O Problema do k-Servidor

Autor: Mário César San Felice
Orientador: Orlando Lee

22 de março de 2010, IC-Unicamp

Estrutura

A seguir temos a estrutura geral da dissertação.

- Otimização combinatória
- Computação *online*
- Análise competitiva
- Problema do k-Servidor
- Algoritmo da função trabalho

Problemas de Otimização

Um problema de otimização combinatória é um problema de maximização ou minimização que:

- Possui um conjunto de entradas.
- Possui um conjunto de soluções.
- Para cada entrada I e solução O apresenta um custo $c(I, O)$.

Modelos de Computação

Problemas de otimização podem ser abordados por algoritmos de diferentes modelos.

- **Computação Offline** - Neste modelo o algoritmo recebe todas as partes da entrada antes de começar a construir uma solução para o problema.
- **Computação Online** - Neste modelo o algoritmo recebe as partes da entrada em sequência e para cada parte recebida constrói uma parte da solução, sem conhecimento das partes que estão por vir.

Chamamos algoritmos destes modelos, respectivamente, de algoritmos *offline* e *online*.

Áreas de Interesse

Por simplicidade, quando consideramos um problema no modelo de computação *offline* ou *online* dizemos, respectivamente, tratar-se de um problema *offline* ou *online*.

Problemas *online* são comuns em diversas áreas, como:

- Ciência da computação.
- Economia.
- Pesquisa Operacional.

Problemas em Ciência da Computação

Em ciência da computação algoritmos para problemas *online* surgiram em diversos contextos.

- **Problemas Clássicos** - Como os problemas de Escalonamento, Empacotamento e Balanceamento de Cargas.
- **Problemas Tipicamente Online** - Como os problemas de Paginação em Memória Virtual e de Roteamento em Redes de Comunicação.
- **Gerenciamento de Estruturas de Dados Dinâmicas** - Como o problema de Acesso à Lista.

Abordagens de Análise

Existem abordagens específicas para analisar a qualidade de algorimos *online*.

- **Complexidade de Caso Médio** - Abordagem tradicional em que se estabelece uma distribuição de eventos e calcula-se o custo total esperado ou custo esperado por evento.
- **Análise Competitiva** - Abordagem em que se obtém uma garantia de qualidade para as soluções do algoritmo *online* analisado quando estas são comparadas às soluções do algoritmo ótimo para o problema *offline* correspondente.

Razão de Competitividade

A garantia de qualidade utilizada em análise competitiva é expressa por uma razão, chamada razão de competitividade.

Seja ALG um algoritmo *online* para um problema e OPT um algoritmo ótimo para o problema *offline* correspondente.

Dizemos que ALG é c-competitivo se, para toda entrada I e algum α constante que não depende de I, ele apresenta razão de competitividade c que satisfaz a seguinte expressão:

$$ALG(I) \leq cOPT(I) + \alpha.$$

Análise como um Jogo

A abordagem de análise competitiva pode ser vista como um jogo entre um jogador *online* e um adversário malicioso.

- O jogador *online* é o algoritmo *online* cuja qualidade será avaliada. Seu objetivo é minimizar o custo de suas soluções.
- O adversário malicioso conhece o comportamento do algoritmo *online* e constrói uma entrada I^* que maximiza a razão entre $ALG(I^*)$ e $OPT(I^*)$.

Como analisamos ALG considerando a "pior entrada" I^* dizemos que análise competitiva é uma abordagem de pior caso.

Exemplo - Balanceamento de Cargas

Para explicar melhor a abordagem de análise competitiva consideramos o problema de Balanceamento de Cargas. Este problema pode ser definido da seguinte forma:

- Temos N máquinas capazes de realizar tarefas.
- As máquinas devem processar uma sequência de tarefas $\sigma = t_1, t_2, \ldots, t_n$.
- Cada tarefa t_i tem um tamanho próprio e deve ser alocada em uma máquina.
- A tarefa t_{i+1} só surge depois que t_i é alocada.
- Cada máquina possui uma carga igual ao somatório dos tamanhos das tarefas alocadas nela.
- O objetivo é alocar as tarefas de σ nas máquinas de modo a minimizar a carga máxima dentre estas.

Exemplo - Algoritmo *Online* Guloso

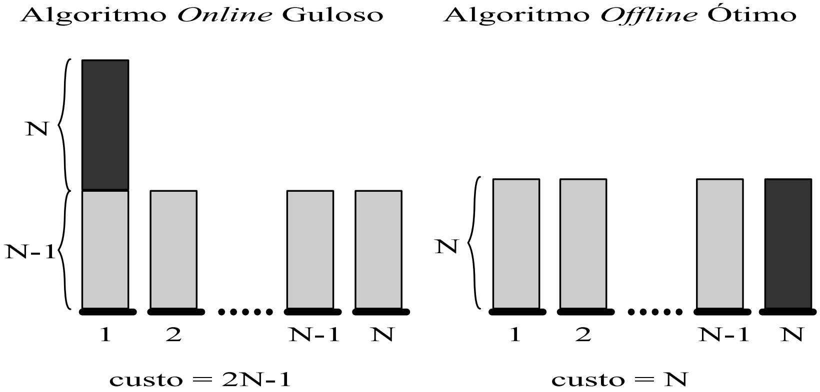
Neste exemplo consideramos um algoritmo *online* guloso com o seguinte comportamento:

"Quando recebe uma tarefa o algoritmo a aloca na máquina que possui a menor carga. Em caso de empate escolhe arbitrariamente uma máquina."

A seguir mostraremos que este algoritmo é $\left(2 - \frac{1}{N}\right)$-competitivo.

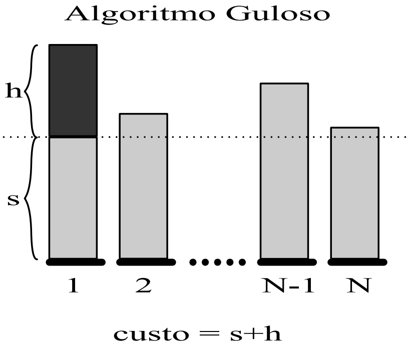
Exemplo - Limitante Inferior

Para mostrar o limitante inferior consideramos uma sequência de tarefas σ com $N(N-1)$ tarefas de tamanho 1 seguida de uma tarefa de tamanho N.



Exemplo - Limitante Superior

Para mostrar o limitante superior consideramos que para uma entrada σ o algoritmo *online* guloso alocou uma carga máxima igual a $s + h$ na máquina 1.



Temos que $OPT(\sigma) \geq h$ e $OPT(\sigma) \geq s + \frac{h}{N}$.

Exemplo - Limitante Superior

Logo,

$$\begin{aligned} ALG(\sigma) &= s + h \\ &\leq OPT(\sigma) - \frac{h}{N} + h \\ &= OPT(\sigma) + \left(1 - \frac{1}{N}\right) h \\ &\leq OPT(\sigma) + \left(1 - \frac{1}{N}\right) OPT(\sigma) \\ &= \left(2 - \frac{1}{N}\right) OPT(\sigma). \end{aligned}$$

O Problema do k-Servidor

O problema do k-Servidor é o principal problema tratado na dissertação.

- Ele foi proposto por Manasse, McGeoch e Sleator (1988).
- É central na área de computação *online*, sendo uma generalização do problema de Paginação em Memória Virtual.
- Apresenta problemas importantes em aberto, como a conjectura do k-Servidor.

Definição do Problema

O problema do k-Servidor é definido da seguinte forma:

- Temos k-Servidores num espaço métrico.
- Os servidores devem atender uma sequência de requisições $\sigma = r_1, r_2, \ldots, r_n$.
- Cada requisição r_i é um ponto do espaço e é atendida se um servidor está sobre ela.
- A requisição r_{i+1} só surge depois que r_i é atendida.
- O objetivo é minimizar o somatório das distâncias percorridas pelos servidores para atender σ.

Conjectura do k-Servidor

Esta conjectura é responsável por grande parte do interesse no problema do k-Servidor.

- Ela foi proposta por Manasse, McGeoch e Sleator (1988).
- Diz que: "qualquer espaço métrico admite um algoritmo determinístico k-competitivo para o problema do k-Servidor".
- Koutsoupias e Papadimitriou (1995) mostraram que o algoritmo da função trabalho WFA apresenta razão de competitividade $2k-1$ para o problema do k-Servidor em qualquer espaço métrico.
- Existem diversos casos particulares do problema do k-Servidor para os quais a conjectura vale.
- Sabe-se que nenhum espaço métrico com pelo menos $k+1$ pontos permite um algoritmo determinístico com razão de competitividade menor que k.

Casos Particulares Importantes

A seguir temos alguns casos particulares para os quais a conjectura vale seguidos dos algoritmos que comprovadamente a satisfazem.

- **Problema do k-Servidor na Linha** - Algoritmo das Duas Coberturas DC e algoritmo WFA.
- **Problema do k-Servidor na Estrela Ponderada** - Algoritmo WFA.
- **Problema do k-Servidor na Árvore** - Algoritmo das Coberturas Múltiplas DC-Tree.
- **Problema do k-Servidor em Espaços com $k+2$ Pontos** - Algoritmo WFA.
- **Problema do 2-Servidor** - Algoritmo WFA.
- **Problema do 3-Servidor no Plano Manhattan** - Algoritmo WFA.

Exemplo - Algoritmo *Online* Guloso

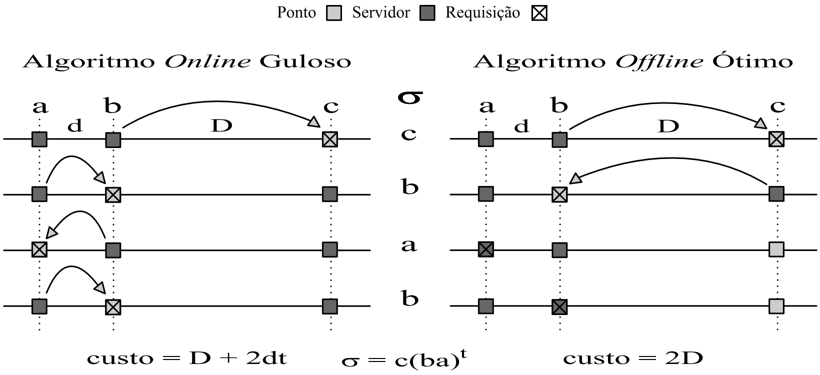
Um algoritmo *online* guloso para o problema do k-Servidor apresenta o seguinte comportamento:

Quando recebe uma requisição o algoritmo a atende com o servidor mais próximo. Em caso de empate escolhe arbitrariamente um servidor.

A análise deste algoritmo é interessante, pois demonstra uma dificuldade típica do problema do k-Servidor.

Exemplo - Limitante Inferior

Na figura a seguir mostraremos que a estratégia gulosa não é competitiva mesmo em um caso bastante simples do problema do *k*-Servidor.



Algoritmo da Função Trabalho

A seguir temos alguns fatos importantes sobre este algoritmo.

- A idéia do algoritmo foi proposta independentemente por Howard Karllof e por Mcgeoch e Sleator.
- Chrobak e Larmore (1992) apresentaram o algoritmo como o conhecemos e apresentaram a primeira prova de competitividade deste, para o caso $k = 2$.
- Koutsoupias e Papadimitriou (1995) mostraram que o algoritmo é $(2k - 1)$-competitivo para o problema do k-Servidor em qualquer espaço métrico.
- Desde então foram obtidas provas da k-competitividade de WFA para diversos casos particulares.

Definição Função Trabalho

A seguir definimos funções trabalho, utilizadas na definição de WFA.

- Uma função trabalho é definida sobre uma configuração inicial C_0 e uma sequência de requisições σ.
- Sua entrada é uma configuração C, e denotamos seu valor por $w_\sigma(C)$.
- Podemos interpretá-la como o custo ótimo para, começando com os servidores em C_0, atender σ e terminar com os servidores em C.

Definição do Algoritmo

Sendo C_σ a configuração de WFA depois de atender uma sequência de requisições σ, o algoritmo da função trabalho apresenta o seguinte comportamento:

Quando, depois que WFA atendeu σ, surge uma requisição r, WFA atende esta requisição com o servidor s^* que satisfaz:

$$s^* = \arg\min_{s \in C_\sigma} \{ w_\sigma(C_\sigma - s + r) + d(r, s) \}.$$

Esta expressão permite perceber como é difícil analisar este algoritmo, pois a cada passo seu comportamento depende do valor de diversas funções trabalho, cujos valores dependem dos passos anteriores.

Considerações Finais

Neste trabalho estudamos o problema do k-Servidor, dedicando especial atenção à conjectura do k-Servidor, e por consequência ao algoritmo da função trabalho.

Assim, este trabalho consiste de uma compilação dos principais resultados relevantes para a conjectura.

Nesta compilação contribuímos com um paradigma que permitiu identificar partes comuns às provas e assim simplificá-las.

Acreditamos que o objetivo inicial do trabalho foi alcançado e esperamos que esta dissertação seja útil para estudantes e pesquisadores com interesse na área de computação *online*.

Referências

Y. Bartal and E. Koutsoupias.
On the competitive ratio of the work function algorithm for the k-server problem.
Theor. Comput. Sci., 324(2-3):337–345, 2004.

W. Bein, M. Chrobak, and L.L. Larmore.
The 3-server problem in the plane.
In *ESA '99: Proceedings of the 7th Annual European Symposium on Algorithms*, pages 301–312.
Springer-Verlag, 1999.

A. Borodin and R. El-Yaniv.
Online Computation and Competitive Analysis.
Press Syndicate of the University of Cambridge, 1998.

Referências

M. Chrobak, H. Karloff, T. Payne, and S. Vishwanathan.
New results on server problems.
In *SODA '90: Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 291–300. Society for Industrial and Applied Mathematics, 1990.

M. Chrobak and L.L. Larmore.
An optimal on-line algorithm for k-servers on trees.
SIAM J. Comput., 20(1):144–148, 1991.

M. Chrobak and L.L. Larmore.
The server problem and on-line games.
DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 7:11–64, 1992.

Referências

M. Chrobak and L.L. Larmore.
Metrical task systems, the server problem and the work function algorithm.
In *Developments from a June 1996 seminar on Online algorithms*, pages 74–96. Springer-Verlag, 1998.

A. Fiat, R.M. Karp, M. Luby, L.A. Mcgeoch, D.D. Sleator, and N.E. Young.
Competitive paging algorithms.
Technical report, CMU-CS, 88-196, Computer Science Department, Carnegie-Mellon University, 1991.

E. Koutsoupias.
Weak adversaries for the k-server problem.
In *FOCS '99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, page 444. IEEE Computer Society, 1999.

Referências

E. Koutsoupias and C.H. Papadimitriou.
On the k-server conjecture.
J. ACM, 42(5):971–983, 1995.

E. Koutsoupias and C.H. Papadimitriou.
The 2-evader problem.
Inf. Process. Lett., 57(5):249–252, 1996.

M.S. Manasse, L.A. McGeoch, and D.D. Sleator.
Competitive algorithms for on-line problems.
In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 322–333. ACM, 1988.

Referências

M.S. Manasse, L.A. McGeoch, and D.D. Sleator.
Competitive algorithms for server problems.
J. Algorithms, 11(2):208–230, 1990.

V.V. Vazirani.
Approximation Algorithms.
Springer, 2001.