

POLLYANNA GRAZIELLE FARIA DIAS

**PROJETO DE REDE COM CUSTOS CONVEXOS  
E BALANCEAMENTO DE FLUXOS**

Belo Horizonte  
11 de março de 2011

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
ESCOLA DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**PROJETO DE REDE COM CUSTOS CONVEXOS  
E BALANCEAMENTO DE FLUXOS**

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Engenharia Elétrica.

POLLYANNA GRAZIELLE FARIA DIAS

Belo Horizonte  
11 de março de 2011



UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Projeto de Rede com Custos Convexos e Balanceamento de Fluxos

POLLYANNA GRAZIELLE FARIA DIAS

Dissertação defendida e aprovada pela banca examinadora constituída por:

Ph. D. RODNEY REZENDE SALDANHA – Orientador  
DEE/UFMG

Ph. D. GILBERTO DE MIRANDA JÚNIOR – Co-orientador  
DEP/UFMG

Ph. D. RICARDO SARAIVA DE CAMARGO  
DEP/UFMG

Ph. D. RICARDO HIROSHI CALDEIRA TAKAHASHI  
DMAT/UFMG

Ph. D. ADRIANO CHAVES LISBOA  
DEE/UFMG

Belo Horizonte, 11 de março de 2011

# Resumo

Problemas de desenho de rede são o tema de um grande número de contextos incluindo sistemas de transportes, de telecomunicações e sistemas de energia. A idéia é estabelecer uma rede de conexões (estradas, canais de fibra óptica, linhas elétricas, etc.) que possibilitam o fluxo de produtos (pessoas, pacotes de dados, eletricidade, etc.) a fim de satisfazer algumas características de demanda. O interesse deste trabalho é o problema de desenho de rede de custo fixo, onde, a fim de utilizar uma conexão, deve-se pagar um custo fixo representando por exemplo, o custo de construção da rodovia, da instalação de uma linha de eletricidade, etc. Definida a relevância de tais problemas, foi desenvolvido um modelo de programação inteira mista não linear, buscando expandir as fronteiras das soluções do mesmo, utilizando formulações de fluxos e técnicas de decomposição. O objetivo é decompor a rede em vários subproblemas enraizados em um terminal (nó) e resolvê-los um a um. A estratégia de solução para cada subproblema é encontrar uma solução inicial factível e aprimorá-la utilizando técnicas de decomposição conhecidas como decomposição de Benders e Aproximação Externa. Ao combinar as duas técnicas, encontra-se rapidamente a solução ótima. Isso acontece devido à grande redução do efeito cauda longa (tail-off), quando se compara com a decomposição de Benders. A utilidade do método também é função da qualidade dos cortes de Benders que podem melhorar a relaxação de programação linear do problema mestre.

**Palavras-chaves:** Projetos de Rede de Custo Fixo; Método de Decomposição de Benders; Aproximação Externa.

# Abstract

Network design problems are central to a large number of contexts including transportation, telecommunications and power systems. The idea is to establish a network of links (roads, optical fibers, electric lines, etc.) that enables the flow of commodities (people, data packets, electricity, etc.) in order to satisfy some demand characteristics. The interest of this work is *fixed-charge network design problems*, where, in order to use a link, one must pay a fixed cost that represents for example, the cost of constructing a road, or installing an electric line. Defined the relevance of such problems, it was developed a model of nonlinear mixed integer programming, seeking a breakthrough using flow formulations and techniques of decomposition. The strategy for each sub-problem is find an initial feasible solution and improve it by decomposition techniques like Benders Decomposition and Outer Approximation. The aim is to use both techniques together, where the goal is to quickly find the optimal solution. This is due to the great reduction of the tail-off effect when comparing to the classical Generalized Benders Decomposition. The utility of the method is also a function of the quality of the Benders cuts that can improve the linear programming relaxation of the master programs.

**Keywords:** Charged-Fixed Network Design Problems, Generalized Benders Decomposition, Outer Approximation Technique.

Aos meus pais, Wander e Elizabeth  
Ao meu irmão Bruno  
Amo vocês!!!

*"É muito melhor arriscar coisas grandiosas, alcançar triunfos e glórias, mesmo expondo-se a derrota, do que formar fila com os pobres de espírito que nem gozam muito nem sofrem muito, porque vivem nessa penumbra cinzenta que não conhece vitória nem derrota".*

Theodore Roosevelt

# Agradecimentos

À Deus, sem Ele nada seria possível.

Aos meus pais, meu muitíssimo e eterno obrigada pelo apoio, incentivo e carinho durante toda minha vida. Amo vocês!

Ao meu irmão, Bruno, pelo constante apoio e carinho.

Ao Prof. Rodney Saldanha, meus sinceros agradecimentos, pela orientação, ensinamentos, paciência e incentivos durante a realização deste trabalho.

Ao Prof. Gilberto de Miranda Jr., meu co-orientador, pela amizade, apoio e incentivo. Foram muito valiosos seus conselhos.

Aos companheiros de mestrado pelo apredizado diário.

Aos meus verdadeiros e eternos amigos.

À todos aqueles que, de alguma forma, colaboraram para a concretização deste trabalho.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Revisão da Literatura</b>	<b>4</b>
2.1	Problemas de desenho de redes não capacitadas . . . . .	5
2.1.1	Os pares origem-destino . . . . .	5
2.1.2	Desenho de rede de acesso local . . . . .	7
2.1.3	As duas tecnologias origem-única UNDP (duas tecnologias para o problema de desenho de rede de acesso local) . . . . .	10
<b>3</b>	<b>Modelagem Matemática</b>	<b>12</b>
3.1	Introdução . . . . .	12
3.2	Uma Formulação de Fluxos Multiproduto . . . . .	12
3.2.1	Variáveis e Parâmetros . . . . .	13
3.3	Modelo Não Linear Inteiro Misto . . . . .	13
<b>4</b>	<b>Técnicas de Decomposição</b>	<b>16</b>
4.1	Decomposição de Benders . . . . .	16
4.1.1	Problema Mestre . . . . .	18
4.1.2	O Problema Mestre Relaxado . . . . .	19
4.1.3	SubProblemas Primal e Dual . . . . .	22
4.1.4	SubProblema Primal . . . . .	23
4.1.5	SubProblema Dual . . . . .	24
4.2	Aproximação Externa . . . . .	25
<b>5</b>	<b>Experimentos Computacionais</b>	<b>29</b>
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>45</b>
	<b>Referências Bibliográficas</b>	<b>46</b>

# Lista de Figuras

4.1	Algoritmo para a resolução do problema . . . . .	17
5.1	Análise da Instância de 12 nós . . . . .	30
5.2	Exemplo de um caso com 18 nós . . . . .	31
5.3	Efeito Tail-Off da instância 17 sob congestionamento igual a 10% . . . . .	32
5.4	Comparação de Tempo Computacional para todas as instâncias com congestionamento igual a 10% . . . . .	32

# Lista de Tabelas

5.1	Espaço das Variáveis das 12 instâncias . . . . .	29
5.2	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento,número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 12 nós . . . . .	33
5.3	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento,número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 14 nós . . . . .	34
5.4	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento,número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 15 nós . . . . .	35
5.5	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento,número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 16 nós . . . . .	36
5.6	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento,número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 17 nós . . . . .	37
5.7	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento,número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 18 nós . . . . .	38
5.8	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento,número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 20 nós . . . . .	39
5.9	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento,número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 21 nós . . . . .	40
5.10	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento,número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 22 nós . . . . .	41

5.11	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 24 nós . . . . .	42
5.12	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 25 nós . . . . .	43
5.13	Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 27 nós . . . . .	44

# Capítulo 1

## Introdução

*"A coisa mais bela que podemos experimentar é o mistério. Essa é a fonte de toda a arte e ciências verdadeiras."*

Albert Einstein

No mercado global hiper-competitivo atual, são particularmente importantes os custos de transporte em qualquer cadeia logística que envolva a distribuição de um dado produto desde sua fabricação ou extração até o consumidor final. Tal relevância se justifica pelo fato de que os custos logísticos podem responder por até 40% dos custos totais de operação em uma cadeia produtiva, o que significa que tem impacto direto sobre a competitividade e a adaptabilidade de uma dada organização.

Torna-se necessário então tratar com cuidado os problemas típicos da otimização de tais sistemas logísticos, que podem ser tanto de transporte de produtos, como de distribuição de energia ou de telecomunicações. Na verdade, pode-se citar uma gama bastante ampla de aplicações contemporâneas, uma vez que quase todos os serviços oferecidos estão organizados sob a forma de redes: as malhas ferroviária e rodoviária, o sistema de distribuição de energia elétrica, o sistema de fornecimento de água, os serviços de telecomunicações, o sistema de segurança pública, os serviços de saúde, os postos de atendimento das várias organizações governamentais. Isso sem falar das redes de computadores: dos sistemas ponto a ponto, passando pelas pequenas redes de escritório, indo além da Internet.

Um grande número de aplicações práticas podem ser representadas por meio de modelos de redes de custo fixo. Uma importante área é problema de desenho de redes de serviços, no qual se aplicam, por exemplo, companhias áreas e companhias de transporte terrestre. A idéia é maximizar o lucro a partir da definição das rotas e horários, dado um conjunto de restrições de recurso. Por exemplo, para as companhias áreas deve-se determinar a cobertura de rotas e a frequência de vôos considerando o tráfego aéreo e a disponibilidade da tripulação. Similarmente, companhias aéreas de

entrega devem estabelecer rotas, associando o tráfego aéreo e decidir sobre o fluxo de pacotes.

Várias aplicações podem também ser encontradas no setor de telecomunicações. O desenho da rede de acesso local com uma ou duas tecnologias, o projeto de layout de terminal e a interconexão entre essas redes são exemplos.

Em todos esses problemas, um projeto de desenho de rede adequado pode produzir níveis de operações melhores e redução de custos. O montante total destas reduções é obviamente relacionada com cada problema específico. Entretanto, a importância econômica de muitos dos problemas citados e o intuito dos desenhos de rede em sistema operativos sugerem que as economias de custo podem ser significantes.

Provavelmente por causa da importância econômica associada a esses problemas, várias soluções metodológicas para problemas de desenho de rede são avaliadas. Entre as abordagens mais eficientes, destaca-se a decomposição de Benders. A idéia básica por trás do método é decompor o problema em duas partes simples: a primeira parte, chamada de *problema mestre*, resolve a versão relaxada do problema e obtém-se os valores de um subconjunto de variáveis. A segunda parte, chamada de *subproblema*, obtém os valores das variáveis remanescentes enquanto mantém as primeiras fixas, e utiliza estas para gerar cortes para o problema mestre. O problema mestre e o subproblema são resolvidos iterativamente até que cortes não possam mais ser gerados. A conjunção das variáveis encontradas na última instância do problema mestre e da iteração do subproblema é a solução da formulação original.

A estrutura de problemas de desenho de rede de custo fixo apresenta um esquema de decomposição natural para a abordagem de Benders: as variáveis apresentam a construção das conexões que são solucionados no problema mestre enquanto alguns representam os fluxos atuais de produtos que mantém o subproblema. Entretanto, a cada iteração do problema mestre, a solução encontra uma possível rede no qual o subproblema encontra o fluxo ótimo de produtos.

Devido à necessidade de solucionar o problema mestre e o subproblema diversas vezes, a abordagem da decomposição é somente razoável se os problemas possam ser resolvidos eficientemente. Este é o caso dos problemas de desenho de rede, onde na maioria das vezes, é muito mais fácil de se resolver problemas decompostos do que o problema original. Entretanto, especialmente para o subproblema, é possível algumas vezes continuar além da decomposição (por produto, por período, etc), resultando em problemas com soluções até mais eficientes.

Outra técnica de decomposição conhecida como Aproximação Externa, do inglês Outer Approximation (OA), é uma técnica simples porém muito eficiente para resolução de problemas inteiros não lineares, utilizando uma abordagem de plano cortantes. O método também possui uma técnica de coordenação entre o problema mestre e o

subproblema, trabalhando similarmente a técnica de DB, mas os problemas mestres de OA são escritos no espaço de todas as variáveis do problema, dispensando a projeção do espaço entre as variáveis inteiras. Este recurso melhora o poder dos planos cortantes associados e assegura a convergência em um número de iterações menor que os obtidos ao se aplicar a técnica de DB. Como desvantagem, a solução do problema mestre é mais cara, geralmente aumentando o tempo de solução e o esforço computacional à medida que o problema aumenta.

Este trabalho apresenta um modelo de desenho de projeto de rede adicionado o efeito de custo de congestionamento. No capítulo 2 descreve-se a revisão bibliográfica sobre o projeto de desenho de rede. O capítulo 3 descreve a modelagem matemática do problema, o capítulo 4 descreve as técnicas utilizadas para a resolução do problema, o capítulo 5 apresenta os testes computacionais e o capítulo 6, finalmente sumariza as principais conclusões e contribuições alcançadas.

# Capítulo 2

## Revisão da Literatura

*"Feliz é aquele que transfere o que sabe e aprende o que ensina."*

Cora Coralina

Problemas de Desenho de Rede lidam com a seleção de arcos de um grafo a fim de satisfazer, à mínimo custo, alguns requisitos de fluxo normalmente expressados na forma das demandas do par origem destino. O problema de desenho de rede de custo fixo (PDRCF) possui uma particularidade que cada arco é associado a um custo fixo no qual precisa ser pago caso este arco faça parte da solução.

Neste capítulo será apresentada uma revisão dos algoritmos de decomposição de Benders propostos para o PDRCF. Embora na literatura encontram-se os modelos como capacitados e não capacitados, serão de interesse deste trabalho somente os modelos não capacitados.

A idéia é apresentar diferentes versões para os problemas de desenho de rede de custo fixo não capacitado, dando a cada um, uma formulação matemática e comentários gerais das aplicações dos métodos da Decomposição de Benders para cada problema encontrado na literatura.

Em todos os desenvolvimentos desse capítulo, define-se  $G(V, A, K)$  um grafo não direcionado onde  $N$  representa o conjunto de nodos,  $A$  representa o conjunto das arestas, e  $K$  representa o conjunto de produtos a serem transportados (pessoas ou bens, pacotes de dados, energia elétrica, etc.) Variáveis  $x_{ij}^k$  representam o fluxo atual do produto  $k$  passando através do nodo  $i$  ao nodo  $j$ , e as variáveis  $y_{ij}$  são variáveis inteiras associadas com a utilização de uma conexão (caso não capacitado) ou utilizando a sua capacidade (caso capacitado).

Note que, apesar do grafo ser não direcionado, faz sentido definir ambas as variáveis  $x_{ij}^k$  e  $x_{ji}^k$ , porque o fluxo torna-se direcionado. Então, considera-se que uma aresta em  $A$  possui sentido duplo representado pelos arcos  $(i, j)$  e  $(j, i)$ . Entretanto, quando cita-

se as variáveis inteiras (ou seus coeficientes) define-se que somente a aresta  $(i, j)$  com  $i < j$  é considerada.

## 2.1 Problemas de desenho de redes não capacitadas

Nesta seção os problemas de desenho de redes não capacitadas (PDRNC) são apresentados. Nestes problemas, não existem limite nos fluxos que podem circular através dos *links* selecionados. Muitas aplicações da vida real pode sem representados por estes modelos, Radazzo et al. (2001); Gavish (1983).

### 2.1.1 Os pares origem-destino

Nos PDRNC, cada *commodity*  $k = 1, \dots, |K|$  é associada a uma demanda  $d_k$ , a um nodo origem  $O(k)$  e a um nodo de destino  $D(k)$ . Em muitos casos, os custos fixos cheios representam o real custo da rede, como por exemplo, não existe um volume de custo associado com o fluxo de um produto em um conexão. Entretanto a fim de apropriar alguns problemas, muitas vezes deve-se incluir custos baseados em volume, como no trabalho de Radazzo et al. (2001).

Usando a notação introduzida no início deste capítulo, apresenta-se abaixo uma formulação da classe destes problemas:

(F1) Formulação pares Origem-Destino UNDP:

$$\text{Minimize} \quad \sum_{(i,j) \in A} \left( \sum_{k \in K} c_{ij}^k x_{ij}^k + f_{ij} y_{ij} \right) \quad (2.1)$$

sujeito à:

$$\sum_{j|(i,j) \in A} x_{ij}^k - \sum_{j|(j,i) \in A} x_{ji}^k = w \quad (2.2)$$

$$x_{ij}^k \leq d_k y_{ij}, \quad \forall (i, j) \in A, \forall k \in K \quad (2.3)$$

$$x_{ij}^k \geq 0, \quad \forall (i, j) \in A, \forall k \in K \quad (2.4)$$

$$y_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A. \quad (2.5)$$

$$\text{onde } w = \begin{cases} d_k, & i = O(k), \forall k \in K, \\ 0, & i \notin \{O(k), D(k)\}, \forall k \in K, \\ -d_k, & i = D(k), \forall k \in K, \end{cases}$$

Variáveis  $y_{ij}$  são associadas com a construção da conexão  $(i, j) \in A : y_{ij} = 1$  se  $(i, j)$  pertence a solução final, caso contrário  $y_{ij} = 0$ . A variável  $x_{ij}^k$  representa o fluxo do

produto  $k$  na conexão  $i, j$ . A função objetivo 2.1 é a soma dos custos fixos e variáveis. Nesta função,  $c_{ij}^k$  é o custo linear associado ao fluxo de uma unidade de produto  $k$  através da conexão  $(i, j)$  e  $f_{ij}$  é o custo fixo associado a seleção da conexão  $(i, j)$  na solução final. Algumas versões não lineares dessa função podem ser consideradas pela *Decomposição de Benders*. (veja por exemplo, o trabalho de [Hoang \(1982\)](#)). Entretanto, por simplicidade, concentra-se nas funções de objetivo linear.

O conjunto de restrições 2.2 garante que o fluxo do produto  $k$  sai do nodo de origem  $O(k)$  e chega até o nodo de destino  $D(k)$ , e também é responsável pela conservação do fluxo. As restrições 2.3 limitam o fluxo dos produtos nos arcos selecionados, as restrições 2.4 definem a não negatividade das variáveis  $x_{ij}^k$ , e as restrições 2.5 impõem a integralidade das variáveis  $y_{ij}$ . As restrições 2.3 podem ser substituídas pelas restrições de agregação, na forma

$$\sum_{k \in K} x_{ij}^k \leq \left( \sum_{k \in K} d_k \right) y_{ij}, \quad \forall (i, j) \in A \quad (2.6)$$

Esta restrição de agregação pode reduzir substancialmente o número de restrições. Entretanto, foi provado que os cortes de Benders obtidos com desagregação (cortes fortes), as formulações são mais eficientes do que as obtidas com as restrições de agregação, [Magnanti et al. \(1986\)](#).

*Aplicação da Decomposição de Benders:* O caso básico do modelo não capacitado foi resolvido via decomposição de Benders em um trabalho de [Magnanti et al. \(1986\)](#). No trabalho, o problema mestre propõe uma rede experimental fixando as variáveis inteiras e o subproblema encontra a distribuição do fluxo das variáveis contínuas. Os autores utilizam o conjunto de instâncias acima de 30 nodos e 130 arcos para provar dois pontos:

- (i) o uso de técnicas auxiliares pode melhorar a performance da decomposição de Benders (neste caso, o pré-processamento baseado em um procedimento *dual-ascent* a fim de eliminar variáveis);
- (ii) a escolha inteligente dos cortes, especificamente dos cortes de Pareto-ótimo podem formentemente afetar a performance do algoritmo.

O trabalho de [Magnanti et al. \(1986\)](#) foi estendido dez anos mais tarde por [Gutierrez et al. \(1996\)](#) que propôs uma abordagem robusta capaz de considerar a incerteza dos custos de transporte  $c_{ij}^k$  e as demandas  $d_k$ . A incerteza dos dados são descritos entranto por um conjuntos de cenários, cada um com diferentes valores de  $c_{ij}^k$ . Note que, desde que os autores solucionaram o caso não capacitado, as mudanças em  $c_{ij}^k$

podem refletir em ambos os custos quanto na demanda (com a escala apropriada). A solução do modelo é feita via um algoritmo de Benders multi-mestre, onde um problema mestre individual é associado com cada possível cenário. Cada vez em que o problema mestre é resolvido, cria-se uma geração de cortes de Benders cruzada, por exemplo, os subproblemas geram um corte para cada um dos problemas mestre. Os autores usam um conjunto de instâncias baseadas naquela apresentada por [Magnanti e Wong \(1981\)](#) com 11 cenários para cada problema original e concluíram que a geração cruzada dos cortes de Benders aceleram a convergência do algoritmo.

Uma versão interessante do problema de desenho de rede não capacitado foi proposto por [Hoang \(1982\)](#). O autor desenvolveu uma formulação para este problema com custos não lineares. O objetivo é selecionar entre vários projetos (cada um correspondendo a um conjunto de investimentos em rede) respeitando algumas restrições de orçamento. Neste caso, não existe explicitamente nenhum limite superior nas capacidades dos arcos como em [2.3](#): eles são indiretamente descritos pela convexidade nos custos das funções. A Decomposição de Benders (DB) é utilizada, juntamente com uma heurística Lagrangeana para resolver o problema mestre, e um algoritmo de custo convexo de multiproduto é aplicado a solução do subproblema. Duas instâncias são resolvidas com sucesso, a maior contendo 155 nodos, 376 arcos e 720 produtos.

### 2.1.2 Desenho de rede de acesso local

Um importante e especial caso deste problema é apresentado na seção [2.1.1](#) onde todos os pontos de demanda precisam ser conectados a um nodo *single*. Esta é uma situação, por exemplo do desenho de redes de acesso local (LAN). Enquanto a função objetivo é bem representada somente pelas variáveis de custo, o desenho da LAN pode ser formulado como um problema de *single source transshipment problem*. Por outro lado, se existem somente custos fixos, torna-se um problema de Steiner NP-Difícil, ou um *minimum spanning tree problem* (se todos os nodos precisam ser alcançados). Em um último caso, uma situação importante é onde o número de *links* conectados direta ou indiretamente com a fonte é limitado. Este problema é conhecido *capacitated minimum spanning tree problem*.

Uma formulação matemática é definida para o caso geral onde cada conexão possui um custo fixo e variável associado. Em F2 considera-se que cada demanda como um produto diferente, no qual diferentes custos de transporte  $c_{ij}^k$  são dependentes ambos da conexão e do produto. O nodo  $O$  é a origem de todas os produtos.

## (F2) Formulação Single-Origin UNDP

$$\text{Minimize} \quad \sum_{(i,j) \in A} \left( \sum_{k \in K} c_{ij}^k x_{ij}^k + f_{ij} y_{ij} \right) \quad (2.7)$$

sujeito à:

$$\sum_{j|(i,j) \in A} x_{ij}^k - \sum_{j|(j,i) \in A} x_{ji}^k = w \quad (2.8)$$

$$x_{ij}^k \leq d_k y_{ij}, \quad \forall (i,j) \in A, \forall k \in K \quad (2.9)$$

$$x_{ij}^k \geq 0, \quad \forall (i,j) \in A, \forall k \in K \quad (2.10)$$

$$y_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A. \quad (2.11)$$

$$\text{onde } w = \begin{cases} d_k, & i = O, \forall k \in K, \\ 0, & i \notin \{O, D(k)\}, \forall k \in K, \\ -d_k, & i = D(k), \forall k \in K, \end{cases}$$

Apesar do fato que todos os produtos são agora originadas em um nodo *single-origin*, a formulação F2 é idêntica a formulação F1. Entretanto, a presença do nodo *single-origin* produz uma formulação simplificada para o caso onde as variáveis de custo são independentes do produto. F'2 apresenta uma formulação com um origem única simplificada e apenas um produto.

(F'2) Formulação *single-origin, single-commodity* UNDP:

$$\text{Minimize} \quad \sum_{(i,j) \in A} (c_{ij} x_{ij} + f_{ij} y_{ij}) \quad (2.12)$$

sujeito à:

$$\sum_{j|(i,j) \in A} x_{ij} - \sum_{j|(j,i) \in A} x_{ji} = w \quad (2.13)$$

$$x_{ij} \leq D y_{ij}, \quad \forall (i,j) \in A, \quad (2.14)$$

$$x_{ij} \geq 0, \quad \forall (i,j) \in A, \quad (2.15)$$

$$y_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A. \quad (2.16)$$

$$\text{onde } w = \begin{cases} D, & i = O, \\ -d_i, & i \neq O, \end{cases}$$

O parâmetro  $d_i$  representa a demanda em cada nodo e  $D$  é a soma das demandas em todos os nodos exceto ao nodo de origem. A função objetivo 2.12 da formulação F'2 ainda minimiza a soma dos custos fixos e variáveis. Entretanto, não há necessidade de diferenciação entre as *commodities*. O único requisito é que a quantidade correta

de fluxos deixa o nodo de origem e alcança cada um dos nodos de demanda, onde é assegurado pelas restrições 2.13. As restrições 2.14 ainda proibem fluxo nas conexões fechadas.

Como apontado por Magnanti e Wong (1981), escolher a formulação apropriada é um fator importante que afeta a performance computacional da decomposição de Benders. Neste caso, nota-se que desde a restrição 2.14 da formulação F'2 utilizada soma de todas as demandas  $D$ , uma equivalência as restrições agregadas apresentadas em 2.6. Entretanto, deve-se esperar que a formulação seja mais fraca do que a formulação F2, no qual utiliza a restrições desagregadas como expressadas em 2.9.

*Aplicação da Decomposição de Benders:* Randazzo e Luna (2001) apresentaram uma comparação entre a relaxação Lagrangeana baseada em branch-and-bound, branch and cut e os métodos de decomposição de Benders para o problema de desenho de LAN apresentadas nesta seção. Antes de utilizar a decomposição de Benders, os autores usaram a relaxação linear e o algoritmo de menor caminho para obter a solução viável, que são utilizados para gerar os valores iniciais para as variáveis duais. Restrições redundantes fortes são usadas no problema mestre, para aumentar a probabilidade da geração da solução viável. O subproblema é decomposto em  $k$  problemas de fluxo de redes fáceis, um para cada produto. A decomposição de Benders apresentou dois outros métodos em três conjuntos pequenos de instâncias. Talvez a mais importante conclusão obtida pelos autores é o fato da decomposição de Benders, apesar de ter sido mais devagar que os cortes de branch-and-cut em seis das 30 instâncias, foi o único algoritmo capaz de resolver para a otimalidade todas as instâncias com o tempo limite (24 h), conduzindo a idéia que este é o método mais robusto.

Em outro artigo, Gavish (1982) apresentou o algoritmo de Decomposição de Benders para o *capacited minimal spanning tree problem*. O algoritmo é bastante padrão: o problema mestre fixa as variáveis inteiras produzindo duas situações:

- (i) as arestas selecionadas formam uma árvore expandida (spanning tree): então, a solução é ótima (se o número de conexões conectadas respeitam o limite) ou se as conexões conectadas a fonte podem ser usadas para gerar um corte e resolver o problema mestre novamente;
- (ii) O grafo é não conexo, no qual caso o corte possa ser gerado e o problema mestre é resolvido novamente. O algoritmo foi testado em redes contendo mais de 12 nós. O número de cortes gerados foi muito grande, resultando em uma performance fraca do algoritmo.

### 2.1.3 As duas tecnologias origem-única UNDP (duas tecnologias para o problema de desenho de rede de acesso local)

A terceira situação mais comum em problemas de desenho de rede não capacitados é o caso onde duas tecnologias, cada uma tendo suas próprias vantagens, estão disponíveis. Cada tecnologia pode ter um custo fixo muito grande mas uma variável de custo muito pequena e vice-versa (como acontece com as fibras ópticas e cobre em redes de telecomunicações).

A formulação é uma extensão da formulação F2 com a duplicação de variáveis  $x_{ij}^k$  e  $y_{ij}$ . Entretanto, em F3,  $y_{ijt}$  é uma variável binária associada com a construção do *link* com a tecnologia  $t$ .  $x_{ijt}^k$  representa o fluxo do produto  $k$  através da conexão com a tecnologia  $t$ .

(F3) As duas tecnologias origem-única UNDP

$$\text{Minimize} \quad \sum_{(i,j) \in A} \sum_{t=1}^2 \left( \sum_{k \in K} c_{ijt}^k x_{ijt}^k + f_{ijt} y_{ijt} \right) \quad (2.17)$$

sujeito à:

$$\sum_{t=1}^2 \left( \sum_{j|(i,j) \in A} x_{ijt}^k - \sum_{j|(j,i) \in A} x_{jit}^k \right) = w \quad (2.18)$$

$$x_{ijt}^k \leq d_k y_{ijt}, \quad \forall (i,j) \in A, \forall k \in K, t = 1, 2 \quad (2.19)$$

$$\sum_{t=1}^2 y_{ijt} \leq 1, \quad \forall (i,j) \in A \quad (2.20)$$

$$x_{ij}^k \geq 0, \quad \forall (i,j) \in A, \forall k \in K \quad (2.21)$$

$$y_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A. \quad (2.22)$$

$$\text{onde } w = \begin{cases} d_k, & i = O, \forall k \in K, \\ 0, & i \notin \{O, D(k)\}, \forall k \in K, \\ -d_k, & i = D(k), \forall k \in K, \end{cases}$$

Em F3, o objetivo 2.17 é minimizar a soma das variáveis e os custos fixos de ambas tecnologias. Isto deve ser feito respeitando os requisitos de demanda 2.18 e o fato que, para ambas tecnologias, onde o fluxo não pode ser transportado pela conexão se a conexão é instalada 2.19. Cada conexão utiliza somente uma tecnologia, como expressado pela restrição 2.20.

Observa-se que F3 é a formulação mais completa das quatro formulações. De fato, F1, F2, F'2 podem todas ser consideradas como casos especiais de F3. Além disso, F3

se adapta facilmente ao caso onde mais de uma tecnologia estar disponível (apesar do caso com duas tecnologias ser mais comum na prática.)

Extensões simples de F3 incluem modelos que requerem um custo para conectar conexões com diferentes tecnologias. Estes modelos são usuais para situações comuns onde equipamento extra é preciso para representar a conexão. Outros modelos podem ser usados para limitar o número destas conexões ou restringir os nodos onde foram criados.

Restrições adicionais podem formar alguns nodos primários podem ser servidos pela tecnologia primária. [Carneiro \(1996\)](#) apresenta uma aplicação interessante deste modelo para o problema de desenho de redes de distribuição de energia secundária. Neste caso, todas as demandas devem ser supridas pela tecnologia secundária. Entretanto, ainda é vantajoso utilizar a tecnologia primária (neste caso, as altas voltagens) até muito próximo aos pontos de demanda para que tenha custos das variáveis bem menores. A conversão de uma primeira tecnologia para outra requer a instalação de equipamentos adicionais nos nodos da rede (transformadores elétricos) e é limitado a capacidade desses transformadores.

Aplicação da Decomposição de Benders: [Radazzo et al. \(2001\)](#) utilizam a decomposição de Benders para solucionar este problema. O algoritmo é muito similar aquele apresentado por [Randazzo e Luna \(2001\)](#), mas sem a utilização da relaxação linear. A primeira solução viável é obtida por meios do algoritmo do caminho mínimo considerando somente a variável de custos. A solução inicial é utilizada para gerar os limites iniciais inferiores e superiores. Os testes foram conduzidos em um conjunto limitado de instâncias acima de 41 nós (todas as instâncias poderiam ser solucionadas na otimalidade com o modelo de relaxação linear) onde a decomposição de Benders foi usualmente mais rápida (especialmente para as instâncias maiores) que o Cplex 3.0.

# Capítulo 3

## Modelagem Matemática

### 3.1 Introdução

*"O que vale na vida não é o ponto de partida e sim a caminhada. Caminhando e semeando, no fim terás o que colher.."*

Cora Coralina

O problema de projeto de redes consiste em ligar um conjunto de servidores aos seus diversos clientes, satisfazendo a demanda a mínimo custo. Cada arco da rede possui três custos associados: um custo operacional variável por unidade de fluxo, um custo fixo de instalação de infra-estrutura e um custo não linear de congestionamento, que penaliza fluxos próximos de níveis de capacidade ímplicitos. Este problema pode ser visto como uma generalização do problema da árvore de Steiner em um grafo direcionado [Miranda Jr \(2004\)](#). De fato, ao considerar custos variáveis nos arcos, tem-se basicamente o problema de Steiner [Maculan \(1987, 1986\)](#); [Winter \(1987\)](#); [Hwang e Richards \(1992a,b\)](#). Desprezando os custos fixos, tem-se um problema de transbordo de fonte única [Dantzig \(1962\)](#).

### 3.2 Uma Formulação de Fluxos Multiproduto

Uma rede de comunicação é representada por um grafo direcionado  $G(V, E)$ , onde  $V$  representa o conjunto de nós,  $E$  representa o conjunto de arcos que representam os enlaces válidos de comunicação entre os nós. Um arco, ou enlace de comunicação entre 2 nós  $i$  e  $j$ , é representado por  $(i, j) \in E$  onde  $i, j \in V$ . Seja ainda o nó  $k$ , o servidor, que deve receber os dados de um número  $|K|$  de nós ativos. Cada um deles com uma demanda  $d_k$  diferente onde  $k \in K$  e  $K \subseteq V$ .

### 3.2.1 Variáveis e Parâmetros

O custo unitário de transporte/transmissão da demanda do nó  $k \in K$  no arco  $(i, j) \in E$  é representado por  $c_{ijk}$ . Assume-se que  $c_{ijk} = \gamma^k l_{ij}$ ,  $\forall k \in K$ , onde  $\gamma^k$  é o custo específico de transporte da demanda do nó  $k$  e  $l_{ij}$  é a distância entre os nós  $i$  e  $j$ . Desta forma, o custo unitário de transporte fica dependente tanto do nó da demanda e quanto do comprimento do arco. Caso isto não seja necessário, faz-se  $\gamma^k = \gamma$ ,  $\forall k \in K$ . O custo de ativação do arco é dado por  $b_{ij}$ , onde  $b_{ij} = \zeta_{ij} l_{ij}$ ,  $\forall (i, j) \in E$  e  $\zeta_{ij}$  é o custo da instalação da infra-estrutura por unidade de distância.

A formulação proposta utiliza ainda as seguintes variáveis de decisão:  $x_{ij} \in \{0, 1\}$ ,  $\forall (i, j) \in E$ , que indica se o arco  $(i, j)$  é ativado ( $x_{ij} = 1$ ) ou não ( $x_{ij} = 0$ );  $f_{ijk} \geq 0$ , para representar o fluxo da demanda do nó  $k \in K$  passando pelo arco  $(i, j) \in E$ ; e  $g_{ij} \geq 0$ , representando o fluxo global do arco  $(i, j) \in E$ .

A função convexa não-linear de degradação de serviço é representada pelo custo de congestionamento no arcos, sendo separável por arco e representada por  $\tau_{ij}(g_{ij}) = e g_{ij}^p$ ,  $\forall (i, j) \in E$ , onde  $e > 0$  e  $p \geq 1$  são escalares representando a participação dos custos de congestionamento na composição do custo total. Valores elevados para  $e$  e  $p$  implicam em um peso maior do congestionamento no projeto da rede.

## 3.3 Modelo Não Linear Inteiro Misto

O modelo de programação não linear balanceia o consumo enquanto roteia o produto através da rede.

Então, o modelo matemático M é dado por:

$$\min \sum_{(i,j) \in E} [b_{ij}x_{ij} + \tau_{ij}(g_{ij}) + \sum_{k \in K} c_{ijk}f_{ijk}] \quad (3.1)$$

sujeito a:

$$\sum_i x_{ij} \leq 1 \quad \forall i \in V \quad (3.2)$$

$$\sum_{k \in K} f_{ijk} - g_{ij} \leq 0 \quad \forall (i, j) \in E \quad (3.3)$$

$$\sum_{(o,j) \in E} f_{ojk} = d_k \quad \forall k \in K \quad (3.4)$$

$$\sum_{(i,k) \in E} f_{ikk} = d_k \quad \forall k \in K \quad (3.5)$$

$$\sum_{(i,j) \in E} f_{ijk} = \sum_{(j,l) \in E} f_{jlk} \quad \forall j \in V \setminus \{o\},$$

$$k \in K : k \neq j \quad (3.6)$$

$$f_{ijk} \leq d_k x_{ij} \quad \forall (i, j) \in E, k \in K \quad (3.7)$$

$$f_{ijk} \geq 0 \quad \forall (i, j) \in E, k \in K \quad (3.8)$$

$$g_{ij} \geq 0 \quad \forall (i, j) \in E \quad (3.9)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \quad (3.10)$$

A função objetivo (3.1) possui 3 termos: O primeiro termo contabiliza o custo total da instalação dos arcos; o segundo termo sumariza os custos de alocação de banda e congestionamento; enquanto que o terceiro termo totaliza o custo total de transporte dos produtos pelos arcos  $(i, j)$  com destino os nodos de demanda  $k$ .

A restrição(3.2) assegura que o número de arcos que deixa o nodo é menor ou igual a um. A restrição (3.3) calcula o fluxo total de transporte dos produtos que passam através de cada arco  $(i, j)$ . A restrição (3.4) garante que o fluxo total do produto  $k$  originado no nodo  $o$  (origem) seja igual à somatória de todas as demandas de todos os pontos  $k$ . A restrição (3.5) assegura que o fluxo total do produto  $k$  que chega a um nodo de destino seja igual à demanda do produto  $k$ . A restrição (3.6) garante que a conservação do fluxo de energia para cada nodo funcione como um *multi-hoop* ou nodo de Steiner. As restrições de acoplamento  $f$  e  $x$  (3.7) permite que o fluxo passe através do arco  $(i, j)$  somente se o arco  $(i, j)$  existir. Finalmente, as restrições (3.8) e (3.9) são restrições de não negatividade dos fluxos das variáveis  $f_{ijk}$  e  $g_{ij}$ , respectivamente, enquanto que a restrição (3.10) afirmam a integralidade das variáveis  $x_{ij}$ .

As variáveis inteiras  $x_{ij}$  são as variáveis responsáveis pelo caminho entre os nodos sorvedouros até o nodo raiz, sendo consideradas como variáveis topológicas. Por conseguinte, uma propriedade do modelo M é que, para os valores fixos viáveis das variáveis inteiras  $x_{ij}$  afetam as restrições (3.7) e (3.10), obtém-se  $|K|$  subproblemas de fluxo em redes não lineares. Os subproblemas não lineares resultantes possuem uma função objetivo convexa diferenciável, restrições lineares e variáveis de fluxo contínuas, implicando o uso de estratégias de dualização para tratar de termos não lineares. A

utilização dessas estratégias aponta para um método que ajustaria os valores das variáveis inteiras  $x_{ij}$  iterativamente até a otimalidade ser obtida. Um desses métodos é o método clássico da decomposição [Benders \(1962\)](#), mais especificamente Decomposição Generalizada de Benders proposta por [Geoffrion \(1972a\)](#) que lida com problemas não lineares.

Antes de introduzir o método de Decomposição de Benders, existem 4 propriedades teóricas que devem ser destacadas.

1. A relaxação contínua do modelo  $M$ , chamada de  $M_{Lin}$  gera um politopo *quasi-integral*. Um politopo *quasi-integral* é um politopo no qual as arestas da casca convexa e todos os seus pontos inteiros fazem parte da aresta do politopo. Portanto, se a solução ótima inteira existe, então existe um caminho através dos pontos extremos inteiros do politopo.
2. Além do mais, para  $\tau_{ij}(g_{ij}) = 0, \forall \beta_{ij} = 0$  e  $d_k = 1, \forall k \in K$ , o modelo é reduzido a formulação de fluxo de multi-produto do problema de Steiner em grafos direcionados, apresentados por [Claus e Maculan \(1983\)](#) e outros [Wong \(1984\)](#), [Maculan et al. \(1988\)](#), [Goemans e Mying \(1993\)](#).
3. Considerando somente a função objetivo [3.1](#) e as restrições [\(3.4 - 3.10\)](#), tem-se um caso particular da formulação de programação linear para o problema da árvore de Steiner mínima direcionada, também introduzida por [Maculan \(1986\)](#).
4. [Queiroz e Humes \(2001\)](#) demonstraram uma topologia de árvore no qual é associada com uma solução ótima quando  $\tau_{ij}(g_{ij})$  é uma função de custo concâva do fluxo  $g_{ij}$  que passa através do arco  $(i, j)$ , com  $b_{ij} = 0$  e  $c_{ijk} = 0, \forall k \in K$ .

Todas essas 4 propriedades teóricas podem ajudar a entender por que a relaxação da programação linear para o modelo  $M$  pode automaticamente lidar com soluções ótimas integrais em muitas instâncias para este caso particular.

No próximo capítulo, um algoritmo eficiente derivado da formulação proposta e baseado nas técnicas Decomposição Generalizada de Benders [Geoffrion \(1972a\)](#) e Aproximação Externa [Duran e Grossman \(1986\)](#) será descrito.

# Capítulo 4

## Técnicas de Decomposição

*"Para realizar grandes conquistas, devemos não apenas agir, mas também sonhar; não apenas planejar, mas também acreditar"*

Anatole France

### 4.1 Decomposição de Benders

Decomposição de Benders é uma abordagem clássica de solução de problemas de otimização combinatória, baseada nas idéias de particionamento e geração de restrições atrasadas. Em 1962, [Benders \(1962\)](#) propôs um método de particionamento para resolver problemas de programação inteira mista e não linear. A abordagem de Benders define uma relaxação do algoritmo para resolver um problema que trabalha iterativamente em um problema inteiro e linear. O método particiona o problema em dois subproblemas: o problema mestre e o problema escravo. O primeiro consiste em uma versão relaxada do problema original com um conjunto de restrições associadas, enquanto que o segundo consiste na versão original do problema com os valores das variáveis inteiras fixadas para o problema mestre.

O sucesso computacional da Decomposição de Benders em resolver problemas de grande escala tem se confirmado desde o paper pioneiro de [Geoffrion e Graves \(1974\)](#), o problema de desenho de rede não capacitado com arcos não direcionados de [Magnanti et al. \(1986\)](#), o problema de associação das locomotivas e vagões de [Cordeau e Desrosiers \(2000, 2001\)](#) e o problema de desenho de rede multicasting de [Miranda Jr \(2004\)](#) e [Randazzo e Luna \(2001\)](#).

Vários autores tem trabalhado em extensões e aprimoramentos do método de Decomposição de Benders. [Geoffrion \(1972b\)](#) aplicou o algoritmo básico para problemas não lineares, formulando a decomposição generalizada de Benders. [Magnanti e Wong \(1981\)](#) propôs uma metodologia para melhorar a performance do algoritmo de Decom-

posição de Benders abordando a questão da seleção da variável dual. [McDaniel e Devine \(1977\)](#) desenvolveram um algoritmo para acelerar a decomposição removendo temporariamente as restrições de integralidade do problema mestre substituindo-as pela relaxação linear. [Balas e Bergthaller \(1983\)](#) propuseram uma abordagem que gerasse cortes fortes no início das iterações do método.

Para explicar o método de Decomposição de Benders, será necessário reformular o modelo  $M$  em um modelo com poucas variáveis, porém com um conjunto considerável de restrições, sendo uma restrição para cada solução topológica viável. Felizmente, a maioria dessas restrições são inativas na otimalidade, não precisam de consideração explícita. Então será criado um problema mestre relaxado (PRM), onde, para cada iteração, o método de Decomposição de Benders fixa as variáveis complicantes  $x$  (variáveis topológicas) em um nível superior, (RPM) produzindo um limite inferior; enquanto que no nível inferior, chamado de subproblema (SP), os problemas de fluxos contínuos não lineares são solucionados, produzindo um limite superior. Após obter uma solução ótima para o subproblema, uma nova restrição, chamada de corte de Benders, é adicionada ao problema mestre para ajustar as variáveis topológicas na próxima iteração. O procedimento termina quando os limites inferior e superior se igualam. O ajuste da função objetivo não linear no subproblema é necessário. Isso é conseguido através da dualização das restrições acopladas associadas a não linearidade. Após a dualização, o resultado dos subproblemas são relativamente fáceis de resolver.

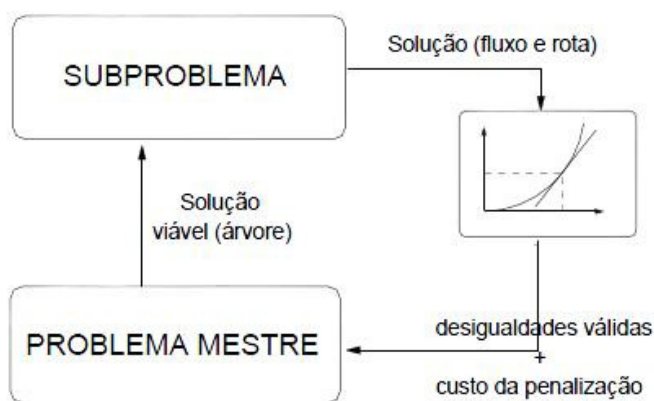


Figura 4.1: Algoritmo para a resolução do problema

### 4.1.1 Problema Mestre

O método de Decomposição de Benders trata essencialmente problemas de projeção, seguido por linearização externa, dualização e relaxação. Será demonstrada uma projeção do problema  $M$  no espaço das variáveis topológicas  $x$ , resultando no seguinte problema:

$$\min_{x \in X} \sum_{(i,j) \in E} b_{ij}(x_{ij}) + v(x) \quad (4.1)$$

onde  $X = \{x \mid \text{valores fixos de } x \text{ no qual existem fluxos viáveis que satisfazem (3.2 - 3.9)}\}$  e onde  $v(x)$  é calculado resolvendo o seguinte problema:

$$v(x) = \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij} g_{ij} + \sum_{k \in K} c_{ijk} f_{ijk}] \quad (4.2)$$

sujeito a condição (3.7) para  $x$  fixo onde  $G = \{(f, g) \mid f \geq 0 \text{ e } g \geq 0\}$  satisfazendo (3.2 - 3.6).

O requisito da viabilidade dos fluxos das variáveis topológicas  $x \in X$  implica uma arborescência roteada do nodo de origem  $o$  (nodo fonte) desde que exista um caminho a partir de um nodo de demanda  $k \in K$  para o nodo raiz  $o$ . Isso garante que já não são mais necessárias restrições de viabilidade para o domínio do problema projetado. Além do mais, as condições de Karush-Kunh-Tucker (KKT) são necessárias e suficientes para a otimalidade, desde que o subproblema 4.2 possua uma função objetivo convexa diferenciável, restrições lineares e um mínimo assegurado pela minimização da função convexa em um conjunto compacto não vazio. Então o problema 4.2 se torna propício para a aplicação das técnicas de Decomposição. Geoffrion (1972b)

Com o vetor  $\alpha \geq 0$  das variáveis duais, as restrições de acoplamento são dualizadas (3.7). Desde que não haja um gap de dualidade para qualquer  $x \in X$ , o valor ótimo do subproblema (4.2) é dado por:

$$v(x) = \max_{\alpha \geq 0} [\min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} c_{ijk} f_{ijk} + \sum_{k \in K} \alpha_{ijk}(f_{ijk} - d_k x_{ij})]] \quad (4.3)$$

ou

$$v(x) = \max_{\alpha \geq 0} \left[ \sum_{(i,j) \in E} \sum_{k \in K} -\alpha_{ijk} d_k x_{ij} + \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij} g(ij) + \sum_{k \in K} (c_{ijk} + \alpha_{ijk}) f_{ijk}] \right] \quad (4.4)$$

O problema completo (4.1) é equivalente a:

$$\min_{x \in X} \left\{ \sum_{(i,j) \in E} b_{ij} x_{ij} + \max_{\alpha \geq 0} \left[ \sum_{(i,j) \in E} \sum_{k \in K} -\alpha_{ijk} d_k x_{ij} + \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij} g(ij) + \sum_{k \in K} (c_{ijk} + \alpha_{ijk}) f_{ijk}] \right] \right\} \quad (4.5)$$

ou sabendo que o supremo é o menor limite superior e com a ajuda da variável  $t$  representando o melhor limite inferior da soma dos custos de alocação de banda e congestionamento, o problema (3.1-3.10) é equivalente ao problema mestre

$$\min_{t, x \in X} \sum_{(i,j) \in E} b_{ij} x_{ij} + t \quad (4.6)$$

sujeito à:

$$t \geq \sum_{(i,j) \in E} \sum_{k \in K} -\alpha_{ijk} d_k x_{ij} + \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij} g(ij) + \sum_{k \in K} (c_{ijk} + \alpha_{ijk}) f_{ijk}], \quad \forall \alpha \geq 0 \quad (4.7)$$

O problema mestre equivalente ao problema original pode resultar em um sistema onde o número de restrições (4.7) é uma função exponencial do número de  $x$  variáveis. Então, na próxima seção, descreve-se como solucionar este problema através da projeção de  $x$  variáveis e substituindo nas restrições, originando o problema mestre relaxado. Apesar disso parecer resultar em um grande número de restrições, estas restrições são adicionadas a base necessária, além do mais, somente um pequeno número dessas restrições são necessárias para obter a solução ótima.

### 4.1.2 O Problema Mestre Relaxado

O método de Decomposição de Benders generalizado resolve o problema (4.6-4.7) baseado na estratégia de relaxação, por exemplo, ignorando um pequeno grupo de restrições

(4.7). Esta estratégia requer um procedimento que adiciona iterativamente as restrições ao problema mestre quando necessário. Então, a cada iteração  $h$ , o valor ótimo de  $v(x^h)$  é encontrado a partir de (4.4) após resolver o subproblema de uma dado desenho de rede  $x^h$  e recupera o vetor dos multiplicadores  $\alpha^h$  ( $\alpha = \alpha^h$ ). Portanto o valor ótimo de  $v(x^h)$  é dado por

$$v(x^h) = \sum_{(i,j) \in E} \sum_{k \in K} -\alpha_{ijk}^h d_k x_{ij}^h + \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij} g(i,j) + \sum_{k \in K} (c_{ijk} + \alpha_{ijk}^h) f_{ijk}] \quad (4.8)$$

A partir de (4.7), tem-se a seguinte restrição associada com  $\alpha^h$ :

$$t \geq \sum_{(i,j) \in E} \sum_{k \in K} -\alpha_{ijk}^h d_k x_{ij}^h + \min_{(f,g) \in G} \sum_{(i,j) \in E} [\tau_{ij} g(i,j) + \sum_{k \in K} (c_{ijk} + \alpha_{ijk}^h) f_{ijk}] \quad (4.9)$$

e utilizando o valor do mínimo dado por (4.8), obtém-se o seguinte corte baseado em  $x^h$  e  $\alpha^h$ , conhecido como o corte de Benders do tipo I.

$$t \geq v(x^h) + \sum_{(i,j) \in E} \sum_{k \in K} \alpha_{ijk}^h d_k (x_{ij}^h - x_{ij}) \quad (4.10)$$

A cada iteração  $h$  de Benders, faz-se  $\sum_{(i,j) \in E} \sum_{k \in K} \alpha_{ijk}^h d_k x_{ij}^h = 0$ , assim têm-se o problema mestre relaxado

$$\min_{t, x \in X} \sum_{(i,j) \in E} b_{ij} x_{ij} + t \quad (4.11)$$

sujeito à:

$$\begin{aligned} t &\geq v(x^h) - \sum_{(i,j) \in E} \sum_{k \in K} \alpha_{ijk}^h d_k x_{ij}^h, h = 1, \dots, H \\ x_{ij} &\in 0, 1 \forall (i, j) \in E \\ t &\geq 0 \end{aligned} \quad (4.12)$$

onde  $H$  na restrição (4.11) indica o número máximo de cortes de Benders que podem ser inseridos para se obter a solução ótima. Infelizmente, a maneira de como o problema mestre relaxado apresentado acima não garante que a solução obtida a cada iteração  $h$  constitui uma árvore do nodo fonte ao nodos sorvedouros, exemplo, uma arborescência. Algumas vezes, a solução dada pelo problema mestre relaxado pode implicar em ciclos na topologia proposta. A fim de superar esta situação, explora-se a estratégia de trabalhar com ambos os espaços de variáveis de  $x$  e  $g$  no problema mestre relaxado. Fazendo isto, e com a ajuda de algumas restrições redundantes, o balanço de fluxo é

mantido fazendo com que todas as soluções obtidas através da relaxação do problema mestre forme uma arborescência, exemplo, a viabilidade das variáveis estruturais  $x$  é garantida. Esta abordagem explora a informação de acoplamento das variáveis  $f_{ijk}$  e  $g_{ij}$  do modelo original  $M$ . Então, o novo problema mestre relaxado é agora dado por:

$$\min \sum_{(i,j) \in E} b_{ij} x_{ij} + t \quad (4.13)$$

sujeito a:

$$\sum_i x_{ij} \leq 1 \quad \forall i \in V \quad (4.14)$$

$$\sum_{(i,k) \in E} g_{ik} - \sum_{(k,j) \in E} g_{kj} = d_k \quad \forall k \in K \quad (4.15)$$

$$\sum_{(i,j) \in E} g_{ij} - \sum_{(j,l) \in E} g_{jl} = 0 \quad \forall j \in V \setminus K \cup \{o\} \quad (4.16)$$

$$- \sum_{(o,j) \in E} g_{oj} = - \sum_{k \in K} d_k \quad (4.17)$$

$$g_{ij} \leq \sum_{k \in K} d_k x_{ij} \quad \forall (i,j) \in E \quad (4.18)$$

$$t \geq v(x^h) + \sum_{(i,j) \in E} \sum_{k \in K} \alpha_{ijk}^h d_k (x_{ij}^h - x_{ij}) \quad \forall h = 1, \dots, H \quad (4.19)$$

$$g_{ij} \geq 0 \quad \forall (i,j) \in E$$

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

$$t \geq 0$$

As restrições (4.15-4.17) são restrições de balanço de fluxo no espaço de variáveis  $g$ . As restrições adicionais e o conjunto  $|E|$  de variáveis são onerosas, mas estes evitam o uso dos cortes de Benders do tipo II (raios extremos). Esse tipo de restrição também garante a viabilidade do balanço de fluxo ao custo de uma iteração do problema mestre relaxado, exemplo, um enorme e desnecessário esforço computacional que não ajuda na busca da solução ótima.

A estratégia de ignorar um grupo pequeno de restrições do problema mestre, seção 4.1.1, requer a adição dos cortes de Benders, restrição 4.19 ao problema mestre relaxado a cada iteração  $h$ . Então, para cada iteração  $h$  obtém-se os valores ótimos dos multiplicadores  $\alpha_h$  e o valor de  $v(x^h)$ , resolvendo o subproblema, conforme descrito na seção 4.1.3.

### 4.1.3 SubProblemas Primal e Dual

Para uma dada arborescência  $A^h$  associada a um vetor  $x^h$  fixado pela iteração  $h$  do problema relaxado, o cálculo do fluxo de custo mínimo  $v(x^h)$  pode ser separado em uma série de problemas triviais de fluxo em rede. Seja  $C_{ok}^h$  um caminho entre o nodo fonte  $o$  ao nodo de demanda  $k$ , então o subproblema pode ser assim resolvido

$$\min_{g \geq 0, f \geq 0} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} c_{ijk} f_{ijk}] \quad (4.20)$$

sujeito as restrições do fluxo dos arcos de acoplamento:

$$\sum_{k \in K} f_{ijk} - g_{ij} \leq 0, \forall (i, j) \in E \quad (4.21)$$

e também sujeito as restrições (3.4 - 3.7) para um vetor binário fixo  $x = x^h$ .

Desde que  $\tau_{ij}g_{ij}$  é uma função de crescimento e, uma solução ótima, a restrição (3.3) é satisfeita com igualdade restrita, dualiza-se as restrições de acoplamento (3.3) utilizando as variáveis duais (multiplicadores de Lagrange)  $\beta \geq 0$ . O resultado do problema é separado em dois subproblemas: um contendo somente as variáveis  $f_{ijk}$  e o outro contendo as variáveis  $g_{ij}$ . Os multiplicadores de Lagrange  $\beta$  são responsáveis pelo acoplamento destes 2 subproblemas. A dualização  $d(\beta)$  e o problema resultante é mostrado abaixo por (4.22) e (4.23) respectivamente.

$$r(\beta) = \min_{g \geq 0, f \geq 0} \sum_{(i,j) \in E} [\tau_{ij}(g_{ij}) + \sum_{k \in K} c_{ijk} f_{ijk}] + \sum_{((i,j) \in E)} \beta \left( \sum_{k \in K} f_{ijk} - g_{ij} \right) \quad (4.22)$$

e

$$r(\beta) = \sum_{k \in K} \min_{f_k \geq 0} \sum_{(i,j) \in E} (c_{ijk} + \beta_{ij}) f_{ijk} + \sum_{(i,j) \in E} \min_{g_{ij} \geq 0} (\tau_{ij}g_{ij} - \beta_{ij}g_{ij}) \quad (4.23)$$

onde cada  $f_k$  refere-se ao vetor de fluxo do produto  $k$  que é viável nas restrições correspondentes (3.4 - 3.7) para cada  $x = x^h$ .

Como somente um único fluxo  $f_{ijk} = d_k$  pode ser associado a cada arco que pertence ao caminho  $C_{ok}^h$  a partir do nodo fonte  $o$  ao nodo de demanda  $k$ , a única solução ótima  $(f^h, g^h)$  é associada a uma arborescência  $x^h$ . Então, após obter o valor apropriado para  $\beta$ , pode-se calcular o valor ótimo de  $v(x^h)$  para cada iteração  $h$  de Benders utilizando o problema dual

$$v(x^h) = \max_{\beta \geq 0} r(\beta) \quad (4.24)$$

Observa-se que, para uma solução ótima  $(f^h, g^h)$  para o problema primal 4.20, 3.3 e (3.4 - 3.7), uma solução ótima associada  $\beta^h$  para o problema dual 4.24 deve minimizar para cada  $g_{ij} \in E$ , a parcela correspondente a função Lagrangeana em (4.23), que implica

$$\beta_{ij}^h = \tau'(g_{ij}^h), \forall (i, j) \in E \quad (4.25)$$

Como consequência de fixar um vetor ótimo único  $\beta^h$ , pode-se agora separar os 2 subproblemas. O segundo termo em 4.23 é facilmente calculado após ter os valores ótimos de  $g^h$  e  $\beta^h$ . O primeiro termo é resolvido separadamente para cada produto  $k \in K$  para qualquer  $x^h$ . As seções 4.1.4 e 4.1.5 descrevem respectivamente, o par de programação linear primal-dual para ser resolvido pelo primeiro termo. Nota-se que não obtém somente o valor ótimo de  $v(x^h)$ , mas também obtém-se os valores ótimos das variáveis duais  $\alpha_{ijk}$  que são utilizadas para construir os cortes de Benders 4.19 a serem adicionados na iteração  $h + 1$ .

#### 4.1.4 SubProblema Primal

Para o nodo fonte  $o$  na iteração  $h$ , o subproblema para o primeiro termo da equação (4.23) é dado por:

$$\min \sum_{(i,j) \in E} (c_{ijk} + \beta_{ij}^h) f_{ijk}^h \quad (4.26)$$

$$\text{s. a: } \sum_{(o,j) \in E} f_{ojk}^h = d_k \quad \forall k \in K \quad (4.27)$$

$$\sum_{(i,k) \in E} f_{ikk}^h = d_k \quad \forall k \in K \quad (4.28)$$

$$\begin{aligned}
-\sum_{(i,j) \in E} f_{ijk}^h + \sum_{(j,l) \in E} f_{ijk}^h &= 0 \\
\forall j \in V - o, \\
j \neq k \text{ e } \forall k \in k & \quad (4.29)
\end{aligned}$$

$$\begin{aligned}
-f_{ijk}^h &\geq -d_k x_{ij}^h, \\
\forall (i,j) \in E \\
\text{e } \forall k \in K & \quad (4.30)
\end{aligned}$$

$$\begin{aligned}
f_{ijk}^h &\geq 0 \\
\forall (i,j) \in E \\
\text{e } \forall k \in K & \quad (4.31)
\end{aligned}$$

O subproblema primal (4.27)-(4.31) possui solução única dada pela solução trivial de (4.32)

$$f_{ijk}^h = \begin{cases} d_k & \text{se } (i,j) \in C_{ko}^h \subseteq A^h \\ 0 & \text{caso contrário} \end{cases} \quad (4.32)$$

Com a solução do subproblema primal, pode-se calcular o valor de  $v(x^h)$  facilmente a cada iteração de Benders. Infelizmente ao fazer isso, não encontra-se os valores ótimos das variáveis duais  $\alpha_{ijk}^h$ . Para obter os valores através da solução do problema dual do subproblema (4.27-4.31), será demonstrado na seção 4.1.5.

#### 4.1.5 SubProblema Dual

Associando os variáveis duais  $p_{ok}^h$ ,  $p_{kk}^h$ ,  $p_{jk}^h$ ,  $p_{ik}^h$  e  $\alpha_{ijk}^h$  as restrições (4.27), (4.28), (4.29) e (4.30), respectivamente, e lembrando que  $x_{ij}^h$  é fixo para o problema mestre relaxado, o subproblema dual é dado por:

$$\max_{p^h, \alpha^h \geq 0} d_k (p_{kk}^h - p_{ok}^h - \sum_{(i,j) \in E} x_{ij}^h \alpha_{ijk}^h) \quad (4.33)$$

sujeito à:

$$p_{jk}^h - p_{ik}^h - \alpha_{ijk}^h \leq c_{ijk} + \beta_{ij}^h, \forall (l,j) \in E \quad (4.34)$$

## 4.2 Aproximação Externa

A técnica conhecida como Aproximação Externa, do inglês Outer Approximation (OA), foi desenvolvida em trabalhos pioneiros de [Duran e Grossman \(1986\)](#), [Fletcher e Leyfer. \(1994\)](#) e [Yuan et al. \(1988\)](#). É um método muito simples mas uma técnica eficiente para resolução de problemas inteiros não lineares, utilizando uma abordagem de plano cortantes. A técnica de aproximação externa tem sido aplicada em processos de fluxograma estruturados de otimização( [Duran e Grossman \(1986\)](#), [Karuppiah et al. \(2008\)](#)), e em várias outras aplicações de processos de engenharia. Mais recentemente, o método tem sido aplicado em aplicações logísticas [Huang et al. \(2005\)](#) e em problemas de otimização em geral [Grossmann e Kravanja \(1995\)](#).

O método também possui uma técnica de coordenação entre o problema mestre e o subproblema, trabalhando similarmente a técnica de GDB, mas os problemas mestres de OA são escritos no espaço de todas as variáveis do problema, dispensando a projeção do espaço entre as variáveis inteiras. Este recurso melhora o poder dos planos cortantes associados e assegura a convergência em um número de iterações menor que os obtidos ao se aplicar a técnica de GBD. Como desvantagem, a solução do problema mestre é mais cara, geralmente aumentando o tempo de solução e o esforço computacional à medida que o problema aumenta.

Para entender o desenvolvimento da técnica de OA, uma revisão geral do método é necessária. Dado um *PNLIM* em sua representação algébrica mais básica, onde  $x$  e  $z$  representam o conjunto das variáveis contínuas e discretas, respectivamente,  $f : \mathbb{R}^{qxs} \mapsto \mathbb{R}$  e  $g : \mathbb{R}^{qxs} \mapsto \mathbb{R}^m$  são duas funções continuamente diferenciáveis, e  $Z$  e  $X$  são conjuntos poliedrais:

$$\min \quad f(z, x) \tag{4.35}$$

sujeito à:

$$g_j(z, x) \leq 0, \quad \forall j = 1, \dots, m \tag{4.36}$$

$$z \in Z, Z \in \mathbb{Z}^q \tag{4.37}$$

$$x \in \mathbb{X} \tag{4.38}$$

É possível reduzir este problema a um problema não linear puro ao escolher um vetor fixo  $z = z^h, z \in Z$ , para cada iteração  $h$ , produzindo o seguinte subproblema não linear:

$$\min \quad f(z^h, x) \quad (4.39)$$

sujeito à:

$$g_j(z^h, x) \leq 0, \quad \forall j = 1, \dots, m \quad (4.40)$$

$$x \in \mathbb{X} \quad (4.41)$$

Quando o problema acima é resolvido (4.39 - 4.41), este permite deduzir o gradiente da função  $f(z, x)$  e  $g_j(z, x)$ ,  $\forall j$  em  $(z^h, x^h)$ . Se nenhuma restrição de viabilidade é requerida, então é possível reescrever o problema (4.35-4.38) como sendo equivalente ao problema inteiro misto (PIM:)

$$\min \quad \xi \quad (4.42)$$

sujeito à:

$$\xi \geq f(z^h, x^h) + \nabla f(z^h, x^h)^T \begin{pmatrix} z - z^h \\ x - x^h \end{pmatrix}, \quad \forall h = 1, \dots, H \quad (4.43)$$

$$0 \geq g(z^h, x^h) + \nabla g(z^h, x^h)^T \begin{pmatrix} z - z^h \\ x - x^h \end{pmatrix}, \quad \forall h = 1, \dots, H \quad (4.44)$$

$$z \in Z, Z \in \mathbb{Z}^q \quad (4.45)$$

$$x \in \mathbb{X} \quad (4.46)$$

$$\xi \in \mathbb{R} \quad (4.47)$$

onde  $H$  representa o número máximo de iterações e  $\xi$  representa a variável de estimativa para função objetivo.

O problema (4.42)-(4.47) é conhecido como problema mestre OA. As restrições (4.43)-(4.44) são responsáveis pela performance da função objetivo do OA e sua região viável, respectivamente. Quando as restrições  $g(z, x)$  são propriamente convexas e a restrição de qualificação é garantida para cada solução de (4.39 - 4.41), então a restrição (4.44) é necessária e suficiente para o OA aproximar da região viável. A restrição (4.44) têm dupla função no método OA: além de assegurar a viabilidade, são responsáveis por realizar a aproximação externa da região viável, e  $g(z, x)$  for não linear.

No caso da formulação (3.1)-(3.10), a função objetivo é separável em termos lineares e não lineares ( $\sum_{(i,j) \in E} \tau_{ij}(g_{ij})$ ). Logo, este é o único termo que precisa sofrer a

aproximação externa. Então, ao aplicar a técnica OA deve-se substituir somente  $\tau_k(g_k)$  por  $\xi_k$  para cada  $k$  na função objetivo e adicionar a restrição (4.43) no formato de (4.49).

$$\min \sum_{(i,j) \in E} [b_{ij}x_{ij} + \xi_{ij} + \sum_k c_{ijk}f_{ijk}] \quad (4.48)$$

sujeito à

$$(3.4) - (3.10)$$

e

$$(4.15) - (4.17)$$

e

$$\xi_{ij} \geq \tau_{ij}(g_{ij}^h) + \beta_{ij}^h(g_{ij} - g_{ij}^h), \forall (i,j) \in E, \forall h = 1, \dots, H \quad (4.49)$$

$$\xi_{ij} \geq 0 \quad (4.50)$$

A restrição (4.44) não está presente na formulação (4.48)-(4.50), porque as restrições de acoplamento são lineares (poliedrais), fazendo desnecessário a inserção destes cortes. Além do mais, como esperado, esta formulação é muito grande para ser resolvida eficientemente. A grande dimensão do conjunto de variáveis  $x$  pode representar um encargo do computador durante a solução do problema mestre OA. Uma maneira de lidar com esta desvantagem é projetar essas variáveis para fora, desde que algumas manipulações sejam realizadas. A idéia aqui é permitir a solução do problema mestre OA, por meio de algoritmo de Decomposição de Benders. Cria-se então um método híbrido: trata-se com a não linearidade via OA e com a parte de grande escala via Decomposição de Benders.

A idéia aqui é resolver a parcela não-linear, aplicando os cortes OA, e a parcela linear de grande escala, aplicando os cortes de Benders, de forma pseudo-paralela. Isto significa que os cortes (4.52) - (4.53) podem ser adicionados à variante relaxada em qualquer ordem.

$$\min \sum_{(i,j) \in E} [b_{ij}x_{ij} + \xi_{ij} + t_k] \quad (4.51)$$

sujeito à:

$$\xi_{ij} \geq \tau(g_{ij}^h) + \beta_{ij}^h(g_{ij} - g_{ij}^h), \forall (i, j) \in E, \forall h \in H \quad (4.52)$$

$$t_k \geq v(x^s) - \sum_{(i,j)} \alpha_{ijk}^s d_k x_{ij}, \forall k \in K, \forall s \in S \quad (4.53)$$

e

(4.15) - (4.17)

# Capítulo 5

## Experimentos Computacionais

*"Se as coisas são inatingíveis...Ora! Não é motivo para não querê-las... Que triste os caminhos, se não for a presença distante das estrelas!"*

Mário Quintana

Para avaliar a performance dos algoritmos propostos, dois conjuntos de experimentos computacionais foram desenvolvidos. Para estes dois conjuntos, foram utilizadas o conjunto de instâncias disponíveis em [QAPLIB](#). Os nomes das instâncias foram codificadas como dadosn, onde  $n$  é o número da instância ou seja, o número de nós da rede. As instâncias tem tamanho de 12, 14, 15, 16, 17, 18, 20, 21, 22, 24, 25, 27 respectivamente. Para cada instância, o fator de congestionamento foi variado entre 0.001 à 0.9.

O número de variáveis inteiras e contínuas, e os tamanhos dos conjuntos  $E$  e  $K$  são mostrados na Tabela 5.1.

Tabela 5.1: Espaço das Variáveis das 12 instâncias

Instância	E	K	Num Var Int	Num Var Cont
12	132	5	132	729
14	182	3	182	728
15	225	3	225	900
16	256	5	256	1536
17	289	5	289	1734
18	306	6	306	2134
20	380	5	380	2280
21	420	5	420	2520
22	462	5	462	2772
24	552	4	552	2760
25	600	4	600	3000
27	702	5	702	4212

Os dois conjuntos de testes demonstram como a infraestrutura da rede altera quando o fator de congestionamento é incrementado. Dependendo de como os parâmetros da função de custo do congestionamento são alterados, é possível solucionar na otimalidade instâncias grandes utilizando o algoritmo de GBD. Os testes computacionais foram escritos na linguagem AMPL CPLEX utilizando um sistema operacional Linux 64 bits. Os experimentos foram testados em uma estação de trabalho com um processador Intel Core 2 Quad 2.4 GHz e 8Gb de RAM, com tempo limite de 86.400 segundos. Os valores \* descritos em cada tabela representam as instâncias que tiveram tempo computacional superior ao tempo limite.

A figura 5.1 mostra a comparação entre os algoritmos de GBD e OA . A técnica de de OA demonstrou superior performance em instâncias de tamanhos maiores, sendo infinitamente mais rápido e levando menos tempo para resolver o problema em otimalidade.

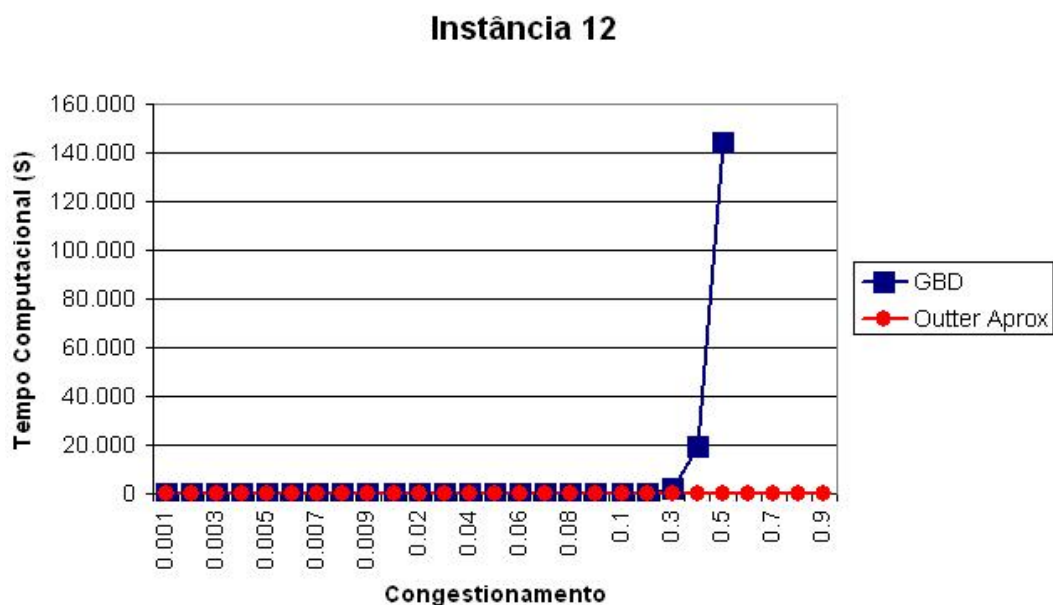


Figura 5.1: Análise da Instância de 12 nós

A figura 5.2 ilustra um exemplo do impacto do congestionamento em uma instância de 18 nós. A parte (a) apresenta os nós e as conexões possíveis, na qual os vértices em negrito e branco representam os nós de demanda unitária e os de transbordo, respectivamente; e os valores do lado das arestas representam o comprimento das mesmas. O sorvedouro é representado pelo círculo 1. Na parte (b), a configuração é mostrada, enquanto na (c) tem-se a estrutura ótima ao se incluir os efeitos de degradação de serviço ( $e=0,1$ ). Nestas duas últimas partes, os valores do lado das arestas representam são o fluxo total passando pelas mesmas. Pode-se ver nas partes (b) e (c) da figura, que as

arestas (2,7), (7,6), (14,13) e (13,18) deixam de ser ativadas ao se considerar os efeitos de congestionamento.

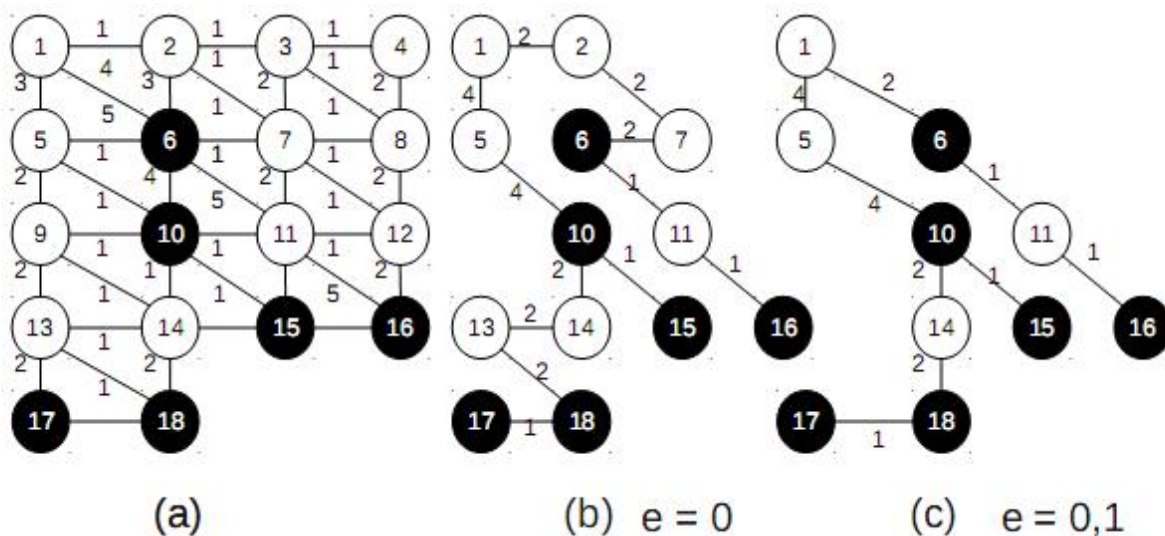


Figura 5.2: Exemplo de um caso com 18 nós

A comparação entre os algoritmos de GBD e OA para cada instância são apresentadas nas tabelas a seguir. Para cada conjunto de testes, foram analisados o valor de congestionamento, o número de iterações, os limites inferior (LB) e superior (UB), o GAP de otimalidade, onde  $GAP = (UB - LB) / UB$ , e tempo computacional gasto.

As tabelas demonstram que o método OA apresenta os melhores desempenhos para níveis maiores de congestionamento. Em média, o método OA é 406,2 vezes mais rápido que o GBD. Para níveis médios ( $e=0.01$ ), o método OA é apenas 37,3 vezes melhor. Observando o número de iterações, o método OA requer em média 158,2 menos iterações do que o GBD.

Isso pode ser melhor entendido ao se observar a redução do efeito de tail-off, ou literalmente o efeito de “cauda longa”, da convergência dos métodos. A Figura 5.3 apresenta a convergência dos dois métodos ao resolver a instância17. Nesta figura os limites LB e UB são plotados em função do número de iterações. Como se pode ver, o método GBD requer um número elevado de iterações até a convergência dos dois limites. Isso é o que é chamado de “cauda longa”. No caso do OA, ambos os limites convergem rapidamente para o ótimo.

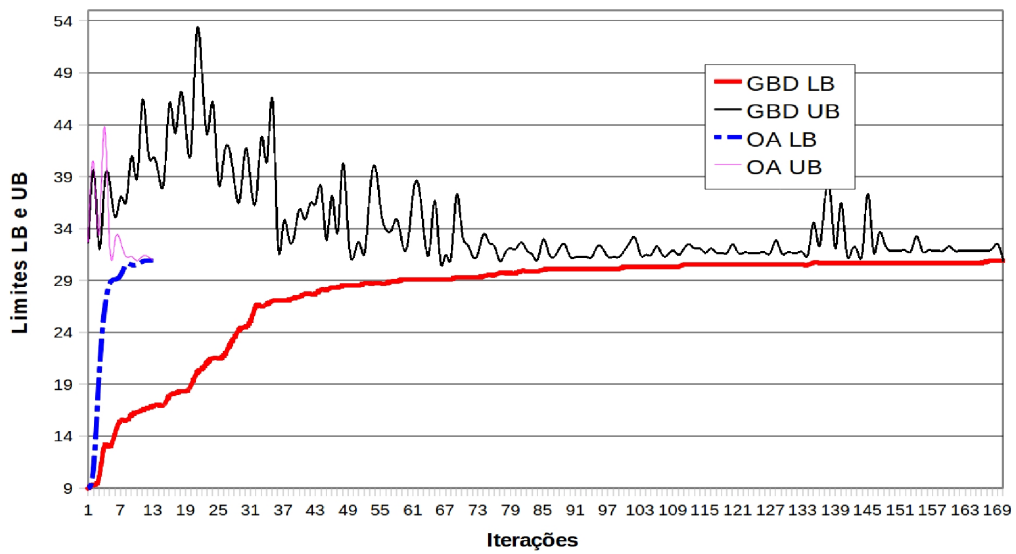


Figura 5.3: Efeito Tail-Off da instância 17 sob congestionamento igual a 10%

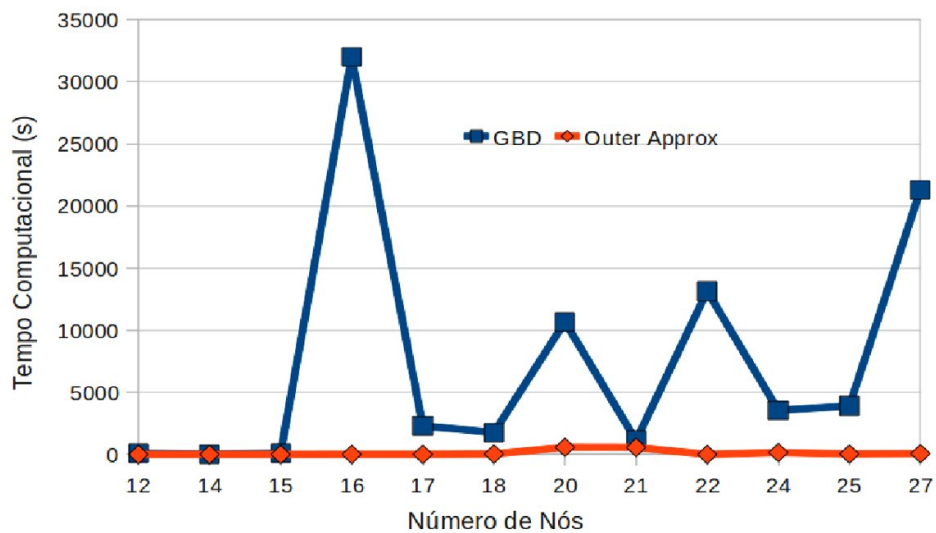


Figura 5.4: Comparação de Tempo Computacional para todas as instâncias com congestionamento igual a 10%

Tabela 5.2: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 12 nós

GDB							OA					
Inst	Cong	It	LB	UP	GAP	T(s)	Cong.	It	LB	UB	GAP	T(s)
12	0,001	8	22,0350	22,0350	0,000	1	0,001	6	22,0350	22,0350	0,000	0
	0,002	8	22,0350	22,0350	0,000	1	0,002	6	22,0700	22,0700	0,000	1
	0,003	9	22,0150	22,0150	0,000	1	0,003	6	22,0150	22,0150	0,000	0
	0,004	10	22,1400	22,1400	0,000	1	0,004	6	22,1400	22,1400	0,000	0
	0,005	10	22,1750	22,1750	0,000	1	0,005	6	22,1750	22,1750	0,000	1
	0,006	11	22,2100	22,2100	0,000	1	0,006	6	22,2100	22,2100	0,000	0
	0,007	14	22,2450	22,2450	0,000	1	0,007	6	22,2450	22,2450	0,000	0
	0,008	10	22,2800	22,2800	0,000	1	0,008	6	22,2800	22,2800	0,000	1
	0,009	10	22,3100	22,3100	0,000	1	0,009	6	22,3100	22,3100	0,000	1
	0,01	10	22,3500	22,3500	0,000	1	0,01	6	22,3500	22,3500	0,000	0
	0,02	10	22,7000	22,7000	0,000	1	0,02	6	22,2700	22,2700	0,000	0
	0,03	16	23,0500	23,0500	0,000	1	0,03	7	23,0500	23,0500	0,000	1
	0,04	27	23,4000	23,4000	0,000	6	0,04	8	23,4000	23,4000	0,000	1
	0,05	42	23,7500	23,7500	0,000	16	0,05	7	23,7500	23,7500	0,000	1
	0,06	51	24,1000	24,1000	0,000	19	0,06	12	24,1000	24,1000	0,000	1
	0,07	53	24,3000	24,3000	0,000	27	0,07	11	24,3300	24,3300	0,000	2
	0,08	60	24,5200	24,5200	0,000	33	0,08	14	24,5200	24,5200	0,000	2
	0,09	78	24,7100	24,7100	0,000	60	0,09	11	24,7100	24,7100	0,000	1
	0,1	89	24,9000	24,9000	0,000	100	0,1	15	24,9000	24,9000	0,000	3
	0,2	125	26,2000	26,2000	0,000	271	0,2	15	26,2000	26,2000	0,000	3
	0,3	218	27,3000	27,3000	0,000	2.294	0,3	12	27,3000	27,3000	0,000	3
	0,4	427	28,4000	28,4000	0,000	19.095	0,4	14	28,4000	28,4000	0,000	4
	0,5			*			0,5	14	29,5000	29,5000	0,000	4
	0,6			*			0,6	14	30,6000	30,6000	0,000	3
	0,7			*			0,7	13	31,6000	31,6000	0,000	4
	0,8			*			0,8	15	32,4000	32,4000	0,000	5
	0,9			*			0,9	16	33,2000	33,2000	0,000	4

Tabela 5.3: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 14 nós

Inst	Cong	GDB					OA					
		It	LB	UP	GAP	T(s)	Cong	It	LB	UB	GAP	T(s)
14	0,001	13	21,0150	21,0150	0,000	1	0,001	7	21,0150	21,0150	0,000	0
	0,002	13	21,0300	21,0300	0,000	2	0,002	7	21,3000	21,3000	0,000	1
	0,003	11	21,0450	21,0450	0,000	1	0,003	7	21,0450	21,0450	0,000	1
	0,004	15	21,0600	21,0600	0,000	2	0,004	7	21,0600	21,0600	0,000	1
	0,005	13	21,0750	21,0750	0,000	3	0,005	7	21,0750	21,0750	0,000	1
	0,006	12	21,0900	21,0900	0,000	1	0,006	8	21,0900	21,0900	0,000	1
	0,007	13	21,1050	21,1050	0,000	2	0,007	7	21,1050	21,1050	0,000	0
	0,008	12	21,1200	21,1200	0,000	2	0,008	8	21,1200	21,1200	0,000	1
	0,009	12	21,1350	21,1350	0,000	2	0,009	8	21,1350	21,1350	0,000	0
	0,01	14	21,1500	21,1500	0,000	2	0,01	7	21,1500	21,1500	0,000	0
	0,02	17	21,3000	21,3000	0,000	4	0,02	6	21,3000	21,3000	0,000	0
	0,03	14	21,4500	21,4500	0,000	3	0,03	7	21,4500	21,4500	0,000	0
	0,04	14	21,6000	21,6000	0,000	2	0,04	6	21,6000	21,6000	0,000	1
	0,05	17	21,7500	21,7500	0,000	6	0,05	8	21,7500	21,7500	0,000	1
	0,06	25	21,9000	21,9000	0,000	9	0,06	9	21,9000	21,9000	0,000	1
	0,07	24	22,0500	22,0500	0,000	12	0,07	11	22,0500	22,0500	0,000	2
	0,08	27	22,2000	22,2000	0,000	13	0,08	7	22,2000	22,2000	0,000	1
	0,09	32	22,3500	22,3500	0,000	25	0,09	16	29,4000	29,4000	0,000	8
	0,1	32	22,5000	22,5000	0,000	20	0,1	8	22,5000	22,5000	0,000	1
	0,2	155	24,0000	24,0000	0,000	2.1888	0,2	8	24,0000	24,0000	0,000	2
	0,3	501	25,5000	25,5000	0,000	48.721	0,3	19	25,5000	25,5000	0,000	9
	0,4			*			0,4	18	26,4000	26,4000	0,000	8
	0,5			*			0,5	24	27,0000	27,0000	0,000	13
	0,6			*			0,6	22	27,6000	27,6000	0,000	7
	0,7			*			0,7	23	28,2000	28,2000	0,000	5
	0,8			*			0,8	23	28,8000	28,8000	0,000	5
	0,9			*			0,9	16	29,4000	29,4000	0,000	8

Tabela 5.4: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 15 nós

Inst	Cong	GDB					OA					
		It	LB	UP	GAP	T(s)	Cong	It	LB	UB	GAP	T(s)
15	0,001	10	22,0300	22,0300	0,000	3	0,001	5	22,0300	22,0300	0,000	1
	0,002	11	22,0600	22,0600	0,000	4	0,002	5	22,0600	22,0600	0,000	1
	0,003	12	22,0900	22,0900	0,000	3	0,003	5	22,0900	22,0900	0,000	0
	0,004	11	22,1800	22,1800	0,000	2	0,004	5	22,1200	22,1200	0,000	1
	0,005	15	22,1500	22,1500	0,000	4	0,005	5	22,1500	22,1500	0,000	1
	0,006	11	22,1800	22,1800	0,000	3	0,006	5	22,1800	22,1800	0,000	1
	0,007	11	22,2100	22,2100	0,000	4	0,007	5	22,2100	22,2100	0,000	1
	0,008	13	22,2400	22,2400	0,000	5	0,008	5	22,2400	22,2400	0,000	0
	0,009	9	22,2700	22,2700	0,000	2	0,009	5	22,2700	22,2700	0,000	0
	0,01	7	22,3000	22,3000	0,000	2	0,01	5	22,3000	22,3000	0,000	0
	0,02	8	22,6000	22,6000	0,000	2	0,02	5	22,6000	22,6000	0,000	1
	0,03	10	22,900	22,9000	0,000	4	0,03	5	22,9000	22,9000	0,000	0
	0,04	14	23,2000	23,2000	0,000	6	0,04	6	23,2000	23,2000	0,000	1
	0,05	18	23,5000	23,5000	0,000	7	0,05	8	23,5000	23,5000	0,000	0
	0,06	27	23,8000	23,8000	0,000	19	0,06	6	23,8000	23,8000	0,000	1
	0,07	29	24,0500	24,0500	0,000	30	0,07	8	24,0500	24,0500	0,000	1
	0,08	42	24,2000	24,2000	0,000	63	0,08	9	24,2000	24,2000	0,000	2
	0,09	30	24,3500	24,3500	0,000	18	0,09	8	24,3500	24,3500	0,000	3
	0,1	50	24,5000	24,5000	0,000	104	0,1	10	24,5000	24,5000	0,000	2
	0,2	161	26,0000	26,0000	0,000	5.658	0,2	9	26,0000	26,0000	0,000	4
	0,3			*			0,3	10	27,5000	27,5000	0,000	3
	0,4			*			0,4	11	28,8000	28,8000	0,000	3
	0,5			*			0,5	10	27,5000	27,5000	0,000	3
	0,6			*			0,6	18	31,2000	31,2000	0,000	15
	0,7			*			0,7	20	31,9000	31,9000	0,000	20
	0,8			*			0,8	19	32,6000	32,6000	0,000	12
	0,9			*			0,9	19	33,3000	33,3000	0,000	12

Tabela 5.5: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 16 nós

Inst	Cong	GDB					OA					
		It	LB	UP	GAP	T(s)	Cong	It	LB	UB	GAP	T(s)
16	0,001	9	26,0560	26,0560	0,000	1	0,001	6	26,0560	26,0560	0,000	0
	0,002	7	26,1120	26,1120	0,000	2	0,002	5	26,1120	26,1120	0,000	1
	0,003	8	26,1680	26,1860	0,000	2	0,003	5	26,1680	26,1680	0,000	1
	0,004	6	26,2240	26,2240	0,000	2	0,004	6	26,2240	26,2240	0,000	1
	0,005	9	26,2800	26,2800	0,000	2	0,005	5	26,2800	26,2800	0,000	0
	0,006	7	26,3360	26,3360	0,000	2	0,006	5	26,3360	26,3360	0,000	1
	0,007	9	26,3920	26,3920	0,000	2	0,007	5	26,3920	26,3920	0,000	1
	0,008	6	26,4480	26,4480	0,000	1	0,008	7	26,4480	26,4480	0,000	1
	0,009	9	26,5040	26,5040	0,000	2	0,009	5	26,5040	26,5040	0,000	1
	0,01	10	26,5600	26,5600	0,000	3	0,01	6	26,5600	26,5600	0,000	1
	0,02	7	26,1120	26,1120	0,000	7	0,02	7	27,1200	27,1200	0,000	1
	0,03	8	26,1680	26,1680	0,000	2	0,03	10	27,6800	27,6800	0,000	3
	0,04	57	28,2400	26,2400	0,000	164	0,04	6	28,2400	28,2400	0,000	2
	0,05	90	28,6000	28,6000	0,000	584	0,05	5	28,2600	28,2600	0,000	3
	0,06	110	28,9000	28,9000	0,000	1.426	0,06	5	26,3360	26,3360	0,000	1
	0,07	164	29,2400	29,2400	0,000	4.778	0,07	11	29,2400	29,2400	0,000	6
	0,08	213	29,5600	29,5600	0,000	8.728	0,08	13	29,5600	29,5600	0,000	8
	0,09	317	29,8000	29,8000	0,000	28.935	0,09	16	29,8000	29,8000	0,000	14
	0,1	301	30,0000	30,0000	0,000	31.965	0,1	13	30,0000	30,0000	0,000	10
	0,2			*			0,2	17	32,0000	32,0000	0,000	19
	0,3			*			0,3	24	34,0000	34,0000	0,000	44
	0,4			*			0,4	25	35,4000	35,4000	0,000	44
	0,5			*			0,5	26	36,5000	36,5000	0,000	52
	0,6			*			0,6	25	37,6000	37,6000	0,000	55
	0,7			*			0,7	33	38,6000	38,6000	0,000	83
	0,8			*			0,8	32	39,4000	39,4000	0,000	99
	0,9			*			0,9	30	40,2000	40,2000	0,000	67

Tabela 5.6: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 17 nós

Inst	Cong	GDB					OA					
		It	LB	UP	GAP	T(s)	Cong	It	LB	UB	GAP	T(s)
17	0,001	20	28,0350	28,0350	0,000	7	0,001	8	28,0350	28,0350	0,000	5
	0,002	18	28,0700	28,0700	0,000	9	0,002	10	28,0700	28,0700	0,000	7
	0,003	22	28,1050	28,1050	0,000	10	0,003	8	28,1050	28,1050	0,000	8
	0,004	19	28,1400	28,1400	0,000	7	0,004	8	28,1400	28,1400	0,000	8
	0,005	24	28,1750	28,1750	0,000	13	0,005	7	28,1750	28,1750	0,000	5
	0,006	19	28,2100	28,2100	0,000	8	0,006	8	28,2100	28,2100	0,000	4
	0,007	17	28,2450	28,2450	0,000	9	0,007	9	28,2450	28,2450	0,000	7
	0,008	22	28,2800	28,2800	0,000	11	0,008	8	28,2800	28,2800	0,000	4
	0,009	22	28,3150	28,3150	0,000	13	0,009	8	28,3150	28,3150	0,000	4
	0,01	18	28,3500	28,3500	0,000	9	0,01	8	28,3500	28,3500	0,000	4
	0,02	24	28,7000	28,7000	0,000	16	0,02	7	28,7000	28,7000	0,000	6
	0,03	26	29,0500	29,0500	0,000	16	0,03	8	29,0500	29,0500	0,000	8
	0,04	35	29,4000	29,4000	0,000	31	0,04	8	29,4000	29,4000	0,000	9
	0,05	56	29,7500	29,7500	0,000	65	0,05	12	29,7500	29,7500	0,000	13
	0,06	96	30,1000	30,1000	0,000	179	0,06	12	30,1000	30,1000	0,000	18
	0,07	106	30,3300	30,3300	0,000	197	0,07	16	30,3300	30,3300	0,000	23
	0,08	127	30,5200	30,5200	0,000	552	0,08	14	30,5200	30,5200	0,000	17
	0,09	187	30,7100	30,7100	0,000	4.171	0,09	17	30,7100	30,7100	0,000	27
	0,1	170	30,900	30,9000	0,000	2.297	0,1	13	30,9000	30,9000	0,000	10
	0,2			*			0,2	15	32,2000	32,2000	0,000	31
	0,3			*			0,3	16	33,3000	33,3000	0,000	21
	0,4			*			0,4	14	34,4000	34,4000	0,000	21
	0,5			*			0,5	17	35,5000	35,5000	0,000	33
	0,6			*			0,6	16	36,6000	36,6000	0,000	30
	0,7			*			0,7	17	37,7000	37,7000	0,000	53
	0,8			*			0,8	18	38,8000	38,8000	0,000	55
	0,9			*			0,9	18	39,9000	39,9000	0,000	43

Tabela 5.7: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 18 nós

Inst	Cong	GDB					OA					
		It	LB	UP	GAP	T(s)	Cong	It	LB	UB	GAP	T(s)
18	0,001	11	25,0310	25,0310	0,000	4	0,001	10	34,0550	34,0550	0,000	7
	0,002	18	25,0620	25,0620	0,000	10	0,002	10	34,1100	34,1100	0,000	10
	0,003	19	25,0930	25,0930	0,000	11	0,003	9	34,1650	34,1650	0,000	6
	0,004	15	25,1240	25,1240	0,000	7	0,004	9	34,2200	34,2200	0,000	6
	0,005	17	25,1550	25,1550	0,000	9	0,005	10	34,2750	34,2750	0,000	6
	0,006	15	25,1860	25,1860	0,000	8	0,006	9	34,3300	34,3300	0,000	7
	0,007	18	25,2170	25,2170	0,000	10	0,007	9	34,3850	34,3850	0,000	6
	0,008	33	25,2480	28,2800	0,000	90	0,008	9	34,4400	34,4400	0,000	8
	0,009	15	25,2790	25,2790	0,000	7	0,009	10	34,4950	34,4950	0,000	13
	0,01	32	25,3100	25,3100	0,000	100	0,01	10	34,5500	34,5500	0,000	8
	0,02	32	25,3100	25,3100	0,000	100	0,02	13	35,1000	35,1000	0,000	10
	0,03	32	25,3100	25,3100	0,000	100	0,03	8	35,6500	35,6500	0,000	11
	0,04	32	26,2400	26,2400	0,000	63	0,04	12	36,1200	36,1200	0,000	13
	0,05	36	26,5500	26,5500	0,000	80	0,05	8	36,4000	36,4000	0,000	12
	0,06	52	26,8600	26,8600	0,000	197	0,06	10	36,6800	36,6800	0,000	11
	0,07	84	27,1200	27,1200	0,000	717	0,07	10	36,9600	36,9600	0,000	19
	0,08	113	27,2800	27,2800	0,000	1.288	0,08	15	37,2400	37,2400	0,000	116
	0,09	106	27,4400	27,4400	0,000	1.021	0,09	13	37,5000	37,5000	0,000	34
	0,1	126	27,6000	27,6000	0,000	1.775	0,1	14	37,8000	37,8000	0,000	37
	0,2			*			0,2	16	40,0000	40,0000	0,000	52
	0,3			*			0,3	16	42,0000	42,0000	0,000	117
	0,4			*			0,4	18	43,8000	43,8000	0,000	202
	0,5			*			0,5	20	45,0000	45,8000	0,000	290
	0,6			*			0,6	19	46,2000	46,2000	0,000	138
	0,7			*			0,7	20	47,4000	47,4000	0,000	338
	0,8			*			0,8	23	48,6000	48,6000	0,000	161
	0,9			*			0,9	22	49,8000	49,8000	0,000	98

Tabela 5.8: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 20 nós

Inst	Cong	GDB					OA					
		It	LB	UP	GAP	T(s)	Cong	It	LB	UB	GAP	T(s)
20	0,001	54	25,0140	25,0140	0,000	45	0,001	10	32,0350	32,0350	0,000	120
	0,002	39	25,0280	25,0280	0,000	31	0,002	12	32,0700	32,0700	0,000	74
	0,003	62	25,0420	25,0420	0,000	120	0,003	12	32,1050	32,1050	0,000	121
	0,004	43	25,0560	25,0560	0,000	16	0,004	11	32,1400	32,1400	0,000	275
	0,005	46	25,0700	25,0700	0,000	27	0,005	10	32,1750	32,1750	0,000	110
	0,006	46	25,0700	25,0700	0,000	27	0,006	12	32,2100	32,2100	0,000	93
	0,007	47	25,0980	25,0980	0,000	43	0,007	12	32,2450	32,2450	0,000	97
	0,008	45	25,1120	25,1120	0,000	19	0,008	12	32,2800	32,2800	0,000	87
	0,009	41	25,1260	25,1260	0,000	12	0,009	13	32,3150	32,3150	0,000	99
	0,01	37	25,1400	25,1400	0,000	10	0,01	12	32,3500	32,3500	0,000	78
	0,02	43	25,2800	25,2800	0,000	28	0,02	13	32,7000	32,7000	0,000	85
	0,03	54	25,4200	25,4200	0,000	144	0,03	12	33,0500	33,0500	0,000	106
	0,04	53	25,5600	25,5600	0,000	60	0,04	15	33,4000	33,4000	0,000	200
	0,05	63	25,7000	25,7000	0,000	167	0,05	16	33,7000	33,7000	0,000	411
	0,06	74	25,8400	25,8400	0,000	157	0,06	18	33,7500	33,7500	0,000	280
	0,07	82	25,9800	25,9800	0,000	210	0,07	19	34,3300	34,3300	0,000	455
	0,08	133	26,1200	26,1200	0,000	3.094	0,08	20	34,5200	34,5200	0,000	307
	0,09	135	26,2600	26,2600	0,000	3.130	0,09	19	34,7100	34,7100	0,000	336
	0,1	156	26,4000	26,4000	0,000	10.665	0,1	21	34,9000	34,9000	0,000	581
	0,2			*			0,2	17	36,2000	36,2000	0,000	468
	0,3			*			0,3	16	37,3000	37,3000	0,000	501
	0,4			*			0,4	18	38,4000	38,4000	0,000	325
	0,5			*			0,5	20	39,5000	39,5000	0,000	620
	0,6			*			0,6	16	40,6000	40,6000	0,000	403
	0,7			*			0,7	21	41,7000	41,7000	0,000	335
	0,8			*			0,8	23	42,8000	42,8000	0,000	320
	0,9			*			0,9	22	43,9000	43,9000	0,000	595

Tabela 5.9: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 21 nós

Inst	Cong	GDB					OA					
		It	LB	UP	GAP	T(s)	Cong	It	LB	UB	GAP	T(s)
21	0,001	38	28,0140	28,0140	0,000	19	0,001	8	43,0570	43,0570	0,000	2
	0,002	33	28,0280	28,0280	0,000	21	0,002	8	43,1160	43,1160	0,000	6
	0,003	33	28,0280	28,0280	0,000	22	0,003	14	43,1710	43,1710	0,000	13
	0,004	39	28,0560	28,0560	0,000	20	0,004	8	43,2280	43,2280	0,000	6
	0,005	47	28,0700	28,0700	0,000	59	0,005	9	43,2850	43,2850	0,000	3
	0,006	33	28,0840	28,0840	0,000	12	0,006	11	43,3420	43,3420	0,000	11
	0,007	33	28,0840	28,0840	0,000	13	0,007	10	43,3990	43,3990	0,000	5
	0,008	33	28,0840	28,0840	0,000	12	0,008	9	43,4560	43,4560	0,000	6
	0,009	38	28,1260	28,1260	0,000	36	0,009	8	43,5130	43,5130	0,000	4
	0,01	42	28,1400	28,1400	0,000	136	0,01	10	43,5700	43,5700	0,000	9
	0,02	34	28,2800	28,2800	0,000	20	0,02	8	44,1400	44,1400	0,000	10
	0,03	36	28,4200	28,4200	0,000	69	0,03	9	44,7100	44,7100	0,000	21
	0,04	36	28,4200	28,4200	0,000	69	0,04	10	45,2800	45,2800	0,000	25
	0,05	54	28,700	28,7000	0,000	119	0,05	10	45,8500	45,8500	0,000	43
	0,06	55	28,7000	28,7000	0,000	150	0,06	15	46,4200	46,4200	0,000	119
	0,07	59	28,8400	28,8400	0,000	209	0,07	16	46,9900	46,9900	0,000	103
	0,08	69	29,1200	29,1200	0,000	291	0,08	18	47,5200	47,5200	0,000	466
	0,09	92	29,6000	29,6000	0,000	1.013	0,09	24	47,9600	47,9600	0,000	395
	0,1	92	29,6000	29,6000	0,000	1.129	0,1	22	48,4000	48,4000	0,000	571
	0,2			*			0,2	27	50,8000	50,8000	0,000	1.235
	0,3			*			0,3	29	52,7000	52,7000	0,000	1.130
	0,4			*			0,4	31	54,6000	54,6000	0,000	1.019
	0,5			*			0,5	31	56,5000	56,5000	0,000	1.247
	0,6			*			0,6	36	58,4000	58,4000	0,000	3.424
	0,7			*			0,7	46	59,7000	59,7000	0,000	2.108
	0,8			*			0,8	34	60,8000	60,8000	0,000	2.171
	0,9			*			0,9	41	61,9000	61,9000	0,000	1.142

Tabela 5.10: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 22 nós

Inst	Cong	GDB					OA					
		It	LB	UP	GAP	T(s)	Cong	It	LB	UB	GAP	T(s)
22	0,001	120	32,0050	32,0050	0,000	80	0,001	12	33,0050	33,0050	0,000	1
	0,002	119	32,0100	32,0100	0,000	119	0,002	12	32,0100	32,0100	0,000	2
	0,003	150	32,0150	32,0150	0,000	217	0,003	11	32,0150	32,0150	0,000	1
	0,004	120	32,0200	32,0200	0,000	99	0,004	12	32,0200	32,0200	0,000	1
	0,005	112	32,0250	32,0250	0,000	195	0,005	12	32,0250	32,0250	0,000	1
	0,006	102	32,0300	32,0300	0,000	74	0,006	11	32,0300	32,0300	0,000	2
	0,007	115	32,0350	32,0350	0,000	74	0,007	12	32,0350	32,0350	0,000	1
	0,008	114	32,0400	32,0400	0,000	81	0,008	12	32,0400	32,0400	0,000	2
	0,009	105	32,0450	32,0450	0,000	105	0,009	11	32,0450	32,0450	0,000	1
	0,01	117	32,0500	32,0500	0,000	67	0,01	12	32,0500	32,0500	0,000	2
	0,02	128	32,1000	32,1000	0,000	110	0,02	12	32,1000	32,1000	0,000	1
	0,03	116	32,1500	32,1500	0,000	308	0,03	10	32,1500	32,1500	0,000	1
	0,04	121	32,2000	32,2000	0,000	335	0,04	10	32,2000	32,2000	0,000	1
	0,05	126	32,2500	32,2500	0,000	157	0,05	10	32,2500	32,2500	0,000	1
	0,06	141	32,3000	32,3000	0,000	759	0,06	10	32,3000	32,3000	0,000	1
	0,07	141	32,3000	32,3000	0,000	924	0,07	10	32,3500	32,3500	0,000	1
	0,08	135	32,4000	32,4000	0,000	600	0,08	10	32,4000	32,4000	0,000	1
	0,09	137	32,4500	32,4500	0,000	301	0,09	10	32,4500	32,4500	0,000	2
	0,1	145	32,5000	32,5000	0,000	841	0,1	10	32,5000	32,5000	0,000	1
	0,2	265	33,0000	33,0000	0,000	13.146	0,2	11	33,0000	33,0000	0,000	1
	0,3			*			0,3	11	33,5000	33,5000	0,000	2
	0,4			*			0,4	11	34,0000	34,0000	0,000	1
	0,5			*			0,5	12	34,5000	34,5000	0,000	1
	0,6			*			0,6	14	35,0000	35,0000	0,000	3
	0,7			*			0,7	14	35,5000	35,5000	0,000	3
	0,8			*			0,8	19	36,0000	36,0000	0,000	5
	0,9			*			0,9	20	36,5000	36,5000	0,000	5

Tabela 5.11: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 24 nós

Inst	Cong	GDB					OA					
		It	LB	UP	GAP	T(s)	Cong	It	LB	UB	GAP	T(s)
24	0,001	79	29,0140	29,0140	0,000	67	0,001	9	33,0310	33,0310	0,000	105
	0,002	214	23,0100	23,0100	0,000	260	0,002	10	33,0620	33,0620	0,000	94
	0,003	225	23,0150	23,0150	0,000	336	0,003	7	33,0393	33,0393	0,000	72
	0,004	227	23,0200	23,0200	0,000	465	0,004	8	33,1240	33,1240	0,000	86
	0,005	230	23,0250	23,0250	0,000	437	0,005	9	33,1550	33,1550	0,000	58
	0,006	206	23,0300	23,0300	0,000	245	0,006	9	33,1860	33,1860	0,000	73
	0,007	234	23,0350	23,0350	0,000	329	0,007	11	33,2170	33,2170	0,000	99
	0,008	227	23,0400	23,0400	0,000	401	0,008	8	33,2480	33,2480	0,000	72
	0,009	208	23,0450	23,0450	0,000	339	0,009	9	33,2790	33,2790	0,000	93
	0,01	207	23,0500	23,0500	0,000	268	0,01	12	33,3100	33,3100	0,000	67
	0,02	226	23,1000	23,1000	0,000	280	0,02	9	33,6200	33,6200	0,000	69
	0,03	207	23,1500	23,1500	0,000	343	0,03	12	33,9300	33,9300	0,000	64
	0,04	228	23,2000	23,2000	0,000	708	0,04	11	34,2400	34,2400	0,000	96
	0,05	212	23,2500	23,2500	0,000	1.167	0,05	10	34,5500	34,5500	0,000	78
	0,06	230	23,3000	23,3000	0,000	1.388	0,06	10	34,8600	34,8600	0,000	63
	0,07	253	23,3500	23,3500	0,000	1.480	0,07	12	35,1700	35,1700	0,000	139
	0,08	249	23,4000	23,4000	0,000	2.709	0,08	9	35,4800	35,4800	0,000	110
	0,09	254	23,4500	23,4500	0,000	3.095	0,09	11	35,7900	35,7900	0,000	133
	0,1	254	23,5000	23,5000	0,000	3.548	0,1	12	36,1000	36,1000	0,000	157
	0,2			*			0,2	19	38,0000	38,0000	0,000	504
	0,3			*			0,3	26	39,5000	39,5000	0,000	483
	0,4			*			0,4	24	40,8000	40,8000	0,000	396
	0,5			*			0,5	26	42,0000	42,0000	0,000	785
	0,6			*			0,6	29	43,2000	43,2000	0,000	2.265
	0,7			*			0,7	33	44,4000	44,4000	0,000	789
	0,8			*			0,8	37	45,6000	45,6000	0,000	826
	0,9			*			0,9	33	46,3000	46,3000	0,000	1.767

Tabela 5.12: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 25 nós

Inst	Cong	GDB					OA					
		It	LB	UP	GAP	T(s)	Cong	It	LB	UB	GAP	T(s)
25	0,001	186	22,0050	22,0050	0,000	263	0,001	7	29,0300	29,0300	0,000	3
	0,002	171	22,0100	22,0100	0,000	385	0,002	8	29,0600	29,0600	0,000	5
	0,003	179	22,0150	22,0150	0,000	654	0,003	8	29,0900	29,0900	0,000	9
	0,004	190	22,0200	22,0200	0,000	244	0,004	8	29,1200	29,1200	0,000	5
	0,005	173	22,0250	23,0250	0,000	308	0,005	8	29,1500	29,1500	0,000	2
	0,006	194	22,0300	22,0300	0,000	371	0,006	10	29,1800	29,1800	0,000	7
	0,007	184	23,0350	23,0350	0,000	967	0,007	9	29,2100	29,2100	0,000	4
	0,008	198	22,0400	22,0400	0,000	376	0,008	8	29,2400	29,2400	0,000	4
	0,009	205	22,0450	22,0450	0,000	1.230	0,009	9	29,2700	29,2700	0,000	14
	0,01	169	22,0500	22,0500	0,000	383	0,01	10	29,3000	29,3000	0,000	4
	0,02	184	22,1000	22,1000	0,000	892	0,02	8	29,6000	29,6000	0,000	3
	0,03	190	22,1500	22,1500	0,000	580	0,03	9	29,9000	29,9000	0,000	5
	0,04	200	22,2000	22,2000	0,000	544	0,04	9	30,2000	30,2000	0,000	36
	0,05	201	22,2500	22,2500	0,000	1.544	0,05	10	30,5000	30,5000	0,000	57
	0,06	176	22,3000	22,3000	0,000	1.316	0,06	10	30,8000	30,8000	0,000	16
	0,07	208	22,3500	23,3500	0,000	2.242	0,07	12	31,1000	31,1000	0,000	23
	0,08	224	22,4000	22,4000	0,000	2.396	0,08	11	31,4000	31,4000	0,000	30
	0,09	224	22,4500	22,4500	0,000	1.790	0,09	11	31,7000	31,7000	0,000	205
	0,1	211	22,5000	22,5000	0,000	3.908	0,1	12	32,0000	32,0000	0,000	40
	0,2			*			0,2	19	34,0000	34,0000	0,000	511
	0,3			*			0,3	20	35,5000	35,5000	0,000	1.045
	0,4			*			0,4	19	36,8000	36,8000	0,000	1.228
	0,5			*			0,5	24	38,0000	38,0000	0,000	1.285
	0,6			*			0,6	22	39,2000	39,2000	0,000	480
	0,7			*			0,7	26	40,4000	40,4000	0,000	8.348
	0,8			*			0,8	29	41,6000	41,6000	0,000	3.564
	0,9			*			0,9	28	42,3000	42,3000	0,000	6.595

Tabela 5.13: Comparativo entre as técnicas de GDB e OA: Fator de congestionamento, número de iterações, Limite Inferior, Limite Superior, GAP e tempo computacional para a instância de 27 nós

Inst	Cong	GDB					OA					
		It	LB	UP	GAP	T(s)	Cong	It	LB	UB	GAP	T(s)
27	0,001	333	29,0050	29,0050	0,000	2.603	0,001	8	44,0310	44,0310	0,000	4
	0,002	376	29,0100	29,0100	0,000	3.010	0,002	6	44,0620	44,0620	0,000	2
	0,003	339	29,0150	29,0150	0,000	1.856	0,003	7	44,0930	44,0930	0,000	3
	0,004	323	29,0200	29,0200	0,000	1.913	0,004	7	44,1240	44,1240	0,000	2
	0,005	328	29,0250	29,0250	0,000	1.578	0,005	8	44,1550	44,1550	0,000	3
	0,006	330	29,0300	29,0300	0,000	1.732	0,006	9	44,1860	44,1860	0,000	6
	0,007	313	29,0350	29,0350	0,000	2.876	0,007	9	44,2170	44,2170	0,000	5
	0,008	323	29,0400	29,0400	0,000	1.453	0,008	9	44,2480	44,2480	0,000	4
	0,009	352	29,0450	29,0450	0,000	3.362	0,009	9	44,2790	44,2790	0,000	4
	0,01	351	29,0500	29,0500	0,000	2.017	0,01	12	44,3100	44,3100	0,000	8
	0,02	325	29,1000	29,1000	0,000	3.566	0,02	9	44,6200	44,6200	0,000	4
	0,03	327	29,1500	29,15000	0,000	2.467	0,03	7	44,9300	44,9300	0,000	4
	0,04	307	29,2000	29,2000	3.177	3.177	0,04	7	45,2400	45,2400	0,000	6
	0,05	331	29,2500	29,2500	0,000	2.923	0,05	9	45,5500	45,5500	0,000	8
	0,06	360	29,3000	29,3000	0,000	5.152	0,06	8	45,8600	45,8600	0,000	10
	0,07	338	29,3500	29,3500	0,000	4.167	0,07	10	46,1700	46,1700	0,000	12
	0,08	357	29,4000	29,4000	0,000	9.535	0,08	11	46,4800	46,4800	0,000	16
	0,09	392	29,4500	29,4500	0,000	38.587	0,09	11	46,7900	46,7900	0,000	17
	0,1	438	29,5000	29,5000	0,000	21.304	0,1	11	47,1000	47,1000	0,000	68
	0,2			*			0,2	15	49,4000	49,4000	0,000	214
	0,3			*			0,3	22	51,6000	51,6000	0,000	3.294
	0,4			*			0,4	27	53,8000	53,8000	0,000	2.235
	0,5			*			0,5	38	56,0000	56,0000	0,000	2.847
	0,6			*			0,6	43	57,0000	57,0000	0,000	6.618
	0,7			*			0,7	43	58,0000	58,0000	0,000	8.046
	0,8			*			0,8	50	59,0000	59,0000	0,000	7.117
	0,9			*			0,9	46	60,0000	60,0000	0,000	6.618

# Capítulo 6

## Conclusões e Trabalhos Futuros

*"A mente que se abre a uma nova idéia jamais voltará ao seu tamanho original."*

Albert Einstein

Este trabalho apresentou duas técnicas diferentes para a resolução de problemas de grande escala sob efeito de congestionamento: A técnica GBD e o método de OA. Ambos foram capazes de resolver instâncias de até 27 nós (729 arcos) utilizando a linguagem AMPL. A AMPL é conhecida com uma linguagem de alto nível. Apesar disto, os resultados foram razoavelmente significativos, chegando a um tempo computacional de 6618 segundos para a técnica de OA com 27 nodos. Estes resultados podem ser melhorados utilizando em conjunto a linguagem C++ juntamente com o IBM CPLEX 11 Concert Technology, pois a linguagem C++ é uma linguagem de médio nível.

De modo geral a abordagem proposta para a solução de programas não lineares inteiros mistos, como o modelo de projeto de redes com custos não-lineares, se comporta bem e o esquema de decomposição de Benders associado à técnica de decomposição aproximação externa é bem sucedido ao tratar um problema de tais características. Isso acontece devido à grande redução do efeito cauda longa (Tail-Off) e também a grande utilidade dos cortes de Benders para melhorar a relaxação de programação linear do problema mestre.

Pesquisas futuras devem agregar cortes Pareto-Ótimos implementados via Branch-and-Cut e a medição do *"trade of"* entre o uso de restrições lineares de configuração de cortes de viabilidade.

# Referências Bibliográficas

- Balas, E. e Bergthaller, C. (1983). Benders' method revisited. *Journal of Computational and Applied Mathematics*, 9:3–12.
- Benders, J. (1962). Partitioning procedures for solving mixed integer variables programming problems. *Numerische Methematik*, 4:238–252.
- Carneiro, M. S. (1996). Long-range planning of power distribution systems: secondary networks. *Computers and Electrical Engineering*, 22:179–191.
- Claus, A. e Maculan, N. (1983). Une nouvelle formulation du probeme de steiner sur un graphe. Technical Report 280, Centre de Recherche sur les Tranports, Université de Montréal, Canada.
- Cordeau, J. e Desrosiers, J. (2000). A benders decomposition approach for the locomotive e car assignment problem. *Transportation Science*, 34:133–149.
- Cordeau, J. e Desrosiers, J. (2001). Simoultaneous assignment of locomotives and cars to passenger trains. *Operations Research*, 49(4):531–548.
- Dantzig, G. (1962). *Linear Programming and Extensions*. Princeton University Press,.
- Duran, M. e Grossman, I. (1986). An outer-approximation algorithm for a claar of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307.
- Fletcher, R. e Leyfer., S. (1994). Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66:327.
- Gavish, B. (1982). Topological design of centralized computer networks-formulations and algorithms. *Networks*, 12:355–377.
- Gavish, B. (1983). Formulations and algorithms for the capacitated minimal directed tree problem. *Journal of the Association for Computing Machinery*, 30:118–132.
- Geoffrion, A. (1972a). Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10:237–260.

- Geoffrion, A. (1972b). Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10:237–260.
- Geoffrion, A. e Graves, G. (1974). Multicommodity distribution system design by benders decomposition. *Management Science*, 20:822–844.
- Goemans, M. e Mying, Y. (1993). A catalogue of steiner tree formulations. *Networks*, 23:19–28.
- Grossmann, I. e Kravanja, Z. (1995). Mixed-integer nonlinear programming techniques for process systems engineering. *Computers & Chemical Engineering*, 19:189–204.
- Gutierrez, G.; Kouvelis, P. e Kurawarwala, A. (1996). A robustness approach to uncapacitated network design problems. *European Journal of Operations Research*, 94:362–376.
- Hoang, H. (1982). Topological optimization of networks: a nonlinear mixed integer model employing generalized benders decomposition. *IEEE Transactions on Automatic Control*, 27:164–169.
- Huang, S.; Batta, R. e Nagi, R. (2005). Distribution network design: Selection and sizing of congested connections. *Naval Research Logistics*, 52(8):701–712.
- Hwang, F. e Richards, D. (1992a). Steiner tree problems. *Networks*, 22:55–89.
- Hwang, F. e Richards, D. (1992b). Steiner tree problems. *North-Holland*, 22:55–89.
- Karuppiah; Ramkumar; Furman, K. e Grossman, I. (2008). Global optimization for scheduling refinery crude oil operations. *Computers & Chemical Engineering*, 32(11):2745–2766.
- Maculan, N. (1986). A new linear programming formulation for the shortest s-directed spanning tree problem. *Journal of Combinatorics, Information e Systems Sciences*, 11:53–56.
- Maculan, N. (1987). The steiner problem in graphs. *Annals of Discrete Mathematics*, 31:185–212.
- Maculan, N.; Arpin, D. e Nguyen, S. (1988). Le problème de steiner sur un graphe orienté: Formulations et relaxations. *Matemática Aplicada e Computacional*, 7:109–118.

- Magnanti, T.; Marchandani, P. e Wong, R. (1986). Tailoring benders decomposition for uncapacitated network design. *Mathematical Programming Study*, 26:112–154.
- Magnanti, T. e Wong, R. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–483.
- McDaniel, D. e Devine, M. (1977). A modified benders' partitioning algorithm for mixed integer programming. *Management Science*, 24(3):312–319.
- Miranda Jr, G. (2004). *Facility Location and Network Design with Congestion Costs and Interdependency*. PhD thesis, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais.
- Queiroz, M. e Humes, C. (2001). The projected pairwise multicommodity flow polyhedron. *Applied Mathematics Letters*, 14(4):443–448.
- Radazzo, C. D.; Luna, H. e P., M. (2001). Benders decomposition for local access network design with two technologies. *Discrete Mathematics and Theoretical Computer Science*, 4:235–246.
- Randazzo, C. e Luna, H. (2001). A comparison of optimal methods for local access uncapacitated network design. *The Annals of Operations Research*, 106:263–286.
- Winter, P. (1987). Steiner problem in networks: a survey. *Networks*, 17:129–167.
- Wong, R. (1984). A dual ascent algorithm for steiner problem in direct graphs. *Mathematical Programming*, 28:271–287.
- Yuan, X.; Zhang, S.; Piboleau, S. e Domenech, S. (1988). Une methode d'optimisation nonlineare an variables pour la conception de procedes. *Operations Research*, 33:331.