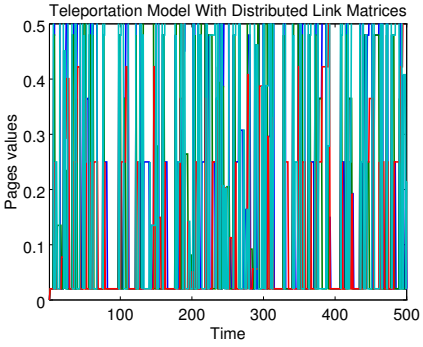

Figure 10: Resultado das simulações do modelo distribuído.

A não convergência ocorre devido as matrizes distribuídas possuirem 1's em suas diagonais, o que implicaria, por exemplo, usando-se a matriz A_1 como referência, nos nós 2, 3 e 4 só terem

saídas para si mesmo, seguindo o esquema apresentado na figura 11.

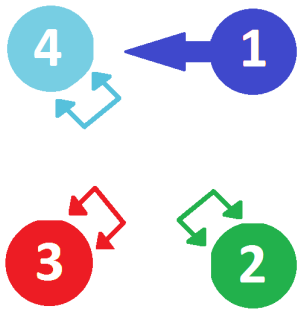

Figure 11: Abstração dos *links* das páginas 2, 3 e 4 a partir da matriz A_1.

Entretanto, feita a média dos valores do modelo distribuído os resultados convergem, conforme é mostrado na figura 12. Mas ainda assim é possível observar algumas oscilações no gráfico, esta simulação também foi feita ao longo de 500 iterações.

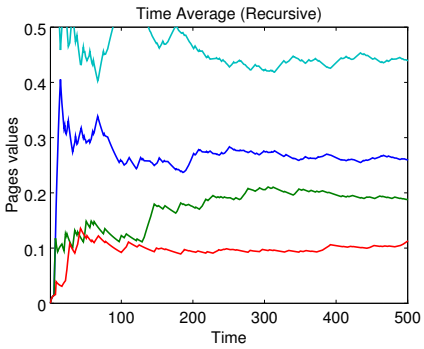

Figure 12: Resultado das simulações do modelo distribuído com a média no tempo.

Além disso, observa-se que na distribuição limite são atingidos valores bem próximos aos anteriormente encontrados para cada um dos estados no modelo do *Power Method*, o que pode ser constatado a partir da figura 13.

Método de *Monte Carlo* Aplicado após Modelo Recursivo da Média

Por fim, foi feita uma simulação usando método de Monte Carlo [1], no qual para cada uma das 500 iterações que podem ser vistas no gráfico foram feitas mais 500 iterações antes dos resultados serem plotados de forma a fazer uma média com os valores obtidos depois da média no tempo. Os resultados desta simulação encontram-se na figura 14.

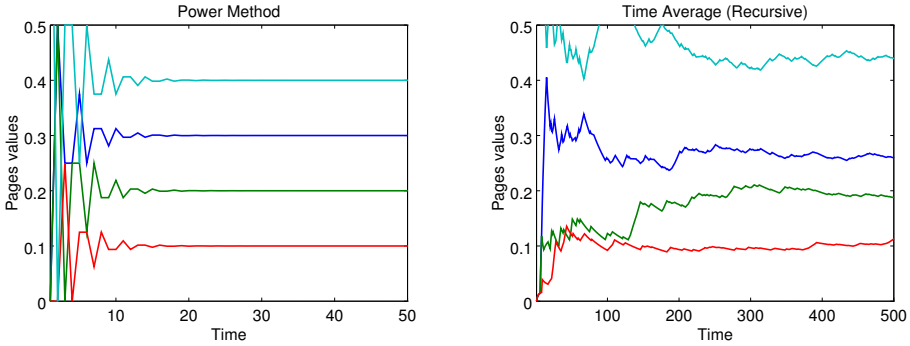

Figure 13: Comparação entre o *Power Method* e a média no tempo do modelo distribuído.

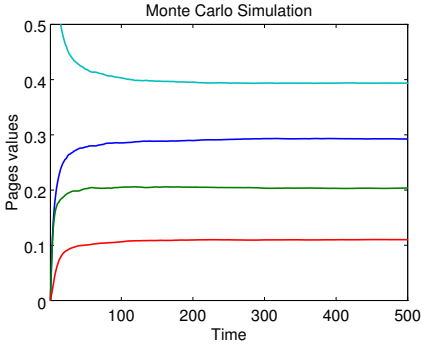

Figure 14: Resultado das simulações com o método de Monte Carlo.

Comparando a simulação em que usou-se do método de Monte Carlo com a simulação da média no tempo, observa-se que o gráfico do método de Monte Carlo possui uma curva mais suavizada, o que pode ser constatado nos resultados apresentados na figura 14. Além disso a evolução dos valores está mais bem definida, principalmente quando comparados ao valor final da distribuição limite.

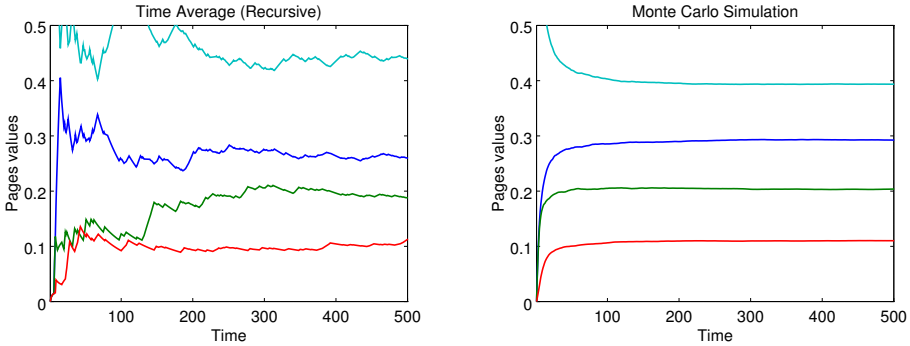

Figure 15: Comparação entre a média no tempo do modelo distribuído e o método de Monte Carlo.

E ainda comparando os resultados da simulação com Monte Carlo e da simulação do *Power*

Method 16, nota-se que os valores finais estão de acordo com os alcançados no modelo do *Power Method*, o que sugere que o modelo é válido.

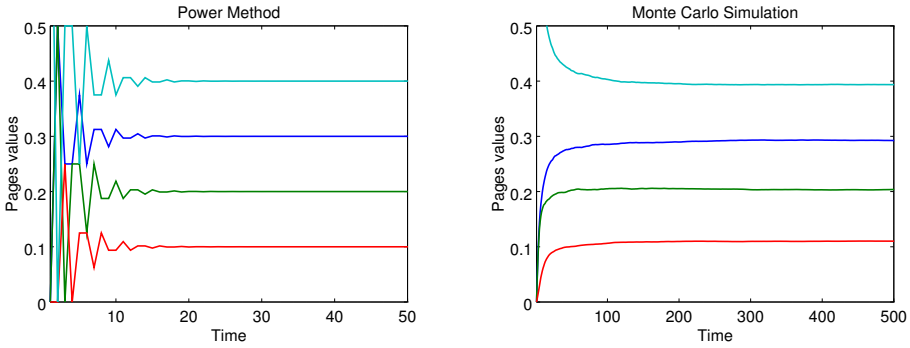

Figure 16: Comparação entre os resultados do *Power Method* e os obtidos com o método de Monte Carlo.

Considerações Finais

Desde o início das pesquisas em desenvolvimento de ferramentas de busca em 1994, conseguir fazer um ranqueamento de páginas *Web* de acordo com os interesses do usuário tem sido a parte mais desafiadora no desenvolvimento desses sistemas. Foi talvez por partir desse problema, e com uma boa solução, que o buscador Google obteve tamanho sucesso desde sua criação ao final da década de 90. As técnicas e modelos por trás do ranqueamento desse conjunto de documentos foram publicadas e, desde então, a comunidade científica tem trabalhado na pesquisa de novos modelos para tornar o cálculo do *PageRank* mais eficiente.

O cálculo do *PageRank* pode ser representado por diversos modelos, mas de forma geral é apresentado como uma equação de diferença, com algumas modificações na matriz de transição, a fim de tratar possíveis problemas de convergência. Os modelos e as propostas mais recentes do algoritmo estão voltados para o problema da distribuição do cálculo do ranquemento, no intuito de torná-lo cada vez mais eficiente. Mas a atenção ao algoritmo não tem por fim somente o ranqueamento de páginas *Web*, a ideia do cálculo do *PageRank* pode ser aplicada a diversos outros problemas.

Durante o trabalho foi realizada uma revisão dos modelos propostos para o algoritmo do *PageRank* na literatura. De forma que, inicialmente fosse efetuado um estudo dos modelos matemáticos por trás do algoritmo e em seguida fossem realizadas as simulações. Também para que fossem compreendidos os modelos matemáticos por trás do algoritmo, foi realizado um estudo sobre questões relacionadas a sistemas dinâmicos, sistemas estocásticos e probabilidade.

Neste trabalho ainda pretende-se utilizar outros métodos válidos na simulação do *PageRank*. Como a cada ano novos artigos são publicados no tema, novos modelos e técnicas estão sempre sendo publicados pela comunidade científica. Porém uma atenção maior seria dada a modelos agregados de cadeias de Markov, e problemas de consenso. E ainda como uma continuação deste trabalho, poderia ser feita uma implementação com *links* já coletados por um *Web Crawling*. De forma a por em prática os modelos num ambiente real e usando-se de técnicas de computação distribuída, afim de otimizar o cálculo e por em prática os algoritmos distribuídos. Além de usar outras linguagens de programação, como C e C++, por questões de desempenho e por serem mais adequadas a implementações em *clusters*.

Bibliografia

[1] Konstantin Avrachenkov, Nelly Litvak, Danil Nemirovsky, and Natalia Osipova. Monte carlo methods in pagerank computation: When one iteration is sufficient. *SIAM Journal on Numerical Analysis*, 45(2):890–904, 2007.

[2] Sergey Brin and Lawrence Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, 56(18):3825–3833, 2012.

[3] K. Bryan and T. Leise. The $25,000,000,000 eigenvector: The linear algebra behind Google. *SIAM Rev.*, 48(3):569–581, 2006.

[4] O. L. V. Costa, M. D. Fragoso, and R. P. Marques. *Discrete-Time Markov Jump Linear Systems*. Probability and Its Applications. Springer-Verlag, New York, 2005.

[5] H. Ishii and R. Tempo. Distributed randomized algorithms for the PageRank computation. *IEEE Trans. Automat. Control*, 55(9):1987–2002, 2010.

[6] Hideaki Ishii and Roberto Tempo. The pagerank problem, multiagent consensus, and web aggregation: A systems and control viewpoint. *Control Systems, IEEE*, 34(3):34–53, 2014.

[7] Jianjun Lei and Han-Fu Chen. Distributed randomized pagerank algorithm based on stochastic approximation. 2015.

[8] Nazar Zaki, Jose Berengueres, and Dmitry Efimov. Detection of protein complexes using a protein ranking algorithm. *Proteins: Structure, Function, and Bioinformatics*, 80(10):2459–2468, 2012.