Críston Pereira de Souza

# Algoritmos Eficientes para Atribuição de Hotlinks em Diretórios Web

## DISSERTAÇÃO DE MESTRADO

DEPARTAMENTO DE INFORMÁTICA
Programa de Pós–graduação em Informática

Rio de Janeiro
Janeiro de 2004

Pontifícia Universidade Católica
do Rio de Janeiro

**Críston Pereira de Souza**

# Algoritmos Eficientes para Atribuição de Hotlinks em Diretórios Web

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática do Departamento de Informática da PUC–Rio

Orientador: Prof. Eduardo Sany Laber
Co–Orientador: Prof. Artur Alves Pessoa

Rio de Janeiro
Janeiro de 2004

Pontifícia Universidade Católica
do Rio de Janeiro

**Críston Pereira de Souza**

# Algoritmos Eficientes para Atribuição de Hotlinks em Diretórios Web

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC–Rio.Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Eduardo Sany Laber**
Orientador
Departamento de Informática — PUC–Rio

**Prof. Artur Alves Pessoa**
Co–Orientador
Departamento de Informática — PUC–Rio

**Prof. Ruy Luiz Milidiú**
Departamento de Informática — PUC-RJ

**Prof. Claudson Ferreira Bornstein**
Departamento de Ciência da Computação — UFRJ

**Prof. José Eugenio Leal**
Coordenador Setorial do Centro Técnico Científico —
PUC–Rio

Rio de Janeiro, 05 de Janeiro de 2004

**Críston Pereira de Souza**

Graduou–se em Ciência da Computação na Universidade Federal da Bahia. Trabalhou em projeto de otimização de distribuição de derivados de petróleo por oleodutos, no Laboratório de Engenharia de Algoritmos e Redes Neurais da PUC–Rio.



Ficha Catalográfica

Souza, Críston Pereira de

Algoritmos Eficientes para Atribuição de Hotlinks em Diretórios Web/ Críston Pereira de Souza; orientador: Eduardo Sany Laber; co–orientador: Artur Alves Pessoa. — Rio de Janeiro : PUC–Rio, Departamento de Informática, 2004.

v., 63 f: il. ; 29,7 cm

1. Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Teses. 2. Internet. 3. Hotlinks. 4. Algoritmos Aproximativos. 5. Programação Inteira. 6. Programação Dinâmica. I. Laber, Eduardo Sany. II. Pessoa, Artur Alves. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

## Agradecimentos

Ao meu orientador, Prof. Eduardo Laber, e ao meu co-orientador, Prof. Artur Pessoa, pela atenção, incentivo e compreensão. A excelente orientação que recebi, fruto da dedicação e motivação destes pesquisadores, foi a principal responsável pela realização deste trabalho.

À CAPES e à PUC–Rio pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

À minha mãe Silene, a pessoa que mais se sacrificou para que um dia este resultado fosse alcançado. Aos meus familiares, em especial minha avó Maria, que estiveram torcendo para que tudo desse certo no Rio.

À Renata, alguém que apesar da distância, não poupou paciência e atenção.

Aos amigos, principalmente Aurisan, Ricardo, Margaret, Dionísio, Jovânia, Fabio, Roberto e Elder, cuja companhia tornou a distância de casa menos sofrida.

Às pessoas que compõe o Laboratório de Sistemas Distribuídos da Universidade Federal da Bahia, principalmente Aline, Macêdo, Flávio e George, pois despertaram em mim o interesse pela atividade acadêmica.

Ao pessoal do Departamento de Informática pela ajuda e orientação.

## Resumo


Souza, Críston Pereira de; Laber, Eduardo Sany; Pessoa, Artur Alves. **Algoritmos Eficientes para Atribuição de Hotlinks em Diretórios Web**. Rio de Janeiro, 2004. 63p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Uma maneira de localizar uma informação em uma base de dados grande e caótica como a Internet é utilizar um índice hierárquico que respeita alguma maneira de categorizar os dados. Exemplos desta hierarquia são os serviços de diretório, comuns em sites de busca. Porém, esta abordagem pode apresentar algumas desvantagens, como a necessidade de percorrer muitas páginas até chegar em uma informação muito acessada. Uma maneira de tratar este problema é o uso de *hotlinks*, hyperlinks adicionais que servem como atalho em uma busca. Estudamos algoritmos eficientes para atribuir hotlinks em um diretório web, de modo a reduzir o número máximo ou o número médio de acessos em uma busca. Fornecemos para o problema de minimização do número máximo de acessos um algoritmo (14/3)-aproximado e um algoritmo polinomial exato baseado em programação dinâmica. Por outro lado, para o problema de minimizar o número médio de acessos, adaptamos o algoritmo exato do problema anterior. Entretanto, este algoritmo adaptado é polinomial apenas para sites representados por árvores com altura $O(\log n)$. Por isso, introduzimos um parâmetro que permite ao usuário reduzir o tempo de execução em detrimento da qualidade da solução. Para este problema de minimizar o número médio de acessos, realizamos também experimentos comparando nosso algoritmo, um modelo em programação inteira, e alguns algoritmos propostos por outros autores. Introduzimos modificações práticas que melhoraram a performance do nosso algoritmo.



## Palavras–chave

Internet; hotlinks; algoritmos aproximativos; programação inteira; programação dinâmica.

## Abstract

Souza, Críston Pereira de; Laber, Eduardo Sany; Pessoa, Artur Alves. **Efficient Hotlinks Assigment Algorithms for Web Directories**. Rio de Janeiro, 2004. 63p. MSc. Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

An approach to search an information in a large and chaotic data base like the Internet is to use a hierarquical index regarding some categorization of the data. As an example, we have the web directories, usually found in search engines. However, this approach may have problems, as the need of visiting too many web pages to find a very accessed information. A way to address this problem is the use of *hotlinks*, which are hyperlinks added to the web site and used as shortcuts in a search. We studied efficient algorithms to assign hotlinks in web directories, in such a way to minimize the maximum or the average number of accesses to find an information. For the problem of minimizing the maximum number of accesses, we provide an (14/3)-approximation algorithm and an exact polinomial time algorithm based on dynamic programming. On the other hand, for the problem of minimizing the expected number of accesses, we adapted the previous exact algorithm. However, this adapted algorithm is polinomial only for web sites represented by trees with height $O(\log n)$. So, we introduce a parameter that allows the user to reduce the execution time under the cost of reducing the solution quality. For this problem of minimizing the expected number of accesses, we also made experiments comparing our algorithm, an integer programming model, and some algorithms proposed by other authors. We introduce pratical changes that improved the performance of our algorithm.

## Keywords

Internet; hotlinks; approximation algorithms; integer programming; dynamic programming.

# Conteúdo


1 Introdução **11**
1.1 Motivação 11
1.2 Definição do problema 12
1.3 Trabalhos relacionados 16
1.4 Organização 19

2 Propriedades estruturais **20**
2.1 Limites superiores 20
2.2 Limites inferiores 25

3 Algoritmos para o PBH **27**
3.1 Algoritmo APPROX 27
3.2 Algoritmo PATH 32

4 Algoritmos para o MBH **41**
4.1 Algoritmo GREEDY-BFS 41
4.2 Algoritmo KRANAKIS 41
4.3 Modelo em Programação Inteira 42
4.4 Algoritmo M-PATH 44

5 Experimentos **46**
5.1 Ambiente e instâncias 46
5.2 Aspectos de implementação 49
5.3 Resultados 54

6 Conclusões e trabalhos futuros **61**

Referências Bibliográficas **62**

# Lista de Figuras


1.1 Página web que corresponde a um nó interno na árvore que representa o diretório do Yahoo. 11
1.2 Página web que corresponde a um nó folha na árvore que representa o diretório do Yahoo. 12
1.3 (a) a árvore $T$. (b) a árvore $T^A$, onde $A = \{(u, v), (z, w)\}$. 13

2.1 Pseudo-código do algoritmo ASSIGN-HOTLINK. 21
2.2 (a) Árvore $T$. (b) Árvore $T^{\{(r,u)\}}$ obtida pelo procedimento LOG após adicionar o hotlink $(r, u)$ em $T$ 22
2.3 (a) Árvore melhorada, depois de executar ASSIGN-HOTLINK. (b) Árvore obtida da árvore anterior depois de executar o algoritmo LOG Modificado nas sub-árvore com apenas nós internos. 24

3.1 (a) e (b) mostram os valores de $\mathbf{w}$ para as árvores $T$ e $T^{\{(r,u)\}}$, respectivamente. Os nós sombreados na Figura 3.1.(a) são os nós da sub-árvore binária, com raiz em $r$, com o maior número de hotleaves. 28
3.2 Os nós $s_i$ e $s'_i$ no caminho que conecta $u$ à raiz de $T'$. 29
3.3 (a) árvore com raiz em $r$ e $f$ filhos. (b) par de sub-problemas gerado pela escolha de apontar o hotlink de $r$ para a sub-árvore filha $f$. (c) par de sub-problemas gerado pela escolha de apontar o hotlink de $r$ para alguma outra sub-árvore filha. 33
3.4 (a) uma instância do problema P-PBH. (b) a árvore melhorada obtida pela adição do hotlink $(q_2, r)$. (c) o sub-problema correspondente gerado no caso 1. 35
3.5 (a) uma instância do problema P-PBH. (b) e (c) a decomposição no caso 2, quando $c = (0, 1, 0, 0, 1)$. 36
3.6 Pseudo-código do algoritmo APPROX. 40

4.1 Pseudo-código do algoritmo GREEDY-BFS. 42
4.2 Pseudo-código do algoritmo KRANAKIS. 42
4.3 As quatro configurações possíveis para um par de hotlinks em um caminho. Os hotlinks em (a) são "aninhados". Os hotlinks em (b) são "cruzados". 43

5.1 (a) Instância do P-MBH. (b) Árvore resultante da atribuição de hotlinks $A$. 50
5.2 Ganho médio (%) de cada algoritmo em função de $H$. 56
5.3 Diferença da solução de cada algoritmo em relação a solução do algoritmo M-PATH. 56
5.4 Ganho (%) e uso de memória (MB) do algoritmo M-PATH, variando o parâmetro $D$, para as instâncias 13 (em cima) e 73 (em baixo). 60

# Lista de Tabelas


5.1 As instâncias obtidas dos sites de universidades brasileiras e americanas. 48

5.2 Ganho ótimo de cada instância, ganho e tempo de execução dos algoritmos GREEDY-BFS e KRANAKIS, tempo de execução do MIP. Para o M-PATH dispondo de 16MB para armazenar a tabela da programação dinâmica, o tempo de execução, o tamanho da tabela em KB, e a redução percentual de memória com as melhorias (redução do número de sub-problemas e enfraquecimento da restrição de altura). 58

5.3 Tempo de execução do algoritmo MF-PATH e memória disponível para a tabela da programação dinâmica, para as instâncias 63, 70 e 78. 59

5.4 Memória utilizada na tabela da programação dinâmica, tempo de execução, ganho e a altura da árvore melhorada resultante do algoritmo M-PATH, para $D$ variando de 1 até 16, para as instâncias 13 e 73. Memória disponível limitada em 1GB. Tempo de execução limitado em aproximadamente 1 hora. O "-" indica que o valor não está disponível, pois algum limite foi violado. 59

... pleasure has probably been the main goal all along. But I hesitate to admit it, because computer scientists want to maintain their image as hard-working individuals who deserve high salaries. Sooner or later society will realise that certain kinds of hard work are in fact admirable even though they are more fun than just about anything else.

**D. E. Knuth**, *The Stanford Graphbase: a platform for combinatorial computing.*