

Universidade Federal Fluminense

NOME DO AUTOR NESTA FORMA

Título da Dissertação e Nesta Forma

NITERÓI

2011

NOME DO AUTOR NESTA FORMA

Título da Dissertação e Nesta Forma

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Nome Desta Forma.

Orientador:

Nome do Professor Nesta Forma

Coorientador:

Nome do Professor Nesta Forma

Nome do Professor Nesta Forma

UNIVERSIDADE FEDERAL FLUMINENSE

NITERÓI

2011

Título da Dissertação e Nesta Forma

Nome do Autor Nesta Forma

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia. Área de Concentração: Nome Desta Forma.

Aprovada por:

Prof. Nome Nesta Forma, Ph.D. / IC-UFF(Presidente)

Prof. Nome Nesta Forma, D.Sc. / IM-UFRJ

Prof. Nome Nesta Forma, Ph.D. / IC-UFF

Prof. Nome Nesta Forma, D.Sc. / IF-UFRRJ

Prof. Nome Nesta Forma, Ph.D. / DCC-UFMG

Prof. Nome Nesta Forma, D.Sc. / LNCC

Volta Redonda, 07 de Agosto de 2011.

Dedicatoria. Por exemplo: Para meus pais e familia

Agradecimentos

Para Deus, pela saúde e pela ajuda que mais ninguém pode dar.

A minha família, em especial a minha mãe Joaquina, minha irmã Julia e meu tio Lorenzo pelo apoio e incentivo.

E assim vai agradecendo a quem quizer, cada linha pode ser um agradecimento em específico.

A todos os demais que contribuíram para a realização deste trabalho.

Ao Prof. Tiago Neves do VCE/EEIMVR/UFF pela gentileza de doar este TEMPLATE.

Resumo

Aqui deve ser descrito sucintamente o trabalho de forma bem objetiva. Por exemplo: Uma Rede de Distribuição de Conteúdos (RDC) é composta por computadores interligados através da internet que colaboram para fornecer conteúdos para clientes. Ela é uma rede sobreposta que mantém réplicas de cada conteúdo em seus servidores, com o objetivo de reduzir os atrasos, a carga nos servidores e o congestionamento da rede, melhorando a qualidade do serviço fornecido. Em arquiteturas de RDC tradicionais, as requisições são recebidas por um servidor central e redirecionadas para o servidor mais próximo do cliente. Entretanto, devido aos custos de manutenção das réplicas, não é razoável replicar os conteúdos em toda a rede. Neste trabalho são apresentadas abordagens exatas e heurísticas para resolver um problema de gerenciamento das RDC, conhecido na literatura como Problema de Posicionamento de Réplicas. Este problema consiste em decidir quais réplicas serão colocadas em cada servidor. O objetivo final é minimizar o tráfego na rede sem violar as restrições de qualidade de serviço. Ao todo, são propostas duas abordagens exatas para a versão *offline* do problema, sendo uma formulação matemática e um algoritmo de enumeração, e diversas heurísticas e métodos híbridos, que usam conceitos de métodos exatos e heurísticos simultaneamente, para a versão *online* do mesmo. Resultados indicam que as heurísticas propostas obtêm soluções de boa qualidade e proporcionam uma redução de custo de até cinco ordens de grandeza quando comparadas a heurística usada em RDCs reais.

Abstract

Aqui o resumo anterior deve ser escrito em inglês. Por exemplo: A content distribution network (CDN) is composed of computers networked together across the Internet that cooperate to deliver content to clients. It is an overlay network that maintains replicas of each content in its servers, with the goal of reducing delays, server load and network congestion, therefore improving the quality of the service provided. In traditional CDN architectures, requests are received by a central server and then redirected to the server that is closer to the client. However, because of the costs involved in the maintenance of such replicas, it is not reasonable to replicate the contents over the entire network. In this work, exact and heuristic approaches are presented to solve a problem that appears in CDN management, known in the literature as the Replica Placement Problem. This problem consists of deciding which replicas will be placed in each server. The overall objective is to minimize the traffic in the network without violating quality constraints. Two different exact approaches are proposed to the offline version of the problem: a mathematical formulation and an enumeration algorithm. Besides that, several hybrid methods, that use exact and heuristic concepts simultaneously, are also proposed to the online version of the problem. Results indicate that the proposed heuristics can reduce operational costs in up to five orders of magnitude when compared to the heuristic used in real world CDN.

Palavras-chave

1. Redes de Distribuição de Conteúdos
2. Formulação Matemática
3. Heurísticas
4. Problema de Posicionamento de Réplicas e Distribuição de Requisições
5. Qualidade de Serviço
6. Método Híbrido
7. Meta-heurísticas

Glossário

| | | |
|--------|---|--|
| AEBH | : | Alisamento Exponencial Biparamétrico de Holt |
| AELB | : | Alisamento Exponencial Linear de Brown |
| AGAP | : | <i>Adapted Generalized Assignment Problem</i> |
| ASP | : | <i>Alternative Shortest Path</i> |
| Bote | : | <i>Based on Trends Estimator</i> |
| DNS | : | <i>Domain Naming Service</i> |
| GAS | : | <i>Greedy Allocation per Server</i> |
| GAP | : | <i>Generalized Assignment Problem</i> |
| GHS | : | <i>Global Hosting System</i> |
| HC | : | <i>Hybrid Constructive</i> |
| HCAGAP | : | <i>HC + AGAP</i> |
| HNH | : | <i>Hybrid Network Heuristic</i> |
| IC | : | Instituto de Computação |
| ILS | : | <i>Iterated Local Search</i> |
| ILSAM | : | <i>Iterated Local Search with Adaptive Memory</i> |
| LRU | : | <i>Least Recently Used</i> |
| PFCM | : | Problema de Fluxo de Custo Mínimo |
| PPR | : | Problema de Posicionamento de Réplicas |
| PDR | : | Problema de Distribuição de Requisições |
| PPRDR | : | Problema de Posicionamento de Réplicas e Distribuição de Requisições |

Glossário

| | | |
|------|---|---|
| PLFC | : | Problema de Localização de Facilidades Capacitado |
| PLS | : | Problema de Localização de Servidores |
| PR | : | Problema de Replicação |
| P2P | : | <i>Peer-to-Peer</i> |
| QoS | : | <i>Quality of Service</i> |
| RDC | : | Rede de Distribuição de Conteúdos |
| RTR | : | <i>Record-To-Record</i> |
| RTT | : | <i>Round Trip Time</i> |
| SEA | : | <i>Smart Enumeration Algorithm</i> |
| UFF | : | Universidade Federal Fluminense |
| URL | : | <i>Uniform Resource Locator</i> |

Sumário

| | | |
|----------|---|-----------|
| 1 | Introdução | 11 |
| 2 | Redes de Distribuição de Conteúdos | 14 |
| 3 | O Problema de Posicionamento de Réplicas e Distribuição de Requisições | 16 |
| 4 | Trabalhos Relacionados | 18 |
| 5 | Abordagens Exatas | 20 |
| 5.1 | Formulação para o PPRDR Estático | 20 |
| 5.2 | Formulação para o PPRDR Dinâmico e com Demanda Divisível | 21 |
| 5.3 | Formulação para Atendimento Maximizado | 21 |
| 5.4 | Formulação com Múltiplas Origens de Conteúdos | 22 |
| 5.5 | Formulação com Submissão e Remoção de Conteúdos | 22 |
| 5.6 | Um Algoritmo Enumerativo para o PPRDR | 22 |
| 6 | Abordagens Heurísticas | 25 |
| 6.1 | <i>Framework</i> Unificador | 25 |
| 6.2 | Abordagens para o PDR | 25 |
| 6.2.1 | Formulação Matemática para o PDR | 26 |
| 6.2.2 | Atendimento Local | 26 |
| 6.3 | Técnicas para Estimar a Demanda Futura | 26 |
| 6.3.1 | Estimador Baseado em Médias | 26 |
| 6.3.2 | Estimadores Baseado em Alisamento Exponencial | 26 |

| | | |
|----------|---|-----------|
| 7 | Instâncias de Testes | 27 |
| 8 | Resultados Computacionais | 28 |
| 8.1 | Resultados das Abordagens Exatas | 28 |
| 8.2 | Síntese dos Resultados para a Versão <i>Online</i> do PPRDR | 28 |
| 9 | Conclusões e Trabalhos Futuros | 30 |
| 9.1 | Conclusões | 30 |
| 9.2 | Trabalhos Futuros | 30 |
| | Referências | 31 |

Capítulo 1

Introdução

Este documento oferece informações sobre o formato que deve ser adotado para elaborar a dissertação do curso de mestrado do Programa de Pós-graduação MCCT. Somente dissertações nos moldes deste template serão aceitas. **Logo, sugerimos com muito ênfase que as instruções presentes neste template sejam usadas para escrever a sua DISSERTAÇÃO.**

Este TEMPLATE deve ser compilado em LATEX. Para isto seu computador deve ter instalado os respectivos software para WINDOWS ou para LINUX. Maiores informações podem ser encontradas em:

<http://www.latex-project.org/>

ou para WINDOWS em:

<http://miktex.org/>

Note que a formatação das páginas e parágrafos já está previamente definida. Apenas se deve incluir o texto de forma a não alterar esta formatação.

As referências devem ser citadas usando as normas ABNT. Ou seja, a citação no texto da referência [5, 6] deve ser feita entre parêntesis assim (Tenzakhti et al., 2004; Zhang et al., 1998) ou sem parêntesis assim veja Tenzakhti et al.(2004); Zhang et al.(1998). Não devem ser incluídas referências que não sejam citadas no texto.

E assim continua o texto até terminar este capítulo. Por exemplo, a seguir colocamos um texto correspondente a uma tese de doutorado para completar o primeiro capítulo.

Em arquiteturas de RDC tradicionais, as requisições dos clientes são recebidas por um servidor central que as redireciona para outros servidores capazes de atender a requisição

e que se encontrem nas proximidades do cliente. Na realidade, as requisições dos clientes nem sempre serão atendidas pelo servidor mais próximo. Em muitos casos, é mais vantajoso usar um servidor que se encontra um pouco mais distante, mas que esteja menos sobrecarregado.

O problema que é abordado neste trabalho, chamado *Problema de Posicionamento de Réplicas e Distribuição de Requisições* ou PPRDR, é uma variante do PPR e consiste em encontrar os melhores locais para posicionar as réplicas dos conteúdos e definir quais servidores irão atender cada uma das requisições a fim de que as exigências de QoS das mesmas não sejam violadas e o tráfego dentro da rede seja minimizado.

O trabalho aqui proposto trata de maneira conjunta o PR, o PPR e o PDR para arquivos extensos, analisando não apenas os custos para associar as requisições aos servidores, mas também uma série de questões que até então não haviam sido abordadas de maneira conjunta, como banda mínima para os clientes, carga nos servidores e presença de múltiplos conteúdos. Além dessas questões, uma série de outras, relacionadas ao fato dos conteúdos serem extensos, como possibilidade de uma requisição ser atendida por mais de um servidor ao mesmo tempo e a possibilidade do atendimento a uma requisição se estender por vários períodos de tempo, também são consideradas neste trabalho.

Dentre as contribuições deste trabalho estão:

1. Formulações matemáticas para várias versões do PPRDR, dentre as quais destaca-se a formulação *FD*, que contempla diversas características reais do problema tais como: versão dinâmica do PPRDR, atendimento otimizado dos clientes, divisão de requisições em períodos, atendimento de requisições por múltiplos servidores, submissão e remoção de conteúdos.
2. Um algoritmo de Enumeração para resolução de instâncias de grande porte.
3. Duas versões da meta-heurística *Iterated Local Search* para a versão *offline* do problema.
4. Um *framework* unificador para heurísticas construtivas para a versão *online* do PPRDR tratado pela formulação *FD*.
5. Abordagens exatas e heurísticas para o PPR.
6. Abordagens exatas e heurísticas para o PDR.
7. Análise da complexidade do PDR, que prova que este problema pertence à classe *P*.

8. Abordagens híbridas compostas como combinação de abordagens para o PPR e para o PDR.
9. Versões otimizadas (através da inclusão de componentes exatos) de heurísticas utilizadas em RDCs reais e apresentadas pela literatura.

O restante deste trabalho está organizado da seguinte maneira: no Capítulo 2 é feita uma breve exposição sobre o funcionamento de uma RDC. O Capítulo 3 faz uma descrição detalhada do PPRDR. O Capítulo 4 apresenta uma revisão bibliográfica do tema. O Capítulo 5 expõe uma discussão a respeito de algumas abordagens exatas para o problema. Já no Capítulo 6 são apresentadas as abordagens heurísticas centralizadas propostas para o problema. O Capítulo 7 mostra como foram geradas as instâncias para o problema bem como suas principais características. O Capítulo 8 expõe os resultados obtidos, bem como uma análise dos mesmos, e no Capítulo 9 encontram-se as conclusões do trabalho e algumas propostas para trabalhos futuros.

Capítulo 2

Redes de Distribuição de Conteúdos

As figuras devem ser colocadas como nos dois exemplos a seguir. O programa fornece impressoras laser branco e preto. A impressão de figuras coloridas é responsabilidade do aluno.

A Figura 2.1 ilustra como é a estrutura de um sistema cliente servidor no paradigma da internet. Nela são mostrados três clientes (C1,C2 e C3) requisitando dois conteúdos diferentes para um único servidor central (SC). As requisições são indicadas pela letra *Q* seguida de um número, assim, a sequência *Q1* representa uma requisição para o conteúdo 1, e a sequência *Q2* representa requisições para o conteúdo 2.

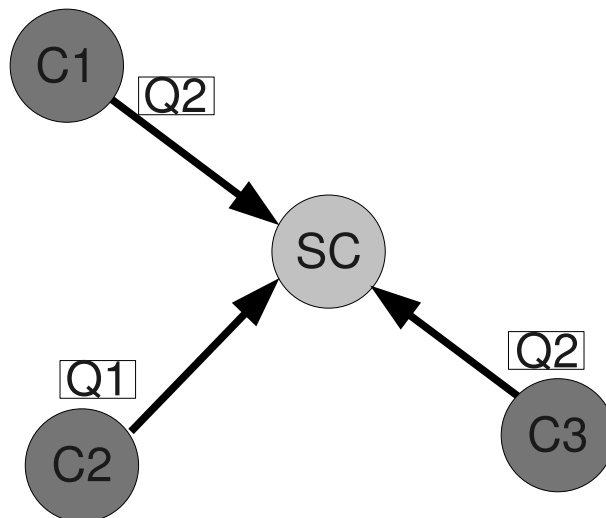


Figura 2.1: Sistema Cliente-Servidor

A Figura 2.2 ilustra um possível uso de uma arquitetura de RDC para o cenário proposto na Figura 2.1. Na Figura 2.2 são mostrados os mesmos clientes, com as respectivas requisições da Figura 2.1; porém, o papel do servidor central é desempenhado por

uma RDC de dois servidores ($S1$ e $S2$). Estes servidores abrigam réplicas dos conteúdos, representadas pela letra R seguidas por um número que representa qual conteúdo está replicado. Assim, a sequência $R1$ representa uma réplica do conteúdo 1.

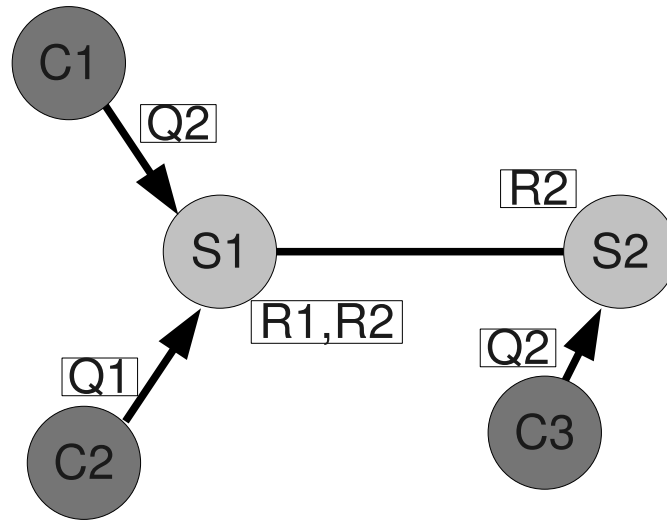


Figura 2.2: Sistema Cliente-Servidor com RDC

Capítulo 3

O Problema de Posicionamento de Réplias e Distribuição de Requisições

As equações devem ser inseridas como no exemplo a seguir.

O PLFC pode ser descrito pela seguinte formulação matemática:

Variáveis binárias:

- $y_j = \begin{cases} 1, & \text{se a facilidade } j \text{ é utilizada} \\ 0, & \text{caso contrário} \end{cases}$
- $x_{ij} = \begin{cases} 1, & \text{se a demanda do cliente } i \text{ é atendida pela facilidade } j \\ 0, & \text{caso contrário} \end{cases}$

Constantes:

- F conjunto de possíveis localizações de facilidades.
- C conjunto de clientes.
- d_i demanda do cliente i .
- s_j capacidade da facilidade j .
- f_j custo para abrir a facilidade j .
- c_{ij} custo de atendimento do cliente i pela facilidade j

As seguintes equações podem descrever o PLFC:

$$\text{Min} \quad \sum_{i \in C} \sum_{j \in F} c_{ij} x_{ij} + \sum_{j \in F} f_j y_j \quad (3.1)$$

S.a.

$$\sum_{j \in F} x_{ij} = 1, \quad \forall i \in C, \quad (3.2)$$

$$\sum_{i \in C} d_i x_{ij} \leq s_j y_j, \quad \forall j \in F, \quad (3.3)$$

$$x_{ij} \leq y_j, \quad \forall i \in C, \forall j \in F, \quad (3.4)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in C, \forall j \in F, \quad (3.5)$$

$$y_j \in \{0, 1\}, \quad \forall j \in F. \quad (3.6)$$

A função objetivo (3.1) minimiza os custos de atendimento e abertura de facilidades. As restrições (3.2) afirmam que todos os clientes devem ser atendidos. A capacidade de cada facilidade utilizada é assegurada pelas restrições (3.3). As restrições (3.4) impedem que clientes sejam atendidos por facilidades que não estejam abertas e as restrições (3.5) e (3.6) são as restrições de integralidade e não negatividade.

Capítulo 4

Trabalhos Relacionados

As tabelas devem ser inseridas como no exemplo a seguir.

A Tabela 4 mostra alguns dos trabalhos considerados, pelos autores, mais sofisticados em termos de abordagens. Em comum todos estes trabalhos tem o fato de utilizar mais de um tipo de metodologia para resolver o problema trabalhado. Outro ponto em comum entre estes trabalhos é que todos eles tratam de múltiplos conteúdos simultaneamente. A segunda linha da tabela indica o(s) problema(s) tratados em cada um dos trabalhos. Note que diversos problemas podem ser trabalhados dentro do contexto de uma RDC. A terceira linha indica a versão do problema trabalhado. A quarta linha apresenta, para cada um dos trabalhos, os pontos onde o problema se mostra dinâmico. Note que na maioria dos trabalhos apenas as requisições são dinâmicas. A quinta linha mostra um diferencial deste trabalho que é o fato de cada requisição poder ser diferente das outras. Todos os trabalhos listados tratam requisições por um mesmo conteúdo originarias de um mesmo servidor como sendo requisições indistinguíveis. Neste trabalho, como cada uma das requisições possui atributos de QoS diferentes, elas não podem ser tratadas de maneira homogênea. A sexta linha mostra a existência (ou ausência) de conteúdos fixos. Nos trabalhos onde existem conteúdos fixos, não existe a preocupação de remover todas as réplicas de um conteúdo, visto que todos os conteúdos possuem uma réplica permanente em algum local da RDC. A existência de cenários assimétricos nas análises dos trabalhos é apresentada na sétima linha. As linhas oito e nove mostram como são as capacidades do servidores. A décima linha mostra como a QoS é tratada nos diversos trabalhos. Note que entre os trabalhos da tabela apenas dois se preocupam com restrições de QoS. A décima primeira linha mostra outro diferencial deste trabalho que é o atendimento simultâneo por múltiplos servidores. A décima segunda linha mostra quais os trabalhos que possuem preocupação com o balanceamento de carga nos servidores. A penúltima linha mostra os tipos de custo

analisados em cada um dos trabalhos. Estes custos estão representados com expressões de letras que possuem o seguinte significado: A letra “A” representa a disponibilidade dos conteúdos no sistema; A letra “C” representa o custo por uso de servidores; O custo por atraso nas requisições é representado pela letra “D”; A letra “H” representa o custo de antedimento das requisições; A letra “L” representa o custo por desbalanceamento do sistema; O custo de replicação dos conteúdos é representado pela letra “R” e o custo de atualização dos conteúdos é representado pela letra “U”. As metodologias de cada trabalho estão expostas na última linha da tabela onde a letra “E” indica o uso de uma metodologia exata, a letra “H” indica o uso de uma metodologia heurística, a letra “Y” indica o uso de metodologias Híbridas e a letra “M” indica o uso de meta-heurísticas.

| | | | | | | |
|----|-----------------------------|-------------|--------------|--------------|-------------|-----------------|
| 1 | Características | [3] | [2] | [6] | [1] | Este Trabalho |
| 2 | Problema Tratado | PPS+PPRDR' | PPR | PR+PPR | PPRDR' | PPRDR |
| 3 | Versão do Problema | Estático | Din. Offline | Din. Offline | Din. Online | Din. Online |
| 4 | Dinamismo | - | Req. | Req. | Req. | Req.+Cont.+Rede |
| 5 | Requisições Heterogêneas | Não | Não | Não | Não | Sim |
| 6 | Conteúdos fixos | Sim | Não | Não | Sim | Não |
| 7 | Cenários Assimétricos | Não | Não | Não | Não | Sim |
| 8 | Disco nos Servidores | Heterogêneo | Homogêneo | Homogêneo | Homogêneo | Heterogêneo |
| 9 | Banda nos Servidores | Infinita | Infinita | Homogênea | Homogênea | Heterogênea |
| 10 | QoS | - | Dist. Max. | - | - | Atraso /Banda. |
| 11 | Atend. Múltiplos Servidores | Não | Não | Não | Não | Sim |
| 12 | Balanceamento de Carga | Não | Não | Sim | Sim | Não |
| 13 | Custos Analisados | C+H | H+U | A+L | H+R+U | H+D+R |
| 14 | Metodologia | E+H | E+H | H+M | E+H | E+Y+H+M |

Tabela 4.1: Trabalhos Relacionados

Capítulo 5

Abordagens Exatas

Exemplos de como construir sub-capítulos dentro de um capítulo são mostrados aqui.

5.1 Formulação para o PPRDR Estático

Também apresentamos aqui outros exemplos de como inserir equações no texto.

Definimos as seguintes variáveis e notações

Variáveis binárias:

- $y_{jk} = \begin{cases} 1, & \text{se o conteúdo } k \text{ está replicado no servidor } j \\ 0, & \text{caso contrário} \end{cases}$
- $x_{ij} = \begin{cases} 1, & \text{se a requisição } i \text{ é atendida pelo servidor } j \\ 0, & \text{caso contrário} \end{cases}$

Constantes:

- R conjunto de requisições a serem atendidas.
- S conjunto de servidores da RDC.
- C conjunto de conteúdos replicados.
- δ' tamanho do horizonte (em segundos).
- L_k o tamanho do conteúdo k (em bytes).
- AS_j espaço em disco disponível no servidor j (em bytes).

- MB_j banda máxima do servidor j (em bytes/segundo).
- BR_i exigência de banda da requisição i (em bytes/segundo).
- NR_k número de réplicas disponíveis para o conteúdo k .
- $G(i)$ o conteúdo exigido pela requisição i .
- c_{ij} custo de atendimento da requisição i no servidor j , calculado pela seguinte equação $c_{ij} = (RTT_{j,origem(i)} + atraso_{j,origem(i)} + ld(i)) \times BR_i$.

Assim, a seguinte formulação é proposta para a versão estática do PPRDR.

$$\text{Min} \quad \sum_{i \in R} \sum_{j \in S} c_{ij} x_{ij} + \sum_{j \in S} \sum_{k \in C} L_k y_{jk} \quad (5.1)$$

S.a.

$$\sum_{j \in S} x_{ij} = 1, \forall i \in R, \quad (5.2)$$

$$\sum_{i \in R} \delta' BR_i x_{ij} \leq \delta' MB_j, \forall j \in S, \quad (5.3)$$

$$x_{ij} \leq y_j G(i), \forall i \in R, \forall j \in S, \quad (5.4)$$

$$\sum_{j \in S} y_{jk} \leq NR_k, \forall k \in C, \quad (5.5)$$

$$\sum_{k \in C} L_k y_{jk} \leq AS_j, \forall j \in S, \quad (5.6)$$

$$x_{ij} \in \{0, 1\}, \forall i \in R, \forall j \in S, \quad (5.7)$$

$$y_{jk} \in \{0, 1\}, \forall j \in S, \forall k \in C. \quad (5.8)$$

5.2 Formulação para o PPRDR Dinâmico e com Demanda Divisível

Outro exemplo de sub-capítulo.

5.3 Formulação para Atendimento Maximizado

Mais um exemplo de sub-capítulo e de equações:

$$x_{it} - b_{i(t-1)} + b_{it} = D_{it} \quad (5.9)$$

5.4 Formulação com Múltiplas Origens de Conteúdos

Mais um exemplo de sub-capítulo.

5.5 Formulação com Submissão e Remoção de Conteúdos

Mais um exemplo de sub-capítulo.

5.6 Um Algoritmo Enumerativo para o PPRDR

Neste sub-capítulo apresentamos como inserir algoritmos. O exemplo de um modelo de algoritmo segue abaixo.

A árvore que descreve o espaço de soluções usado pelo *SEA* pode ser obtida através dos seguintes passos:

- 1) Criar tuplas $\langle t, s, c \rangle$ onde $t \in T$, $s \in S$ e c pertence ao conjunto de conteúdos ativos em t (Conteúdos que possuem t em seu tempo de vida). Armazenar as tuplas em uma lista l . Estas tuplas serão os nós da árvore.
- 2) Ordene l em ordem ascendente de i) períodos, ii) servidores e iii) conteúdos.
- 3) Suponha que a notação $x : xs$ representa uma lista com nó cabeça x e cauda xs . Seja $f(\text{ramo}, x : xs)$ uma função recursiva que recebe um ramo e uma lista de nós, remove o elemento na cabeça da lista (x), liga este elemento em ramo criando dois sub-ramos, chamados de ramoS e ramoN , que representam, respectivamente, a inserção do conteúdo $x.c$ no servidor $x.s$ no período $x.t$ e a **não** inserção de $x.c$ em $x.s$ no período $x.t$. A função f então se invoca recursivamente para ambos os ramos, usando a lista xs como parâmetro ($f(\text{ramoS}, xs)$ and $f(\text{ramoN}, xs)$). A função f retorna quando a lista recebida como parâmetro é vazia.

O Algoritmo 1 apresenta uma possível representação para o *SEA*. O algoritmo procede da seguinte maneira: Primeiramente, uma solução inicial é gerada por uma heurística ou outro método qualquer para ser usada como primeira solução incumbente. A seguir, a lista de tuplas contendo as variáveis de decisão é criada e ordenada convenientemente. O passo mais importante do algoritmo é executado pela função *ExplorarRamo* que de fato executa o processo de enumeração e poda. A melhor solução encontrada pelo método *ExplorarRamo* é retornada como melhor solução encontrada pelo *SEA*, sendo que, caso o processo de enumeração seja concluído, esta melhor solução é uma solução ótima.

Algoritmo 1 SEA()

- 1: Solução melhor = gerarSoluçãoInicial()
 - 2: Construir a lista de tuplas l composta por $\langle t, s, c \rangle, \forall t \in T, \forall s \in S$ e $\forall c \in \text{conteúdosAtivos}(t)$
 - 3: Ordenar l por i) período, ii) servidor e iii) conteúdo
 - 4: Solução atual = novaSolução()
 - 5: ExplorarRamo(atual, melhor, l)
 - 6: Retorne melhor
-

O método *ExplorarRamo* é descrito pelo Algoritmo 2 que segue na seguinte página.

Algoritmo 2 ExplorarRamo(Solução *sol*, Solução *melhor* , lista de tuplas *l*)

```

1: se Existe alguma violação da restrição de disco em sol então
2:   Poda por violação de espaço em disco
3:   Retorne
4: fim se
5: se Não há espaço para acomodar todos os conteúdos do período corrente em sol então
6:   Poda por falta de espaço em disco
7:   Retorne
8: fim se
9: se Todas as decisões de um período t foram tomadas então
10:  se Existe algum conteúdo ausente no período t em sol então
11:    Poda por conteúdo ausente
12:    Retorne
13:  fim se
14:  Seja c1 o valor da função objetivo do PDR correspondente ao período t em sol
15:  Seja c2 a soma dos valores das funções objetivos de todos os períodos anteriores a
    t em sol
16:  Seja c3 a soma dos custos mínimos de atendimento para todos os períodos posteri-
    ores a t
17:  se  $c1 + c2 + c3$  é pior que a função objetivo de melhor então
18:    Poda por custo mínimo adiante
19:    Retorne
20:  fim se
21: fim se
22: se l está vazia então
23:   Todas as decisões foram tomadas
24:   se sol é melhor que melhor então
25:     melhor = sol
26:     Retorne
27:   fim se
28: fim se
29: Tupla q = removerCabeça(l)
30: Alocar(sol , q.t, q.s, q.c, sim)
31: ExplorarRamo(sol, melhor, l)
32: Alocar(sol , q.t, q.s, q.c, não)
33: ExplorarRamo(sol, melhor, l)
34: inserirCabeça(q, l)
35: Retorne

```

Capítulo 6

Abordagens Heurísticas

Aqui segue apenas a estrutura deste capítulo com um exemplo de algoritmo.

6.1 *Framework* Unificador

A seguir outro algoritmo.

Algoritmo 3 *framework* para resolução do PPRDR()

- 1: **para todo** período de tempo **faça**
 - 2: Resolver o PDR
 - 3: Estimar demanda (Não Obrigatório)
 - 4: Resolver PPR
 - 5: **fim para**
-

6.2 Abordagens para o PDR

A seguir uma nova sub-divisão do capítulo.

6.2.1 Formulação Matemática para o PDR

6.2.2 Atendimento Local

6.3 Técnicas para Estimar a Demanda Futura

6.3.1 Estimador Baseado em Médias

6.3.2 Estimadores Baseado em Alisamento Exponencial

Capítulo 7

Instâncias de Testes

Apenas colocamos a estrutura do capítulo.

Capítulo 8

Resultados Computacionais

Além da estrutura deste capítulo apresentamos aqui um exemplo de tabela e de figura colorida.

8.1 Resultados das Abordagens Exatas

| Instancia | Tempo (s) | F.O. | Método |
|-----------|-----------|-----------|--------|
| 10A01 | 1.21 | 3.00E+005 | FD |
| 10A02 | 1.04 | 2.72E+005 | FD |
| 10A03 | 1.15 | 2.53E+005 | FD |
| 10A04 | 1.28 | 3.49E+005 | FD |
| 10A05 | 0.84 | 2.65E+005 | FD |
| 20A01 | 3.47 | 2.00E+006 | FD |
| 20A02 | 8.93 | 2.14E+006 | FD |
| 20A03 | 7.50 | 2.07E+006 | FD |
| 20A04 | 6.76 | 2.08E+006 | FD |
| 20A05 | 6.23 | 2.10E+006 | FD |
| 30A01 | 20.77 | 2.10E+006 | FD |
| 30A02 | 14.86 | 2.16E+006 | FD |
| 30A03 | 15.25 | 2.15E+006 | FD |
| 30A04 | 17.51 | 2.22E+006 | FD |
| 30A05 | 19.12 | 2.24E+006 | FD |
| 50A01 | 59.98 | 4.63E+005 | FD |
| 50A02 | 63.00 | 8.21E+005 | FD |
| 50A03 | 62.16 | 8.78E+005 | FD |
| 50A04 | 60.19 | 6.49E+005 | FD |
| 50A05 | 58.57 | 7.54E+005 | FD |

Tabela 8.1: Melhores Resultados Exatos: Classe A

8.2 Síntese dos Resultados para a Versão *Online* do PPRDR

A Figura colorida.

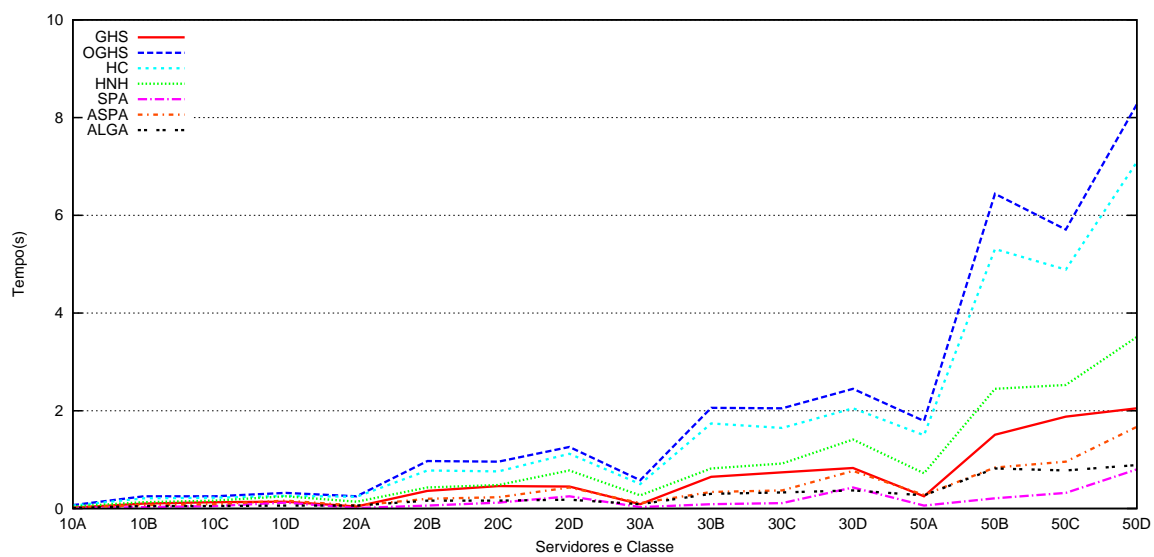


Figura 8.1: Tempos computacionais

Capítulo 9

Conclusões e Trabalhos Futuros

Aqui devem ser apresentadas as conclusões e os trabalhos que permanecem pendentes para o futuro.

9.1 Conclusões

9.2 Trabalhos Futuros

A referência (Holland, 1975) corresponde ao número [4] é um livro. No item referência é mostrado como citar livros.

Referências

- [1] AIOFFI, W., MATEUS, G., ALMEIDA, J., LOUREIRO, A. Dynamic content distribution for mobile enterprise networks. *IEEE Journal on Selected Areas in Communications* 23, 10 (2005), 2022–2031.
- [2] BARTOLINI, N., PRESTI, F., PETRIOLI, C. Optimal dynamic replica placement in content delivery networks. Em *Proceedings of The 11th IEEE International Conference on Networks 2003. ICON2003* (2003), p. 125–130.
- [3] BEKTAS, T., OGUZ, O., OUVEYSI, I. Designing cost-effective content distribution networks. *Computers and Operations Research* 34 (2007), 2436–2449.
- [4] HOLLAND, J. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, 1975.
- [5] TENZAKHTI, F., DAY, K., OULD-KHAOUA, M. Replication algorithms for the world-wide web. *Journal of Systems Architecture* 50 (2004), 591–605.
- [6] ZHOU, X., XU, C.-Z. Efficient algorithms of video replication and placement on a cluster of streaming servers. *Journal of Network and Computer Applications* 30 (2007), 515–540.