

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Inst. de Ciências Exatas e Informática – Eng. de Computação
Disciplina de Algoritmos em Grafos

TRABALHO 2
Estudo sobre um Artigo

Professor: Will Ricardo dos Santos Machado

Aluno: Gabriel Henrique Braz

Belo Horizonte
12 de Dezembro de 2021

I – Introdução:

No mundo contemporâneo, a utilização dos serviços de Web cresce constantemente, e os usuários esperam cada vez mais um acesso mais veloz e eficiente, com mudanças constantes entre as várias páginas e hyperlinks digitais. Nesse contexto, o uso de algoritmos em grafos e suas relações pode auxiliar na melhoria do desempenho dos acessos entre os mais diversos links.

É esse cenário que serviu de motivação para a escolha do artigo “Searching Steiner trees for web graph query”, escrito pelos autores Wookey Lee, Woong-Kee Loh, e Mye M. Sohn [1].

O artigo trata da utilização de modelagem em grafos, através de árvores de Steiner, para um acesso mais eficiente dos hyperlinks e suas associações entre as mais diversas páginas e produtos Web.

Mais especificamente, trata do uso de um modelo de Programação Linear para obter resultados de busca em hyperlinks relacionados, utilizando um algoritmo sob uma árvore de Steiner.

Na análise, partem do pressuposto que a árvore de Steiner dos hyperlinks deve conter o conteúdo de maior rank e relevância para o usuário que iniciou a busca pelo motor.

No artigo, introduzem um novo estilo de motor de busca, no qual um conjunto de termos de coerência, é processado junto das relações entre hyperlinks, gerando uma estrutura de grafos, ao invés de uma lista de páginas Web individuais. Essa estrutura é gerada utilizando um algoritmo de programação linear para ranquear e gerar melhores resultados, obtidos através de análises de relevância dos nós da árvore.

II – Resumo da Metodologia:

Inicialmente partem do pressuposto que o problema de busca parte de um nodo inicial que representa o motor de busca ou o local que iniciou-se a busca, os outros nós são hyperlinks alcançáveis e que sejam representáveis em uma árvore de Steiner.

No artigo, apresentam três métodos de computar os pesos dos nós, métodos necessários para determinar a relevância dos hyperlinks encontrados:

- Método *tf-idf*: Utilizando vetores que representam os queries do usuário se obtém os pesos através de um produto interno entre esses vetores.
- Método PageRank: Desenvolvido pela Google, o método utiliza a matriz de adjacências do grafo e deriva os pesos através de uma equação aplicada sob cada elemento.
- Método bidirecional de arestas: Os dois métodos anteriores concebem apenas ou um peso global ou um peso local, no entanto esse método é capaz de obter ambas informações ao juntar valores obtidos pelos métodos anteriores em uma combinação linear. Esse resultado representa quão próximos estão os dois

objetos Web, alocados nos nós, em um dado vetor de keywords (palavras-chave).

Após anunciar os métodos de definição dos pesos, os autores explicam o modelo de programação linear utilizado para obter os top K objetos Web (como páginas por exemplo), comparando com um modelo chamado de VSM.

Os autores definem o VSM como um modelo que seleciona os nós com maiores pesos e posteriormente roda uma função que os ordena de forma decrescente e escolhe os top K nós.

O modelo linear, de acordo com os autores, permitiria a aplicação de análises de pós-otimização, o que chamam de análise de sensibilidade, que permite reduzir o overhead de realizar novamente toda a query, caso ocorram mudanças em algum dos nós.

Além disso, mencionam que um método de programação linear pode ser mais eficiente do que métodos que utilizam apenas árvores de Steiner.

Os autores apresentam um modelo de árvore de Steiner utilizando programação linear. Nela, os vértices são representados por índices e associados a eles estão parâmetros de relevância, obtidos da relação de pesos, por fim existe uma variável de decisão utilizada no processamento.

$$\begin{aligned}
 & \textbf{[Tree LP]} \\
 & \text{minimize } \sum_{\substack{ij \in N^* \\ (ij) \in A^*}} \widehat{w}_{(ij)} x_{(ij)} \\
 & \text{subject to } \sum_{\substack{i \in N^* \\ (i,0) \in A^*}} x_{(i,0)} = 0 \\
 & \sum_{\substack{i \in N^* \\ (ij) \in A^*}} x_{(ij)} = 1 \quad \text{for all } j \neq 0 \\
 & x_{(i,j)} = 0 \quad \text{for all } i \\
 & \sum_{\substack{i \in N^* \\ (ij) \in A^*}} x_{(ij)} - \sum_{\substack{j \in N^* \\ (ij) \in A^*}} x_{(ij)} = \begin{cases} +1, & \text{if } i = 0 \\ -1, & \text{if } i = t \\ 0, & \text{otherwise} \end{cases} \\
 & 0 \leq x_{(ij)} \leq 1 \quad \text{for all } i, j
 \end{aligned}$$

Figura 1: Modelo utilizado pelos autores

No modelo, funções são utilizadas para garantir o funcionamento da árvore de Steiner e para realizar o processamento. A análise específica da programação linear envolvida não será realizada nessa análise do artigo.

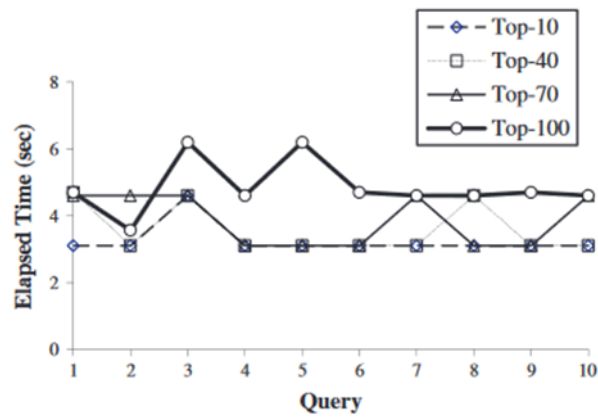


Figura 4: Tempo de processamento por query em VSM-LP, figura obtida de [1].

Após essa análise, realizaram experimentação com o algoritmo Tree_LP e os resultados foram associados ao gráfico abaixo:

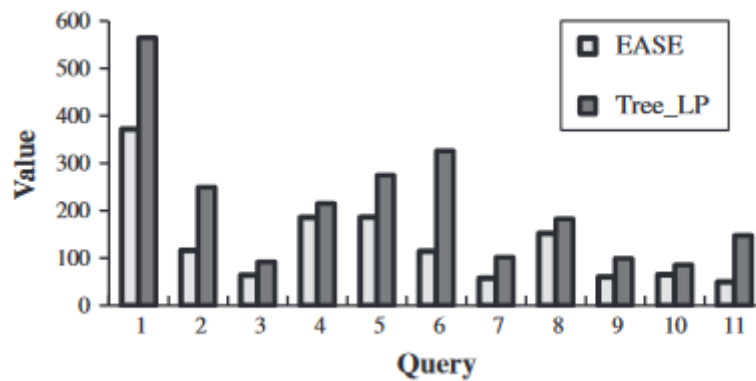


Figura 5: A performance do algoritmo Tree_LP se comparado com um de matrizes, figura retirada de [1].

Como apresentado, o algoritmo das árvores de Steiner apresenta uma performance bem melhor do que o outro algoritmo, que utiliza apenas matrizes, como apresentado pelos autores do artigo.

Na figura abaixo são apresentados os tempos de query, em quadrados brancos de um método de matrizes, que os autores definiram como um método que cria árvores de Steiner através de matrizes, e do uso do Tree_LP definido anteriormente como o método com programação linear e árvores de Steiner.

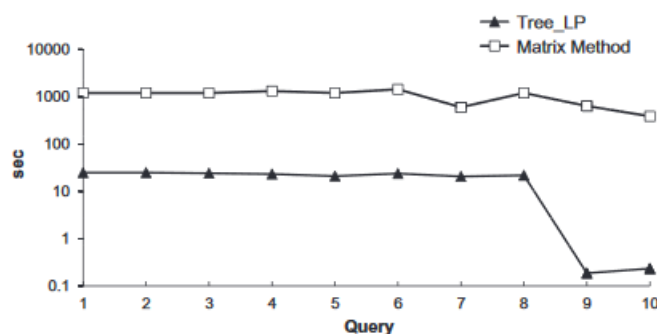


Figura 6: Tempo de query por algoritmo, figura obtida de [1].

A figura 6 mostra que o algoritmo apresentado tem maior eficiência que o utilizado para comparação.

IV – Considerações Finais:

O uso de programação linear em árvores de Steiner pelos autores gerou um algoritmo que, de acordo com os dados apresentados, tem maior eficiência que os utilizados para a comparação. Nas palavras dos autores, isso representa um novo campo de pesquisa que geraria contribuições para o uso da Programação Linear em serviços de recuperação de informações na Web com uso de representação em grafos.

Nesse sentido, os autores afirmam que o resultado pode ser expandido para toda a Web e seus motores de busca, e não apenas em um domínio específico, comprovando que o resultado é abrangente mas que deve ser aplicado a setores mais amplos da Web a fim de determinar sua eficiência.

V – Conclusão:

O artigo escolhido para a análise neste trabalho é bem específico a área de motores de busca Web, no entanto, demonstra bem a aplicabilidade e abrangência dos conceitos de Algoritmos em Grafos nos mais diversos setores da computação.

O uso das árvores de Steiner para mapear hyperlinks e suas associações é apenas uma dentre as diversas aplicações possíveis de grafos e a mistura com outras teorias, como a da programação linear, explicita como a teoria de grafos é robusta e como seu uso em áreas diversas da computação pode trazer maior eficiência na obtenção de resultados.

VI – Referencias:

[1] - Lee, Wookey, Woong-Kee Loh, and Mye M Sohn. "Searching Steiner Trees for Web Graph Query." Computers & Industrial Engineering 62.3 (2012): 732-39. Web.