

ELISANGELA MARTINS DE SÁ

**LOCALIZAÇÃO DE CONCENTRADORES  
APLICADA AO TRANSPORTE PÚBLICO**

Belo Horizonte  
07 de fevereiro de 2011

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

**LOCALIZAÇÃO DE CONCENTRADORES  
APLICADA AO TRANSPORTE PÚBLICO**

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

ELISANGELA MARTINS DE SÁ

Belo Horizonte  
07 de fevereiro de 2011



UNIVERSIDADE FEDERAL DE MINAS GERAIS

## FOLHA DE APROVAÇÃO

Localização de concentradores aplicada ao transporte público

ELISANGELA MARTINS DE SÁ

Dissertação defendida e aprovada pela banca examinadora constituída por:

Ph. D. RICARDO SARAIVA DE CAMARGO – Orientador  
Universidade Federal de Minas Gerais

Ph. D. GILBERTO DE MIRANDA JÚNIOR – Co-orientador  
Universidade Federal de Minas Gerais

Ph. D. MAURÍCIO CARDOSO DE SOUZA  
Universidade Federal de Minas Gerais

Ph. D. RODNEY REZENDE SALDANHA  
Universidade Federal de Minas Gerais

Belo Horizonte, 07 de fevereiro de 2011

# Resumo

Um dos maiores problemas que os grandes centros urbanos enfrentam atualmente é o congestionamento no trânsito. Uma alternativa para solucionar esse problema seria uma maior utilização de meios de transporte público como trens, metrô, fura-fila e etc. No entanto, devido ao alto custo para se instalar conexões diretas entre todos os pares de pontos origem-destino, a rede de transporte público é modelada como uma rede do tipo eixo-raio. Neste trabalho um modelo baseado na formulação proposta por [Contreras et al. \(2009\)](#) é utilizado para modelar o transporte público. Devido as características da formulação usada neste trabalho, o método de decomposição de Benders ([Benders, 1962](#)) é aplicado para resolver o problema. Resultados computacionais mostram que a versão clássica de Benders só consegue resolver problemas de médio porte. Por isso, três variantes do método de decomposição de Benders que possuem um esquema de seleção de cortes que tende a aumentar a convergência do método são apresentadas. Além disso, uma nova técnica para melhorar a geração de cortes, nomeada  $\lambda$ -ótimo, é proposta. Resultados computacionais demonstram que esta nova técnica apresenta o melhor desempenho acelerando a convergência do método.

**Palavras-chaves:** Redes eixo-raio; Método de Decomposição de Benders; Transporte Público.

# Abstract

One of the biggest problems that large cities have faced nowadays is traffic congestion. An alternative to solve this problem would be greater use of public transport like trains, subways, stick-row and so on. However, due to the high cost to install direct connections between all pairs of origin-destination points, modeled the urban transport network as a network-type hub-and-spoke. In this work a model based on the formulation proposed by [Contreras et al. \(2009\)](#) is used to model public transport. Due to the characteristics of the formulation used in this work, the Benders decomposition method ([Benders, 1962](#)) is applied to solve the problem. Computational results show that the classic version of Benders can only solve small problems. Therefore, three variants of Benders decomposition scheme that possesses a selection of cuts that tend to increase the convergence of the method are presented. In addition, a new technique to improve the generation of cuts, namely  $\lambda$ -optimal, is proposed. Computational results show that this new technique has the best performance by accelerating the convergence of the method.

Keywords: Hub-and-spoke networks; Benders decomposition; Public transportation;

*Aos meus pais Fernando e Léa;  
Aos meus irmãos, meus tesouros;*

*“Não deixe que a saudade sufoque, que a rotina acomode, que o medo impeça de tentar. Desconfie do destino e acredite em você. Gaste mais horas realizando que sonhando, fazendo que planejando, vivendo que esperando, porque, embora quem quase morre esteja vivo, quem quase vive já morreu.” (Luiz Fernando Veríssimo)*

# Agradecimentos

Eu gostaria de agradecer primeiramente a **Deus**, meu grande parceiro.

Ao Prof. Ricardo Saraiva de Camargo, meu orientador, pela sua infinita paciência, pela motivação e por acreditar em meu trabalho.

Ao meu co-orientador, Gilberto de Miranda Júnior, pelas valiosas conversas e sugestões, que contribuíram significativamente para o desenvolvimento deste trabalho.

Ao professor, Ricardo H. C. Takahashi, pelo grande incentivo e apoio para entrar no mestrado.

À CAPES pelo apoio financeiro que foi imprescindível para a dedicação integral ao mestrado.

Aos meus amigos e à minha família, pelo grande apoio e pela paciência nos momentos em que me ausentei para me dedicar a este trabalho.

Aos professores, funcionários e colegas do Departamento de Engenharia de Produção.

# Sumário

Nomenclatura	2
<b>1 Introdução</b>	<b>3</b>
<b>2 Exame da Literatura</b>	<b>7</b>
2.1 Localização de Concentradores . . . . .	7
2.2 Aplicações . . . . .	9
2.3 Localização de Concentradores Aplicada ao Transporte Público . . . . .	10
2.4 Método de Decomposição de Benders . . . . .	17
<b>3 Localização de Concentradores Aplicada ao Transporte Público</b>	<b>22</b>
3.1 Definições e Formulação . . . . .	22
3.2 Método de Decomposição de Benders . . . . .	24
3.2.1 Problema Mestre e Subproblema . . . . .	24
3.2.2 Algoritmo de Decomposição de Benders Clássico . . . . .	29
3.3 Resultados Computacionais . . . . .	30
3.4 Comentários . . . . .	42
<b>4 Variantes do Método de Decomposição de Benders</b>	<b>44</b>
4.1 Geração de Cortes Pareto-Ótimo . . . . .	45
4.1.1 Subproblema Independente de Magnanti-Wong . . . . .	47
4.1.2 Algoritmo de Decomposição de Benders Usando Cortes Pareto-Ótimo . . . . .	48
4.2 Subsistema de Inviabilidade Mínimo . . . . .	49
4.2.1 Subproblema para Gerar o MIS . . . . .	51
4.2.2 Algoritmo de Decomposição de Benders Usando MIS . . . . .	51
4.3 Sistema de Viabilidade Máximo . . . . .	52
4.3.1 Subproblema para Gerar o MFS . . . . .	53
4.3.2 Algoritmo de Decomposição de Benders Usando o MFS . . . . .	55
4.4 Seleção de Cortes $\lambda$ -Ótimo . . . . .	55

4.4.1	Subproblema para Cálculo do $\lambda$ -ótimo . . . . .	57
4.4.2	Algoritmo de Decomposição de Benders Usando $\lambda$ -Ótimo . . . . .	57
<b>5</b>	<b>Resultados Computacionais</b>	<b>59</b>
<b>6</b>	<b>Conclusão</b>	<b>73</b>
	<b>Referências Bibliográficas</b>	<b>75</b>

# Lista de Figuras

1.1	(a) Conexões entre pares origem/destino. (b) Rede eixo-raio. . . . .	4
1.2	(a) Alocação Simples (b) Alocação Múltipla . . . . .	4
1.3	Localização de concentradores em árvore. . . . .	5
3.1	Desempenho computacional das versões do método de decomposição de Benders para instâncias AP10. . . . .	33
3.2	Desempenho computacional das versões do método de decomposição de Benders para instâncias AP20. . . . .	33
3.3	Proporção do tempo gasto pela variante Benders com PP e B&C para executar o PM e SP. . . . .	34
3.4	Proporção do tempo gasto pela variante Benders com PP para executar o PM e SP. . . . .	34
3.5	Proporção do tempo gasto pela variante Benders com B&C para executar o PM e SP. . . . .	34
3.6	Evolução do limite inferior e do limite superior. . . . .	35
3.7	Esforço computacional versus a variação no fator de desconto. . . . .	36
3.8	Número de concentradores instalados versus a variação no fator de desconto. . . . .	36
3.9	Configuração da rede para $\alpha = 2$ . . . . .	37
3.10	Configuração da rede para $\alpha = 4$ . . . . .	37
3.11	Configuração da rede para $\alpha = 6$ . . . . .	37
3.12	Configuração da rede para $\alpha = 8$ . . . . .	37
3.13	Esforço computacional versus a variação nos custos de instalação. . . . .	38
3.14	Número de concentradores instalados versus a variação nos custos de instalação. . . . .	38
3.15	Configuração da rede para $ff = 50\%$ . . . . .	39
3.16	Configuração da rede para $ff = 100\%$ . . . . .	39
3.17	Configuração da rede para $ff = 200\%$ . . . . .	39
3.18	Configuração da rede para $ff = 300\%$ . . . . .	39
3.19	Esforço computacional usando instâncias CAB - CPLEX versus versões do Benders clássico. . . . .	40

3.20	Esforço computacional usando instâncias AP - CPLEX versus versões do Benders clássico. . . . .	41
3.21	Proporção média de cortes tipo I e de cortes tipo II. . . . .	42
5.1	Esforço computacional ao variar o número de iterações de pré-processamento para instância CAB15.2. . . . .	60
5.2	Quantidade de melhores de tempos de cada variantes com e sem restrições de AG. . . . .	65
5.3	Quantidade de melhores de tempos entre todas as versões de Benders. . . .	66
5.4	Esforço computacional usando instâncias CAB: CPLEX e Benders Clássico versus Benders CII. . . . .	66
5.5	Esforço computacional usando instâncias AP: CPLEX e Benders Clássico versus Benders CII. . . . .	66
5.6	Esforço computacional usando instâncias CAB sem restrições de AG. . . .	67
5.7	Esforço computacional usando instâncias CAB com restrições de AG. . . .	67
5.8	Esforço computacional usando instâncias AP sem restrições de AG. . . .	67
5.9	Esforço computacional usando instâncias AP com restrições de AG. . . .	67
5.10	Quantidade média de cortes para instâncias AP20. . . . .	68
5.11	Quantidade média de cortes para instâncias AP30. . . . .	68
5.12	Convergência dos limites usando a instância AP20.2. . . . .	69
5.13	Convergência dos limites usando a instância AP30.2. . . . .	69
5.14	Tempo total de execução dos PMs e dos subproblemas usando as instâncias AP20.2 e AP30.2. . . . .	69
5.15	Esforço computacional - variante Benders LO com e sem restrição de AG. .	70
5.16	Quantidade de cortes gerados pela variante Benders LO com e sem restrição de AG. . . . .	72

# Lista de Tabelas

3.1	Resultados computacionais da relaxação linear das formulações para o THL com custo fixo. . . . .	32
3.2	Resultados computacionais com e sem o uso das técnicas HS e B&C. . . . .	33
3.3	Resultados computacionais usando instâncias CAB - CPLEX versus versões do Benders clássico. . . . .	39
3.4	Resultados computacionais usando instâncias AP - CPLEX versus Benders clássico. . . . .	40
3.5	Número médio de iterações de pré-processamento e tempo médio gasto nas iterações inteiras em escala relativa. . . . .	41
5.1	Resultados computacionais dos algoritmos usando instâncias CAB sem restrições de AG. . . . .	61
5.2	Resultados computacionais dos algoritmos usando instâncias CAB com restrições de AG. . . . .	62
5.3	Resultados computacionais dos algoritmos usando instâncias AP sem restrições de AG. . . . .	63
5.4	Resultados computacionais dos algoritmos usando instâncias AP com restrições de AG. . . . .	64
5.5	Resultados computacionais usando instâncias AP - LO. . . . .	71

# Lista de Algoritmos

1	Algoritmo de decomposição de Benders clássico sem restrições de AG . . .	29
2	Algoritmo de decomposição de Benders clássico com restrições de AG . . .	30
3	Algoritmo de decomposição de Benders com cortes pareto-ótimo . . . . .	48
4	Algoritmo de decomposição de Benders com cortes pareto-ótimo usando restrições de AG . . . . .	49
5	Algoritmo de decomposição de Benders usando cortes baseados no MIS	52
6	Algoritmo de decomposição de Benders usando cortes baseados no MFS	55
7	Algoritmo de decomposição de Benders usando $\lambda$ -Ótimo . . . . .	58

# Nomenclatura

AG Árvore Geradora

AP Serviço Postal Australiano

CAB Conselho de Aviação Civil dos Estados Unidos da América

CPLEX Pacote comercial de Otimização

GPT Transporte Público Generalizado

LI Limite Inferior

LS Limite Superior

MFS Sistema de Viabilidade Máximo

MIS Subsistema de Inviabilidade Mínimo

PLCATP Problema de Localização de Concentradores Aplicada ao Transporte Público

PM Problema Mestre

PP Pré-processamento

PT Transporte Público

SIMW Subproblema Independente de Magnanti-Wong

SMW Subproblema de Magnanti-Wong

SP Subproblema

THL Problema de Localização em Árvore

# Capítulo 1

## Introdução

*“Dai-me, Senhor, a perseverança das ondas do mar, que fazem de cada recuo um ponto de partida para um novo avanço”*

Cecília Meireles

Um dos maiores problemas que os grandes centros urbanos enfrentam atualmente é o congestionamento no trânsito provocado tanto por veículos de transporte público (ônibus e taxis) quanto por veículos de transporte privado (carros e motos). Como várias pessoas necessitam se deslocar diariamente de um local para outro, uma alternativa para tornar o sistema de transporte público mais eficiente seria uma maior utilização de meios de transporte como trem, metrô, fura-fila, entre outros. No entanto, se torna inviável instalar conexões diretas entre todos os pares de pontos de origem e de destino devido ao alto custo de instalação dos terminais de acesso e dos corredores de deslocamento. Sendo assim, uma alternativa seria modelar a rede de transporte público como um rede do tipo eixo-raio, do inglês *hub-and-spoke network*.

Em redes do tipo eixo-raio, a demanda entre os pares de origem e de destino não é transportada diretamente. O fluxo de diferentes origens, mas endereçado ao mesmo destino, pode ser consolidado em pontos de transbordo, conhecidos como concentradores, antes de ser enviado, provavelmente através de outros concentradores, em direção a seu destino. Concentradores são então responsáveis pela agregação, roteamento e redistribuição do fluxo. A aglomeração do fluxo nos concentradores aumenta o tráfego em conexões entre concentradores, permitindo assim o uso de meios de transporte de maior volume e mais eficientes, resultando, portanto, em custos unitários de transporte menores (O’Kelly, 1998). Dessa forma, economias de escala podem ser alcançadas. As redes do tipo eixo-raio reduzem então significativamente o custo de instalação e gerenciamento da rede, além de permitir o usufruto da economia de escala ao consolidar os fluxos nos concentradores (Kara e Tansel, 2003).

A Figura 1.1 ilustra a diferença entre uma rede em que todos os pares de pontos de

origem e de destino estão conectados (Figura a) e uma rede do tipo eixo-raio (Figura b), onde os quadrados representam os concentradores e os círculos, os pontos de origem e de destino.

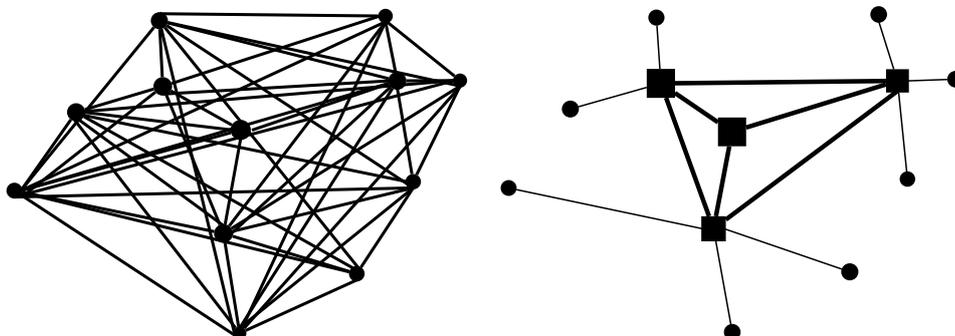


Figura 1.1: (a) Conexões entre pares origem/destino. (b) Rede eixo-raio.

O desenho das redes eixo-raio consiste basicamente em instalar os concentradores e as conexões entre eles, os eixos, e alocar os pontos de origem e de destino aos concentradores, formando os raios. Os pontos de origem e destino podem ser atribuídos aos concentradores de duas formas: alocação simples, cada ponto pode ser alocado a um único concentrador, e alocação múltipla, permite-se que os pontos de origem e de destino possam ser alocados a mais de um concentrador. Na Figura 1.2 está uma ilustração dos dois tipos de alocação: simples e múltipla.

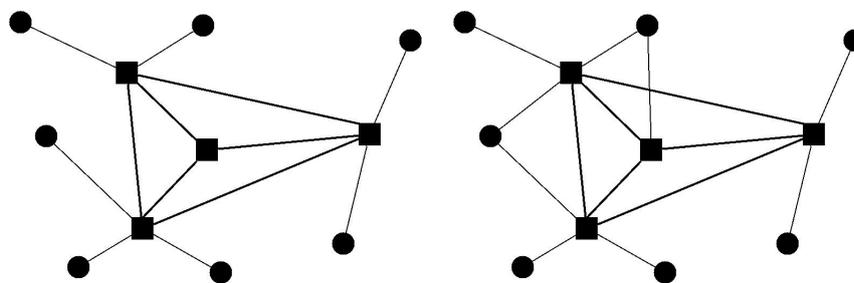


Figura 1.2: (a) Alocação Simples (b) Alocação Múltipla

Nos modelos clássicos de localização de concentradores, geralmente as seguintes suposições são consideradas:

- Os concentradores são completamente conectados, ou seja, existe uma conexão direta entre todos os pares de concentradores.
- Ao se utilizar conexões entre concentradores um desconto, representado por um fator escalar  $\alpha$  ( $0 \leq \alpha \leq 1$ ), é obtido.
- Não são permitidas conexões diretas entre pontos que não são concentradores.

Em uma rede de transporte público do tipo eixo-raio, meios de transporte de maior capacidade (trem e metrô) são usados nas conexões entre concentradores, enquanto linhas locais (de ônibus, por exemplo) são usadas nas conexões entre os pontos de origem e de destino aos concentradores.

Nesse tipo de sistema de transporte, geralmente, não é possível atender às suposições clássicas devido à inviabilidade técnica ou financeira de se conectar todos os concentradores. Logo, ao projetar uma rede de transporte público baseada em uma rede eixo-raio geralmente algumas dessas suposições são relaxadas para se obter um modelo mais realístico. Neste trabalho é utilizado um modelo para transporte público em que a rede de concentradores é uma árvore, isto é, a suposição de concentradores completamente interconectados é relaxada. Veja um exemplo de localização de concentradores em árvore na Figura 1.3. A principal motivação para o uso deste tipo de rede para modelar o transporte público é o fato de que, normalmente, o custo de instalação das conexões entre concentradores é muito alto, sendo conveniente projetar uma rede com o menor número de conexões possíveis.

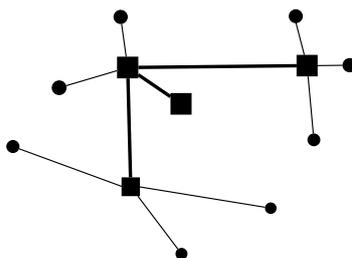


Figura 1.3: Localização de concentradores em árvore.

Uma aplicação real de localização de concentradores em árvore, citada por [Contreras et al. \(2010\)](#), é o projeto de uma rede de trens de alta velocidade na Espanha. Esta rede, que deverá estar pronta por volta de 2020, tem o formato de árvore e foi projetada para que toda cidade com mais de 10.000 habitantes esteja a um raio de 50 km de uma estação (concentradores).

Em função da complexidade do problema e dos altos valores financeiros envolvidos, o desenvolvimento de métodos exatos de resolução se mostra fundamental. Portanto, o principal foco desse trabalho é o desenvolvimento de algoritmos exatos competitivos para resolução de um modelo de localização de concentradores aplicada ao transporte público.

No capítulo 2, é feita uma revisão da literatura dos principais problemas de localização de concentradores e de modelos de localização de concentradores aplicada ao transporte público. Também é feita uma breve introdução do método de decomposição de Benders ([Benders, 1962](#)) que, devido as características da formulação usada neste

trabalho, é um método adequado para resolver o problema.

No capítulo 3, uma formulação baseada no modelo proposto por [Contreras et al. \(2009\)](#) é apresentada para modelar o Problema de Localização de Concentradores Aplicada ao Transporte Público (PLCATP). Em seguida, são apresentados o desenvolvimento do método de decomposição de Benders aplicado ao problema e o algoritmo proposto. Testes computacionais usando instâncias padrão da literatura mostram a competitividade do método de decomposição de Benders frente ao aplicativo comercial CPLEX.

O capítulo 4, apresenta quatro variantes do método de decomposição de Benders que possuem o objetivo de acelerar a convergência do método ao gerar cortes mais eficientes. Nesse capítulo é introduzido o conceito de corte pareto-ótimo apresentado por [Magnanti e Wong \(1981\)](#) e aperfeiçoado por [Papadakos \(2008\)](#), cortes usando o Subsistema de Inviabilidade Mínimo (MIS) proposto por [Fischetti et al. \(2008\)](#) e cortes usando o Sistema de Viabilidade Máximo (MFS) proposto por [Saharidis e Ierapetritou \(2010\)](#). Além disso, uma nova técnica, nomeada  $\lambda$ -ótimo, é proposta para melhorar a geração de cortes pareto-ótimo quando o subproblema de Benders é ilimitado.

No capítulo 5, os resultados dos testes computacionais são apresentados mostrando a eficiência computacional dos algoritmos propostos no capítulo 4. O capítulo é finalizado com uma análise dos resultados computacionais.

Finalmente, o capítulo 6 apresenta uma conclusão final e a possibilidade de trabalhos futuros.

# Capítulo 2

## Exame da Literatura

*“Para obter algo que você nunca teve, precisa fazer algo que nunca fez ”*

Chico Xavier

Neste capítulo é feito um exame da literatura dos problemas de localização de concentradores e das principais aplicações. Também é feita uma revisão de localização de concentradores aplicada ao transporte público e uma descrição dos principais modelos encontrados na literatura. Para finalizar é apresentada uma breve introdução do método de decomposição de Benders.

### 2.1 Localização de Concentradores

A área de localização de concentradores pode ser ramificada em várias classes de problemas. Nesta seção é feito um exame de literatura apresentando os trabalhos pioneiros nesta área.

O’Kelly (1987) apresenta a primeira formulação matemática (modelo quadrático) para modelar o problema de localização de concentradores  $p$ -Mediana com alocação simples. Dado um conjunto de pontos de demanda, o objetivo deste problema é instalar  $p$  concentradores, minimizando o custo total de transporte e garantindo que todas as demandas sejam atendidas e que cada ponto seja alocado a um único concentrador.

Uma formulação linear inteira para o problema acima com alocação múltipla é apresentada por Campbell (1992). Enquanto O’Kelly (1992) propõe um nova variante para o problema de localização de concentradores, introduzindo na função objetivo um custo fixo para instalar os concentradores. Dessa forma, o número de concentradores a serem instalados se torna também uma variável de decisão do problema.

Campbell (1994) propõe uma formulação linear inteira mista para o problema  $p$ -mediana com alocação simples, além de apresentar também formulações para os problemas de localização de concentradores com custo fixo com alocação simples/múltiplas

capacitados/não capacitados. [Campbell](#) também introduz outros dois tipos de problemas: p-centro e cobertura de concentradores. Onde o objetivo do problema p-centro é localizar  $p$  concentradores minimizando o custo máximo de transporte. Enquanto, o problema de cobertura de concentradores tem como objetivo instalar os concentradores de forma a cobrir todas as demandas, minimizando o custo de instalação.

[O’Kelly e Bryan \(1998\)](#) consideram que o uso de um fator fixo de desconto  $\alpha$  subestima a solução ótima, fornecendo descontos para pequenas quantidades de fluxos. Para corrigir este problema, eles incorporam uma função de custo dependente do fluxo na função objetivo para melhor representar a economia de escala.

[Sung e Jin \(2001\)](#) propõem uma variante de localização de concentradores que além do conjunto de  $n$  nós, possui um conjunto com  $p$  agrupamentos dos nós. O objetivo é instalar um concentrador em cada agrupamento, alocando cada nó do agrupamento a este concentrador, minimizando o custo de instalação e o custo total de transporte.

[Elhedhli e Hu \(2005\)](#) consideram o efeito do congestionamento no projeto de rede do tipo eixo-raio. Para modelar o efeito do congestionamento, eles usam uma função convexa que cresce exponencialmente com o aumento do fluxo. Para resolver o problema, é feita uma linearização do modelo usando uma função linear por partes e então aplicado um algoritmo baseado em relaxação langrangiana.

[Campbell et al. \(2005a,b\)](#) introduzem um novo problema que além de decidir quais concentradores serão instalados, também, toma decisões referentes à instalação das conexões entre concentradores. Este problema é muito comum em desenhos de redes de telecomunicação e sistemas de transporte em que é inviável instalar todas as conexões entre os concentradores devido ao alto custo. Além de [Campbell et al.](#), vários outros autores têm trabalhado nesta classe de problemas, dentre eles, [Yoon e Current \(2008\)](#); [Alumur et al. \(2009\)](#); [Calik et al. \(2009\)](#).

Devido a grande aplicação de redes eixo-raio para modelar sistemas de transporte e redes de telecomunicação, os problemas de localização de concentradores têm se tornado, nos últimos anos, um importante campo de pesquisa. Uma revisão de localização de concentradores em projetos de redes de telecomunicação e sistemas de computadores pode ser encontrada em [Klincewicz \(1998\)](#). Enquanto, [Bryan e O’Kelly \(1999\)](#) fazem uma revisão de redes eixo-raio aplicadas ao transporte aéreo de passageiros e sistemas de entrega de pacotes. [Campbell et al. \(2002\)](#) apresentam uma revisão dos principais problemas, aplicações e métodos de resolução. Já, um recente exame de literatura pode ser encontrado em [Alumur e Kara \(2008\)](#), onde mais de 100 artigos nesta área são revisados.

Na próxima seção são apresentadas algumas aplicações de redes eixo-raio em sistemas de transporte aéreo de passageiros e de cargas, redes de telecomunicação e serviços postais.

## 2.2 Aplicações

Existem várias referências na literatura de aplicação de redes eixo-raio para modelar sistemas de transporte aéreo tanto de passageiros quanto de cargas. Isso é justificado principalmente pela economia de escala obtida ao consolidar vôos de diferentes origens e mesmo destino nos aeroportos concentradores. O'Kelly (1998) discute a diferença entre os objetivos ao se projetar sistemas de transporte aéreo de passageiros e de carga, por exemplo, enquanto no transporte de passageiros há o interesse no conforto dos mesmos, no caso do transporte de carga há uma maior preocupação com o custo total da rede.

Jaillet et al. (1996) apresentam um modelo para desenhar uma rede de transporte aéreo sem assumir a princípio a estrutura eixo-raio, adotando três políticas diferentes: uma parada, duas paradas e sem restrição no número de paradas em concentradores diferentes. O resultado obtido sugere a presença de concentradores. Já Sasaki et al. (1999) apresentam um modelo de rede aérea com uma parada apenas, ou seja, entre o ponto de origem e de destino existe apenas um concentrador, baseado no problema p-mediana com alocação múltipla para o transporte aéreo de passageiros. Dois algoritmos para resolver o problema são apresentadas.

Kara e Tansel (2003) apresentam um modelo de localização de concentradores aplicado ao problema de transporte aéreo de cargas. Neste modelo, o tempo gasto nos concentradores também é levado em consideração, ou seja, além do tempo de vôo, o tempo gasto entre os vôos também é considerado. Este modelo é bastante realista levando-se em consideração que este tempo é muito significativo, principalmente para o caso de entregas noturnas.

Uma outra aplicação de redes eixo-raio é no serviço de entrega postal, em que os pontos de origem/destino podem ser, por exemplo, agências de correios e os concentradores, centros de triagem. Ernst e Krishnamoorthy (1996) apresentam uma formulação para o problema de localização de concentradores p-mediana, baseado em um sistema de entrega postal. Neste modelo, considera-se uma estrutura de custos diferente do que geralmente se encontra na literatura, isto é, o custo de transporte para coleta, transferência e distribuição podem ser diferentes. A principal motivação para esta estrutura de custos é o fato de que diferentes modais de transporte ou veículos podem ser usados para cada uma dessas atividades. Neste trabalho, é introduzido também um novo conjunto de dados, instâncias AP (Australian Post), do serviço de entrega postal australiano. Cetiner et al. (2006) fazem um estudo baseado no serviço postal turco, usando o problema integrado de localização de concentradores e roteamento. Para resolver o problema, eles propõem uma heurística que resolve o problema em dois estágios: no primeiro estágio resolve-se o problema de localização e no segundo estágio é feito um roteamento nas cidades alocadas ao concentrador.

Uma outra área em que sistemas eixo-raio são bastante utilizados é no desenho de redes de telecomunicações. Geralmente, os sistemas de telecomunicação possuem formatos variados dependendo da aplicação da rede. [Klincewicz \(1998\)](#) apresenta uma boa revisão das diferentes arquiteturas de redes eixo-raio de telecomunicação. [Dutta e Kubat \(1999\)](#) usam redes eixo-raio para projetar uma rede de telefonia celular onde a localização de concentradores já é conhecida a princípio. [Yamana e Carello \(2005\)](#) projetam uma rede utilizando o problema capacitado de localização de concentradores. Neste problema além do custo fixo para instalar os concentradores, considera-se um custo para instalar as capacidades das conexões. Uma formulação quadrática mista é usada para modelar o problema e dois algoritmos (um exato e outro heurístico) são apresentados para resolver o modelo em questão.

Devido aos vários problemas encontrados atualmente no trânsito e à possibilidade de se aumentar a eficiência do transporte público ao se integrar vários modais, a pesquisa em modelagem de transporte público usando redes eixo-raio tem atraído a atenção de pesquisadores nos últimos anos. Na próxima seção, é apresentado um exame de literatura de modelos de localização de concentradores aplicada ao transporte público.

### 2.3 Localização de Concentradores Aplicada ao Transporte Público

Nesta seção são apresentados os principais trabalhos que utilizam redes eixo-raio para modelar o transporte público. Por se tratar do tema central deste trabalho, esta seção é um pouco mais detalhada.

[Nickel et al. \(2001\)](#) são os primeiros a endereçarem o problema de transporte público usando o paradigma dos sistemas eixo-raio. Para modelar o problema, duas formulações com algumas suposições clássicas relaxadas são apresentadas: PT (Transporte Público) e GPT (Transporte Público Generalizado).

Na formulação PT, permite-se que a rede formada pelos concentradores não seja completamente conectada, ou seja, não é obrigatório que haja uma conexão entre cada par de concentradores. Para auxiliar na decisão de quais conexões entre concentradores serão instaladas, um custo fixo para instalar cada uma destas conexões é utilizado. Esta formulação utiliza as seguintes definições e variáveis:

Dado um conjunto  $N$  de pontos de demanda. Seja  $w_{ij}$ , a quantidade de fluxo a ser roteado da origem  $i$  até o destino  $j$  e  $c_{ij}$ , o custo de transportar uma unidade de fluxo entre os pares de pontos  $i$  e  $j$  ( $i, j \in N$ ). Existe um custo  $f_k$  para instalar o concentrador  $k$  e um custo  $I_{km}$  para se instalar uma conexão entre os concentradores  $k$  e  $m$ .

Ao longo desta seção, os índices  $i, j, k, m \in N$ , sendo assim, para facilitar a apresentação das formulações e sem prejuízo do entendimento, o domínio destes índices é omitido.

Antes de definir o conjunto de variáveis serão definidos dois tipo de arestas: tipo eixo e tipo raio. As arestas do tipo eixo referem-se às conexões entre dois pares de concentradores, enquanto as arestas do tipo raio referem-se às conexões entre pontos de origem/destino e concentradores. Sendo assim, pode-se definir o seguinte conjunto de variáveis:

- $x_{ijkm}$  é a fração de fluxo com origem em  $i$  e destino  $j$ , que será roteada através da aresta  $k - m$  do tipo eixo .
- $s_{ijkm}$  é a fração de fluxo com origem em  $i$  e destino  $j$ , que será roteada através da aresta  $k - m$  do tipo raio .
- $y_{km} \in \{0, 1\}$  é igual a 1 se a aresta  $k - m$  do tipo eixo estiver instalada e 0, caso contrário;
- $h_k \in \{0, 1\}$  é igual a 1 se o concentrador  $k$  estiver instalado e 0, caso contrário;

Segue abaixo a formulação proposta por [Nickel et al. \(2001\)](#) para o PT :

$$\min \sum_i \sum_{\substack{j \\ j \neq i}} \sum_k \sum_{\substack{m \\ m \neq k}} w_{ij} c_{km} (\alpha x_{ijkm} + s_{ijkm}) + \sum_k f_k H_k + \sum_k \sum_{\substack{m \\ m > k}} I_{km} y_{km} \quad (2.1)$$

$$\text{s. a: } \sum_m (x_{ijkm} + s_{ijkm} - x_{ijmk} - s_{ijmk}) = \begin{cases} +1 & \forall i, j, k : k = i, i \neq j \\ -1 & \forall i, j, k : k = j, i \neq j \\ 0 & \forall i, j, k : k \neq i, k \neq j \end{cases} \quad (2.2)$$

$$x_{ijkm} \leq y_{km} \quad \forall i, j, k, m : k < m \quad (2.3)$$

$$x_{ijmk} \leq y_{km} \quad \forall i, j, k, m : k < m \quad (2.4)$$

$$s_{ijik} \leq h_k \quad \forall i, j, k : k \neq j \quad (2.5)$$

$$s_{ijkj} \leq h_k \quad \forall i, j, k : k \neq i \quad (2.6)$$

$$s_{ijij} \leq h_i + h_j \quad \forall i, j \quad (2.7)$$

$$s_{ijkm} = 0 \quad \forall i, j, k, m : k \neq i, m \neq j \quad (2.8)$$

$$y_{km} \leq h_k \quad \forall k, m : k < m \quad (2.9)$$

$$y_{km} \leq h_m \quad \forall k, m : k < m \quad (2.10)$$

$$x_{ijkm}, s_{ijkm} \geq 0 \quad \forall i, j, k, m \quad (2.11)$$

$$h_k \in \{0, 1\} \quad \forall k \quad (2.12)$$

$$y_{km} \in \{0, 1\} \quad \forall k, m : k < m \quad (2.13)$$

No modelo acima a função objetivo (2.1) minimiza o custo total de transporte e o custo de instalação dos concentradores e das conexões entre concentradores. As restrições (2.2) asseguram a conservação do fluxo. As restrições (2.3) e (2.4) permitem que haja fluxo de  $i$  para  $j$  passando pela aresta  $k - m$  do tipo eixo se esta conexão estiver instalada. As restrições (2.5) e (2.6) garantem que as arestas do tipo raio devem começar ou terminar em um concentrador. As restrições (2.7) asseguram que conexões diretas devem passar por pelo menos um concentradores. As restrições (2.8) permitem que arestas do tipo raio existam apenas no início ou final de qualquer caminho. As restrições (2.9)-(2.10) garantem que só haverá a conexão entre concentradores  $k - m$  se os mesmo estiverem instalados. Por fim as restrições (2.11)-(2.13) definem o domínio da variáveis.

Além da formulação acima, Nickel et al. (2001) propõem uma extensão para este modelo, formulação GPT, permitindo que haja conexão direta entre pares de pontos que não são concentradores. Considerando que as conexões entre pontos que não são concentradores, também é uma aresta do tipo raio. Então, em um caminho entre um ponto de origem e de destino pode haver uma seqüência de arestas do tipo eixo e arestas do tipo raio, ao contrário do modelo PT onde as arestas do tipo raio só são permitidas no início ou no final do caminho. Sendo assim, este modelo não exige que a rede de concentradores seja conectada.

Para definir a nova formulação são utilizados os parâmetros e as variáveis do modelo PT e são introduzidos dois novos parâmetros:  $J_{km}$ , custo para instalar uma aresta do tipo raio entre os pontos  $k$  e  $m$ , e  $q$ , número máximo de trocas de tipo de arestas permitidas. Além disso, são definidas as seguinte variáveis:

- $z_{km} \in \{0, 1\}$  é igual a 1 se a aresta  $k - m$  do tipo raio estiver instalada e 0, caso contrário.
- $C_{ijm} \in \{0, 1\}$  é igual a 1 se o tipo da aresta muda ao longo do caminho de  $i$  para  $j$  no ponto  $m$  e 0, caso contrário;

Sendo assim, a formulação proposta por [Nickel et al. \(2001\)](#) para o GPT é dada por :

$$\min \sum_i \sum_{\substack{j \\ j \neq i}} \sum_k \sum_{\substack{m \\ m \neq k}} w_{ij} c_{km} (\alpha x_{ijkm} + s_{ijkm}) + \sum_k \sum_{\substack{m \\ m > k}} (I_{km} y_{km} + J_{km} z_{km}) + \sum_k f_k h_k \quad (2.14)$$

$$\text{s. a: restrições (2.2), (2.3), (2.4), (2.9), (2.10)} \quad (2.15)$$

$$C_{ijm} \geq \sum_k (x_{ijkm} - x_{ijmk}) \quad \forall i, j, m : m \neq i, m \neq j \quad (2.16)$$

$$C_{ijm} \geq \sum_k (x_{ijmk} - x_{ijkm}) \quad \forall i, j, m : m \neq i, m \neq j \quad (2.17)$$

$$\sum_m C_{ijm} \leq q \quad \forall i, j \quad (2.18)$$

$$s_{ijkm} \leq z_{km} \quad \forall i, j, k, m : k < m \quad (2.19)$$

$$s_{ijmk} \leq z_{km} \quad \forall i, j, k, m : k < m \quad (2.20)$$

$$x_{ijkm}, s_{ijkm} \geq 0 \quad \forall i, j, k, l \quad (2.21)$$

$$h_k \in \{0, 1\} \quad \forall k \quad (2.22)$$

$$y_{km}, z_{km} \in \{0, 1\} \quad \forall k, m : k < m \quad (2.23)$$

$$C_{ijm} \in \{0, 1\} \quad \forall i, j, l \quad (2.24)$$

Onde as restrições (2.16) e (2.17) detectam se ocorreu uma troca de tipo de aresta. As restrições (2.18) garantem que o número de trocas não excederá o limite. As restrições (2.19) e (2.20) asseguram que só haverá fluxo passando pela aresta  $k - m$  do tipo raio se esta aresta estiver instalada.

Em [Gelareh \(2008\)](#), várias formulações para o PLCATP com alocação múltipla são apresentadas. Ao permitir a alocação múltipla, isto é, permitir que cada ponto de origem/destino possa ser alocado a mais de um concentrador, ele torna o modelo mais realístico. Além disso, a suposição de que a rede de concentradores é completamente conectada é relaxada.

A formulação proposta por [Gelareh \(2008\)](#) utiliza as seguintes variáveis:

- $x_{ijkm} \in \{0, 1\}$  é igual a 1 se o fluxo com origem  $i$  e destino  $j$  passa pela conexão entre concentradores  $k - m$  e 0, caso contrário;
- $a_{ijk} \in \{0, 1\}$  é igual a 1 se o fluxo com origem  $i$  e destino  $j$  passa pela conexão  $i - k$ , onde  $k$  é um concentrador, e 0, caso contrário;

- $b_{ijk} \in \{0, 1\}$  é igual a 1 se o fluxo com origem  $i$  e destino  $j$  passa pela conexão  $k - j$ , onde  $k$  é um concentrador, e 0, caso contrário;
- $e_{ij} \in \{0, 1\}$  é igual a 1 se o fluxo com origem  $i$  e destino  $j$  segue o caminho direto  $i - j$  e 0, caso contrário;
- $y_{km} \in \{0, 1\}$  é igual a 1 se a conexão entre concentradores  $k - m$  estiver instalada e 0, caso contrário;
- $h_k \in \{0, 1\}$  é igual a 1 se o concentrador  $k$  estiver instalado e 0, caso contrário;

Segue abaixo a formulação proposta por [Gelareh \(2008\)](#) para o PLCATP:

$$\begin{aligned} \min \quad & \sum_k f_k z_{kk} + \sum_k \sum_{m>k}^m I_{km} y_{km} + \sum_i \sum_{\substack{j \\ j \neq i}} \sum_{\substack{k \\ k \neq i \\ k \neq j}} w_{ij} (c_{ik} a_{ijk} + c_{kj} b_{ijk}) + \\ & \sum_i \sum_{\substack{j \\ j \neq i}} w_{ij} \left( \sum_k \sum_{m \neq k} \alpha c_{km} x_{ijkm} + c_{ij} e_{ij} \right) \end{aligned} \quad (2.25)$$

$$\text{s. a: } \sum_{\substack{m \\ m \neq i}} x_{ijim} + \sum_{\substack{m \\ m \neq i \\ m \neq j}} a_{ijm} + e_{ij} = 1 \quad \forall i, j : i \neq j \quad (2.26)$$

$$\sum_{\substack{m \\ m \neq j}} x_{ijmj} + \sum_{\substack{m \\ m \neq i \\ m \neq j}} b_{ijm} + e_{ij} = 1 \quad \forall i, j : i \neq j \quad (2.27)$$

$$\sum_{\substack{m \\ m \neq k \\ m \neq i}} x_{ijkm} + b_{ijk} = \sum_{\substack{m \\ m \neq k \\ m \neq j}} x_{ijmk} + a_{ijk} \quad \forall i, j, k : j \neq i, k \neq i, k \neq j \quad (2.28)$$

$$y_{km} \leq h_m \quad \forall k, m : k < m \quad (2.29)$$

$$y_{km} \leq h_k \quad \forall k, m : k < m \quad (2.30)$$

$$x_{ijkm} + x_{ijmk} \leq y_{km} \quad \forall i, j, k, m : i \neq j, k < m \quad (2.31)$$

$$\sum_{\substack{m \\ m \neq k}} x_{kjdkm} \leq h_k \quad \forall j, k : k \neq j \quad (2.32)$$

$$\sum_{\substack{k \\ k \neq m}} x_{imkkm} \leq h_m \quad \forall i, m : m \neq i \quad (2.33)$$

$$e_{ij} \leq 2 - (h_i + h_j) \quad \forall i, j : j \neq i \quad (2.34)$$

$$a_{ijk} \leq 1 - h_i \quad \forall i, j, k : j \neq i, k \neq i, k \neq j \quad (2.35)$$

$$b_{ijm} \leq 1 - h_j \quad \forall i, j, m : j \neq i, m \neq i, m \neq j \quad (2.36)$$

$$a_{ijk} + \sum_{\substack{m \\ m \neq j \\ m \neq k}} x_{ijmk} \leq h_k \quad \forall i, j, k : j \neq i, k \neq i, k \neq j \quad (2.37)$$

$$b_{ijk} + \sum_{\substack{m \\ m \neq i \\ m \neq k}} x_{ijkl} \leq h_k \quad \forall i, j, k : j \neq i, k \neq i, k \neq j \quad (2.38)$$

$$e_{ij} + 2x_{ijij} + \sum_{\substack{m \\ m \neq i \\ m \neq j}} x_{ijim} + \sum_{\substack{m \\ m \neq i \\ m \neq j}} x_{ijmj} \leq h_i + h_j \quad \forall i, j : j \neq i \quad (2.39)$$

$$x_{ijkm}, y_{km}, h_k, a_{ijk}, b_{ijk}, e_{ij} \in \{0, 1\} \quad (2.40)$$

No modelo acima a função objetivo (2.25) minimiza o custo total de transporte e o custo de instalação dos concentradores e das conexões entre concentradores. As restrições (2.26)-(2.28) asseguram a conservação do fluxo. As restrições (2.29) e (2.30) asseguram que só haverá instalação de conexões entre concentradores se os mesmos estiverem instalados. As restrições (2.31) permitem que só haja fluxo de  $i$  para  $j$  passando pelos concentradores  $k$  e  $m$ , se a conexão  $k - m$  estiver instalada. As as restrições (2.32)((2.33)) permitem que somente fluxos com origem (destino) em um concentrador pode usar uma conexão entre concentradores para deixar a origem (chegar ao destino). As restrições (2.34) garantem que  $e_{ij}$  pode ser ativado somente se  $i$  ou  $j$  for um concentrador, não ambos. As restrições (2.35) permitem que  $a_{ijk}$  seja ativado somente se  $i$  não for um concentrador. Enquanto, as restrições (2.36) asseguram que  $b_{ijm}$  seja ativado somente se  $j$  não for um concentrador. As restrições (2.37) garantem que só haverá fluxo entrando em um concentrador se o mesmo estiver instalado. As restrições (2.38) asseguram que só haverá fluxo saindo de um concentrador se o mesmo estiver instalado. As restrições (2.39) seleciona as arestas no caminho de  $i$  para  $j$  dependendo do status dos dois, ou seja, se ambos são concentradores, apenas um ou nenhum dos dois são concentradores. Por fim, (2.40) define o domínio da variáveis.

Três algoritmos baseados no método de decomposição de Benders (Benders, 1962) e um algoritmo usando uma heurística gulosa com busca em vizinhança são usados para resolver instâncias de médio porte (até 45 nós) com um esforço computacional aceitável. Além disso, Gelareh (2008) apresenta duas extensões e dois modelos adicionais baseados neste modelo: um modelo em que a rede de concentradores é uma árvore e no outro, a rede tem formas poligonais.

Contreras et al. (2009, 2010) apresentam duas formulações, uma com três e outra com quatro índices, para o problema *Tree of Hub Location* (THL) ou Localização de Concentradores em Árvore, onde a rede formada pelos concentradores é obrigatoriamente uma árvore. O objetivo deste modelo é instalar  $p$  concentradores e as conexões

entre eles, de forma que a rede formada pelos concentradores seja uma árvore, minimizando o custo de transporte. Definindo a variável  $x_{ijkm} \geq 0$  representando a fração de fluxo com origem em  $i$  e destino  $j$  passando pelos concentradores  $k - m$ . A variável  $y_{km} \in \{0, 1\}$  é igual a 1 se a conexão entre os concentradores  $k$  e  $m$  for instalada e 0, caso contrário. Enquanto, a variável  $z_{ik} \in \{0, 1\}$  é igual a 1 se o ponto  $i$  for alocado ao concentrador  $k$  e 0, caso contrário. No caso  $i = k$ ,  $z_{kk}$  é igual a 1 se o concentrador  $k$  for instalado e 0, caso contrário. Para simplificar a notação define-se ainda  $O_i = \sum_j w_{ij}$  e  $D_i = \sum_j w_{ji}$  como a quantidade de tráfego que tem o ponto  $i$  como origem e destino, respectivamente.

Segue abaixo a formulação de quatro índices proposta por [Contreras et al. \(2009\)](#) para o problema THL:

$$\min \sum_i \sum_{\substack{k \\ k \neq i}} (c_{ik}O_i + c_{ki}D_i)z_{ik} + \sum_i \sum_{\substack{j \\ j \neq i}} \sum_k \sum_{\substack{m \\ m \neq k}} \alpha w_{ij} c_{km} x_{ijkm} \quad (2.41)$$

$$\text{s. a: } \sum_k z_{kk} = p \quad (2.42)$$

$$\sum_k z_{ik} = 1 \quad \forall i \quad (2.43)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijkm} + z_{im} = \sum_{\substack{r \\ r \neq m}} x_{ijmr} + z_{jm} \quad \forall i, j, m : i \neq j \quad (2.44)$$

$$x_{ijkm} + x_{ijmk} \leq y_{km} \quad \forall i, j, k, m : i \neq j, k < m \quad (2.45)$$

$$\sum_k \sum_{\substack{m \\ m > k}} y_{km} = p - 1 \quad (2.46)$$

$$x_{ijkm} \geq 0 \quad \forall i, j, k, m : i \neq j, k \neq m \quad (2.47)$$

$$y_{km} \in \{0, 1\} \quad \forall k, m : k < m; \quad (2.48)$$

$$z_{ik} \in \{0, 1\} \quad \forall i, k \quad (2.49)$$

Neste modelo, a função objetivo (2.41) minimiza o custo total de transporte. As restrições (2.42) garantem que apenas  $p$  concentradores serão instalados. As restrições (2.43) asseguram que todo ponto  $i$  deve ser alocado a um único concentrador  $k$ . As restrições (2.44) representam a conservação do fluxo. As restrições (2.45) permitem que haja fluxo de  $i$  para  $j$  passando pelos concentradores  $k$  e  $m$  se a conexão  $k - m$  estiver instalada. A restrição (2.46) asseguram que a rede de concentradores seja uma árvore.

A formulação acima apresenta uma relaxação linear bastante justa, ou seja, o valor da solução ótima do problema acima com as restrições de integralidade relaxadas é

um bom limitante inferior para o valor da solução ótima do problema original (com as restrições de integralidade). Para solucionar o problema um algoritmo baseado em relaxação Lagrangeana é usado para resolver instâncias de grande porte (até 100 nós) com um *gap* de otimalidade médio de 2,45%.

Neste trabalho, um modelo baseado na formulação anterior é usado para modelar o problema de localização de concentradores aplicada ao transporte público (PLCATP). As diferenças entre o modelo usado e o modelo proposto por [Contreras et al. \(2009\)](#) é o uso de custo fixo para instalação dos concentradores e a adição de algumas restrições que tornam o modelo ainda mais justo.

A principal motivação para o uso desse modelo é o fato de que geralmente em redes de transporte público, o custo de instalação das conexões entre os concentradores é muito alto, sendo conveniente projetar uma rede com o menor número de conexões possíveis entre os concentradores, ou seja, uma árvore. Além do fator financeiro, questões geográfica e políticas, entre outras, podem requerer uma rede em que exista um único caminho entre cada par de origem e de destino ([Gelareh, 2008](#)). Um exemplo concreto de localização de concentradores em árvore para modelar o transporte público é o projeto de trens de alta velocidade na Espanha. Onde a rede formada pelas estações de trens (concentradores) tem o formato de árvore.

Devido ao fato do modelo usado neste trabalho ser uma formulação linear inteira mista com uma estrutura decomponível, ou seja, fixadas as variáveis inteiras o problema pode ser decomposto em vários subproblemas menores, então o método de decomposição de Benders se mostra adequado para resolver este problema. Na próxima seção é apresentada uma pequena introdução ao método de decomposição de Benders.

## 2.4 Método de Decomposição de Benders

O método de decomposição de Benders ([Benders, 1962](#)) foi proposto para resolver problemas de programação linear inteira mista. A ideia básica do algoritmo é decompor o problema original em dois problemas mais simples: um problema inteiro misto, conhecido como problema mestre (PM), e um problema linear, conhecido como subproblema de Benders (SP).

O PM é uma versão relaxada do problema original com o conjunto de variáveis inteiras e suas respectivas restrições, e a adição de uma variável contínua  $\eta$ . Enquanto, o SP é o problema original com os valores das variáveis inteiras temporariamente fixados pela solução do PM. O algoritmo resolve cada um dos dois problemas mais simples iterativamente, um de cada vez. Em cada iteração, uma nova restrição, conhecida como corte de Benders, é adicionado ao PM. Esta nova restrição é originada pela solução do

problema dual do SP. O algoritmo para quando o limite inferior (LI), solução ótima do problema mestre, é igual ao limite superior (LS) obtido ao resolver o SP.

Considere o seguinte problema :

$$\min f^\top z + c^\top x \quad (2.50)$$

$$\text{s. a: } Ax + Bz \geq b \quad (2.51)$$

$$Dz \geq d \quad (2.52)$$

$$x \geq 0 \quad (2.53)$$

$$z \in \mathbb{Z}^+ \quad (2.54)$$

Onde  $x$  e  $c$  são vetores de tamanho  $n$ ,  $z$  e  $f$  são vetores de dimensão  $p$ , e os vetores  $b$  e  $d$  tem dimensão  $m$  e  $q$  respectivamente. Já,  $B$ ,  $D$  e  $A$  são matriz  $m \times p$ ,  $q \times p$  e  $m \times n$ , respectivamente.

Este problema pode ser expresso como:

$$\min f^\top z + v(z) \quad (2.55)$$

$$Dz \geq d \quad (2.56)$$

$$z \in \mathbb{Z}^+ \quad (2.57)$$

Onde  $v(z)$  é uma função convexa definida por:

$$v(z) = \min c^\top x \quad (2.58)$$

$$\text{s. a: } Ax \geq b - Bz \quad (2.59)$$

$$x \geq 0 \quad (2.60)$$

Seja  $Z = \{z \in \mathbb{Z}^+ : Dz \geq d\}$ . Então, para cada  $z \in Z$ , o valor de  $v(z)$  é a solução ótima de um problema linear cujo dual, conhecido como subproblema de Benders (SP), é dado por:

$$\max (b - Bz)^\top u \quad (2.61)$$

$$\text{s. a: } A^\top u \leq c \quad (2.62)$$

$$u \geq 0 \quad (2.63)$$

Usando a teoria da dualidade, o valor de  $v(z)$  pode ser encontrado usando qualquer um dos dois problemas lineares. No entanto, é mais conveniente o uso da versão dual, uma vez que sua região de viabilidade não depende da variável inteira  $z$ . Seja  $\Omega = \{u \geq 0 : A^\top u \leq c\}$ , o conjunto de soluções viáveis do SP. Assumindo que  $\Omega \neq \emptyset$ , caso contrário, o problema primal seria inviável ou ilimitado, então, pelo teorema de Minkowski, o conjunto  $\Omega$  pode ser gerado a partir de um número finito de pontos extremos e de raios extremos. Onde,  $H$  e  $G$  é conjunto de pontos extremos e raios extremos de  $\Omega$ , respectivamente.

É importante ressaltar que não é conveniente soluções ilimitadas do SP, uma vez, que soluções duais ilimitadas correspondem a soluções primais inviáveis. Então, utilizando o lema de Farkas, as seguintes restrições servem para garantir que somente valores primais viáveis de  $z$  serão usados para resolver  $v(z)$ :

$$0 \geq (b - Bz)^\top u_h \quad \forall h \in H \quad (2.64)$$

Sendo assim, o problema (2.55)-(2.57) pode ser reformulado como:

$$\min f^\top z + \max\{(b - Bz)^\top u : u \in \Omega\} \quad (2.65)$$

$$\text{s. a: } Dz \geq d \quad (2.66)$$

$$0 \geq (b - Bz)^\top u_h \quad \forall u_h \in H \quad (2.67)$$

$$z \in \mathbb{Z}^+ \quad (2.68)$$

Como a solução do problema interno de maximização é sempre um ponto extremo  $u_g \in G$ , então com o auxílio de uma variável contínua  $\eta$  o problema anterior pode ser reformulado novamente como:

$$\min f^\top z + \eta \quad (2.69)$$

$$\text{s. a: } Dz \geq d \quad (2.70)$$

$$\eta \geq (b - Bz)^\top u_g \quad \forall u_g \in G \quad (2.71)$$

$$0 \geq (b - Bz)^\top u_h \quad \forall u_h \in H \quad (2.72)$$

$$z \in \mathbb{Z}^+ \quad (2.73)$$

$$\eta \geq 0 \quad (2.74)$$

Onde, o problema acima é conhecido como Problema Mestre (PM). É importante ressaltar que este problema é equivalente ao problema original, com um número menor

de variáveis, isto é, apenas as variáveis inteiras e a variável auxiliar  $\eta$ , e com um maior número de restrições, uma vez que a cardinalidade dos conjuntos  $G$  e  $H$  pode ser muito grande dependendo da dimensão do problema. Devido a grande quantidade de restrições pode ser inconveniente resolver o PM. Usando o fato que nem todas as restrições estarão ativas na solução ótima, uma alternativa é resolver o PM relaxado, isto é, sem as restrições (2.72) e (2.73) e ir adicionando estas restrições a cada iteração. Estas restrições, conhecidas como cortes de Benders, são geradas a partir da solução do SP.

Sendo assim, sempre que o subproblema primal (2.58)-(2.60) tiver solução ótima, o SP também terá uma solução ótima, que corresponde a um ponto extremo  $u_g \in G$  e dará origem a seguinte restrição, conhecida como corte de otimalidade (corte tipo I):

$$\eta \geq (b - Bz)^\top u_g \quad (2.75)$$

Onde  $\eta$  é uma variável contínua que subestima o custo do SP.

Caso o subproblema primal seja inviável, a solução do SP será ilimitada e corresponderá a um raio extremo  $u_h \in H$  e dará origem a seguinte restrição, conhecida como corte de viabilidade (corte tipo II):

$$0 \geq (b - Bz)^\top u_h \quad (2.76)$$

O trabalho pioneiro de Geoffrion e Graves (1974) comprova a eficiência computacional da decomposição de Benders na resolução de algumas classes de problemas de grande porte. Vários autores, entre eles Camargo et al. (2008); Gelareh (2008); Camargo et al. (2009), utilizam com sucesso o método de decomposição de Benders para resolução de problemas de localização de concentradores.

Para melhorar a convergência do método, várias técnicas foram propostas na literatura. McDaniel e Devine (1977) sugerem usar a relaxação linear do PM nas primeiras iterações do método de decomposição de Benders. Dessa forma são gerados vários cortes para o problema mestre, sem a necessidade de resolver um problema de programação linear inteira mista. Este pré-processamento no método de decomposição de Benders é conhecido como *hot start*. Um outra técnica, conhecida como *Benders-and-cut*, para melhorar a convergência do método é adicionar cortes de Benders nos nós da árvore de *branch and bound* usada para solucionar o problema mestre. Rei et al. (2006) mostram como o *branch* local pode ser usado para melhorar a convergência do método. Outras técnicas propostas na literatura para acelerar a convergência do método são apresentadas no capítulo 4.

No próximo capítulo o desenvolvimento do método de decomposição de Benders aplicado ao problema é feito. Em seguida, um algoritmo baseado no método de decom-

posição de Benders clássico é aplicado para resolver o problema. No final do capítulo, os resultados computacionais são apresentados, comprovando a aplicabilidade do método.

# Capítulo 3

## Localização de Concentradores Aplicada ao Transporte Público

*“Tudo vale a pena, se a alma não é pequena.”*

Fernando Pessoa

Neste capítulo, uma formulação baseada no modelo proposto por [Contreras et al. \(2009\)](#) é apresentada para modelar o PLCATP. A principal diferença entre esta formulação e a formulação proposta por [Contreras et al. \(2009\)](#) é:

- A adição de algumas restrições que fortalecem ainda mais a formulação<sup>1</sup>.
- A adição de um custo fixo para instalar os concentradores, ou seja, o número de concentradores a serem instalados é uma variável do problema;

Na próxima seção, são apresentadas algumas definições e a formulação para o problema. São mostrados ainda resultados dos testes computacionais para instâncias padrão da literatura de médio porte (40 nós).

### 3.1 Definições e Formulação

A formulação para o PLCATP requer as seguintes definições: sejam  $N$  o conjunto de pontos de demanda e  $K$  o conjunto de pontos candidatos a serem instalados os concentradores, onde  $K \subseteq N$ . Neste trabalho, considera-se que  $K \equiv N$ , isto é, todos os pontos de demanda são candidatos a se instalar um concentrador.

Para simplificar a apresentação da formulação, os índices  $i$  e  $j$ , e os índices  $k$  e  $m$  pertencem aos conjuntos  $N$  e  $K$ , respectivamente, sendo omitidos no restante do

---

<sup>1</sup>Comprovado na seção [3.3](#).

trabalho. Sendo assim, para cada ponto  $k$  está associado um custo  $f_k$  para se instalar um concentrado em  $k$ .

Para quaisquer pares de pontos  $i, j$  ( $i \neq j$ ), tem-se  $w_{ij}$  unidades de fluxo a serem roteadas da origem  $i$  até o destino  $j$ . Seja  $c_{ij}$  o custo unitário de se transportar uma unidade de fluxo do ponto  $i$  ao ponto  $j$ . Então o custo de transporte de cada unidade de fluxo entre um par de pontos de origem-destino  $i$  e  $j$  é a soma (i) do custo unitário  $c_{ik}$  de se enviar uma unidade de fluxo de  $i$  para o primeiro concentrador  $k$  no caminho de  $i$  para  $j$ , caso  $i$  não seja um concentrador; (ii) dos custos  $\alpha c_{km}$  para cada conexão entre concentradores no caminho de  $i$  para  $j$ , usufruindo de um fator de desconto  $\alpha$ , tal que  $0 \leq \alpha \leq 1$ ; e (iii) do custo  $c_{mj}$  de se enviar uma unidade de fluxo do último concentrador  $m$  para o ponto de destino  $j$ , caso  $j$  não seja um concentrador.

A variável  $x_{ijkm} \geq 0$  representa a fração de fluxo com origem em  $i$  e destino  $j$  passando pelo conexão entre concentradores  $k - m$ . Enquanto, a variável  $y_{km} \in \{0, 1\}$  é igual a 1 se a conexão entre os concentradores  $k$  e  $m$  for instalada e 0, caso contrário. Por fim, a variável  $z_{ik} \in \{0, 1\}$  é igual a 1 se o ponto  $i$  for alocado ao concentrador  $k$  e 0, caso contrário. No caso  $i = k$ ,  $z_{kk}$  é igual a 1 se o concentrador  $k$  for instalado e 0, caso contrário.

Sendo assim, o PLCATP pode ser formulado da seguinte forma:

$$\min \sum_k f_k z_{kk} + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik} + \sum_i \sum_{\substack{j \\ j > i}} \sum_k \sum_{\substack{m \\ m \neq k}} (\alpha c_{km} w_{ij} + \alpha c_{mk} w_{ji}) x_{ijkm} \quad (3.1)$$

$$\text{s. a: } z_{ik} \leq z_{kk} \quad \forall i, k : i \neq k \quad (3.2)$$

$$\sum_k z_{ik} = 1 \quad \forall i \quad (3.3)$$

$$y_{km} \leq z_{kk} \quad \forall k, m : k < m \quad (3.4)$$

$$y_{km} \leq z_{mm} \quad \forall k, m : k < m \quad (3.5)$$

$$\sum_k \sum_{\substack{m \\ m > k}} y_{km} = \sum_m z_{mm} - 1 \quad (3.6)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijkm} + z_{im} = \sum_{\substack{r \\ r \neq m}} x_{ijmr} + z_{jm} \quad \forall i, j, m : i < j \quad (3.7)$$

$$x_{ijkm} + x_{ijmk} \leq y_{km} \quad \forall i, j, k, m : i < j, k < m \quad (3.8)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijmk} \leq z_{mm} \quad \forall i, j, m : i < j \quad (3.9)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijkm} \leq z_{mm} \quad \forall i, j, m : i < j \quad (3.10)$$

$$x_{ijkm} \geq 0 \quad \forall i, j, k, m : i < j, k \neq m \quad (3.11)$$

$$y_{km} \in \{0, 1\} \quad \forall k, m : k < m; \quad (3.12)$$

$$z_{ik} \in \{0, 1\} \quad \forall i, k \quad (3.13)$$

No modelo acima a função objetivo (3.1) minimiza o custo de instalação dos concentradores e o custo de transporte. As restrições (3.2) garantem que um ponto só poder ser alocado a concentradores instalados. As restrições (3.3) asseguram que todo ponto  $i$  deve ser alocado a um único concentrador. As restrições (3.4) e (3.5) garantem que só haverá instalação de uma conexão entre concentradores se os mesmo estiverem instalados. A restrição (3.6) tenta forçar que a rede de concentradores seja uma árvore, no entanto, como será visto mais a frente esta restrição não é suficiente para garantir a árvore. As restrições (3.7) asseguram a conservação do fluxo. Enquanto, as restrições (3.8) permitem que haja fluxo de  $i$  para  $j$  passando pelos concentradores  $k$  e  $m$  se a conexão  $k - m$  estiver instalada. Por fim, as restrições (3.9) e (3.10) garantem que só haverá fluxo de  $i$  para  $j$  passando pelos concentradores  $k$  e  $m$ , se  $k$  e  $m$  estiverem instalados. Estas restrições são redundantes, mas fortalecem a relaxação linear da formulação.

## 3.2 Método de Decomposição de Benders

Nesta seção, são apresentadas a formulação para o PM e a formulação para o SP. Em seguida, um algoritmo baseado no método clássico de decomposição Benders é apresentado.

### 3.2.1 Problema Mestre e Subproblema

Seja  $\mathcal{Z} = \{(z, y) \in \{0, 1\} : z \text{ e } y \text{ satisfazem as restrições (3.2) - (3.6)}\}$ . Fixando as variáveis inteiras  $z = z^h$  e  $y = y^h$ , tais que  $(z^h, y^h) \in \mathcal{Z}$ , obtem-se o seguinte subproblema primal de Benders.

$$\min \sum_i \sum_{\substack{j \\ j > i}} \sum_k \sum_{\substack{m \\ m \neq k}} (\alpha_{km} w_{ij} + \alpha_{mk} w_{ji}) x_{ijkm} \quad (3.14)$$

$$\text{s.a: } \sum_{\substack{k \\ k \neq m}} x_{ijkm} - \sum_{\substack{r \\ r \neq m}} x_{ijmr} = z_{jm}^h - z_{im}^h \quad \forall i, j, m : i < j \quad (3.15)$$

$$x_{ijkm} + x_{ijmk} \leq y_{km}^h \quad \forall i, j, k, m : i < j, k < m \quad (3.16)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijmk} \leq z_{mm}^h \quad \forall i, j, m : i < j \quad (3.17)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijkm} \leq z_{mm}^h \quad \forall i, j, m : i < j \quad (3.18)$$

$$x_{ijkm} \geq 0 \quad \forall i, j, k, m : i < j, k \neq m \quad (3.19)$$

Fixadas as variáveis inteiras que representam a configuração topológica da rede eixo-raio, o subproblema acima pode ser decomposto em independentes subproblemas de roteamento, um para cada par  $i - j$  ( $i < j$ ), dados por:

$$\min \sum_k \sum_{\substack{m \\ m \neq k}} (\alpha_{km} w_{ij} + \alpha_{mk} w_{ji}) x_{ijkm} \quad (3.20)$$

$$\text{s.a: } \sum_{\substack{k \\ k \neq m}} x_{ijkm} - \sum_{\substack{r \\ r \neq m}} x_{ijmr} = z_{jm}^h - z_{im}^h \quad \forall m \quad (3.21)$$

$$x_{ijkm} + x_{ijmk} \leq y_{km}^h \quad \forall k, m : k < m \quad (3.22)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijmk} \leq z_{mm}^h \quad \forall m \quad (3.23)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijkm} \leq z_{mm}^h \quad \forall m \quad (3.24)$$

$$x_{ijkm} \geq 0 \quad \forall k, m : k \neq m \quad (3.25)$$

É importante ressaltar que ao decompor o problema primal (3.14)-(3.19) em um subproblema para cada par  $i - j$  ( $i < j$ ), evita-se resolver um problema com  $\frac{1}{2}(n^4 - 2n + n^2)$  variáveis e com  $\frac{1}{2}(n^2 - n)(n^2 + 5n)$  restrições, resolvendo  $\frac{1}{2}(n^2 - n)$  subproblemas com apenas  $n^2 - n$  variáveis e  $\frac{1}{2}(n^2 + 5n)$  restrições. Sendo assim, com esta decomposição economiza-se um tempo considerável para encontrar a solução ótima do subproblema.

Associando as variáveis duais  $u_{ijm}$ ,  $e_{ijkm}$ ,  $s_{ijm}$  e  $t_{ijm}$  às restrições (3.21), (3.22), (3.23) e (3.24) respectivamente, tem-se o seguinte SP, para cada par de origem-destino

$i - j$  ( $i < j$ ).

$$\max \sum_m ((z_{jm}^h - z_{im}^h)u_{ijm} - z_{mm}^h(s_{ijm} + t_{ijm})) - \sum_k \sum_{\substack{m \\ m > k}} y_{km}^h e_{ijkm} \quad (3.26)$$

$$\text{s. a: } u_{ijm} - u_{ijk} - e_{ijkm} - s_{ijk} - t_{ijm} \leq \alpha(c_{km}w_{ij} + c_{mk}w_{ji}) \quad \forall k, m : k < m \quad (3.27)$$

$$u_{ijm} - u_{ijk} - e_{ijmk} - s_{ijk} - t_{ijm} \leq \alpha(c_{km}w_{ij} + c_{mk}w_{ji}) \quad \forall k, m : k < m \quad (3.28)$$

$$u_{ijm} \in \mathcal{R} \quad \forall m \quad (3.29)$$

$$s_{ijm} \geq 0 \quad \forall m \quad (3.30)$$

$$t_{ijm} \geq 0 \quad \forall m \quad (3.31)$$

$$e_{ijkm} \geq 0 \quad \forall k, m : k < m \quad (3.32)$$

A partir da função objetivo (3.26) pode-se montar dois tipos de restrições para o PM, conhecidas como cortes de Benders. Caso o problema dual (3.26)-(3.32) possua uma solução ótima limitada, com o auxílio da variável  $\eta_{ij}$  é gerado o seguinte corte de otimalidade (corte tipo I):

$$\eta_{ij} \geq \sum_m ((z_{jm}^h - z_{im}^h)u_{ijm}^h - z_{mm}^h(s_{ijm}^h + t_{ijm}^h)) - \sum_k \sum_{\substack{m \\ m > k}} y_{km}^h e_{ijkm}^h \quad (3.33)$$

Onde  $(u^h, e^h, s^h, t^h)$  é o ponto extremo correspondente à solução do SP em uma dada iteração  $h$  e  $\eta_{ij}$  é uma variável contínua que subestima o custo do SP.

Caso o problema dual (3.26)-(3.32) possua uma solução ilimitada é gerado o seguinte corte de viabilidade (corte tipo II):

$$0 \geq \sum_m ((z_{jm}^h - z_{im}^h)u_{ijm}^h - z_{mm}^h(s_{ijm}^h + t_{ijm}^h)) - \sum_k \sum_{\substack{m \\ m > k}} y_{km}^h e_{ijkm}^h \quad (3.34)$$

Onde  $(u^h, e^h, s^h, t^h)$  é o raio extremo correspondente à solução do SP em uma dada iteração  $h$ .

Logo tem-se o seguinte problema mestre:

$$\min \sum_k f_k z_{kk} + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik} + \sum_i \sum_{\substack{j \\ j > i}} \eta_{ij} \quad (3.35)$$

$$\text{s. a: } z_{ik} \leq z_{kk} \quad \forall i, k : i \neq k \quad (3.36)$$

$$\sum_k z_{ik} = 1 \quad \forall i \quad (3.37)$$

$$y_{km} \leq z_{kk} \quad \forall k, m : k < m \quad (3.38)$$

$$y_{km} \leq z_{mm} \quad \forall k, m : k < m \quad (3.39)$$

$$\sum_{k < m} y_{km} = \sum_m z_{mm} - 1 \quad (3.40)$$

$$\eta_{ij} \geq \sum_m ((z_{jm} - z_{im}) u_{ijm}^h - z_{mm} (s_{ijm}^h + t_{ijm}^h)) - \sum_k \sum_{\substack{m \\ m > k}} y_{km} e_{ijkm}^h \quad \forall i, j, h \in H : i < j \quad (3.41)$$

$$0 \geq \sum_m ((z_{jm} - z_{im}) u_{ijm}^h - z_{mm} (s_{ijm}^h + t_{ijm}^h)) - \sum_k \sum_{\substack{m \\ m > k}} y_{km} e_{ijkm}^h \quad \forall i, j, h \in G : i < j \quad (3.42)$$

$$\eta_{ij} \geq 0 \quad \forall i, j : i < j \quad (3.43)$$

$$y_{km} \in \{0, 1\} \quad \forall k, m : k < m; \quad (3.44)$$

$$z_{ik} \in \{0, 1\} \quad \forall i, k \quad (3.45)$$

Onde G e H é o conjunto do pontos extremos e raios extremos de (3.28)-(3.32), respectivamente.

A medida que se acrescenta cortes de viabilidade ou de otimalidade ao PM, a solução do problema fica cada vez mais lenta. No entanto, por envolverem diretamente a variável  $\eta_{ij}$ , os cortes de otimalidades são mais efetivos para movimentar os limites inferior e superior que cortes de viabilidade.

Sendo assim, uma tentativa de melhorar o desempenho do método é evitar cortes de viabilidade, garantindo que apenas cortes de otimalidade possam ser gerados. Para isto, basta acrescentar restrições ao PM que garantam que o mesmo gere somente soluções viáveis (ou seja, solução do SP limitada). Ou seja, é suficiente acrescentar restrições que forcem que a rede de concentradores seja uma árvore, ou seja, uma rede conexa e sem ciclos. Isto pode ser alcançado através do uso da tradicional formulação de árvore geradora (AG) de [Maculan et al. \(2003\)](#). Nesta formulação, é criado um

ponto fictício (origem), que é alocado a um único concentrador, com uma quantidade de fluxo inicial igual ao número de concentradores. Este fluxo é distribuído na rede, de forma que cada concentrador obtenha uma única unidade de fluxo.

Seja  $f_{lj}$  a quantidade de fluxo que passa pela conexão  $l - j$  e  $q_k$  igual 1 se o concentrador  $k$  estiver conectado à origem, e 0, caso contrário. O seguinte conjunto de restrições garantem que a rede de concentradores é uma árvore.

$$\sum_j f_{0j} = \sum_k z_{kk} \quad (3.46)$$

$$\sum_{\substack{l=0 \\ l \neq k}}^n f_{lk} - \sum_{j \neq k} f_{kj} = z_{kk} \quad \forall k \quad (3.47)$$

$$f_{km} \leq ny_{km} \quad \forall k, m : k < m \quad (3.48)$$

$$f_{km} \leq ny_{mk} \quad \forall k, m : k > m \quad (3.49)$$

$$f_{0k} \leq nq_k \quad \forall k \quad (3.50)$$

$$\sum_k q_k = 1 \quad (3.51)$$

$$q_k \leq z_{kk} \quad \forall k \quad (3.52)$$

$$f_{lj} \geq 0 \quad \forall l \in \{0, \dots, n\}, j : l \neq j \quad (3.53)$$

$$q_k \in \{0, 1\} \quad \forall k \quad (3.54)$$

Onde, a restrição (3.46) impõe que o número de fluxo que sai da origem é igual à quantidade de concentradores. As restrições (3.47) asseguram que cada concentrador obterá uma única unidade de fluxo. As restrições (3.48) e (3.49) permitem que só haja fluxo entre conexões instaladas. A restrição (3.50) garante que um único concentrador será conectado a origem. Enquanto as restrições (3.51) asseguram que o fluxo da origem só poderá ser enviado ao concentrador que está conectado a ela. E por fim, as restrições (3.52) garantem que apenas concentradores podem ser alocados à origem.

Logo o PM com restrição de AG é dado por:

$$\min \sum_k f_k z_{kk} + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik} + \sum_{i < j} \eta_{ij} \quad (3.55)$$

$$\text{s. a: restrições (3.36) - (3.45)} \quad (3.56)$$

$$\text{restrições (3.46) - (3.52)} \quad (3.57)$$

Onde, a restrição (3.56) se refere as restrições do PM original, enquanto a restrição (3.57), as restrições de AG.

### 3.2.2 Algoritmo de Decomposição de Benders Clássico

Seja  $S_{PM}^*$ , a solução ótima do problema mestre, e  $SP_{ij}^*$ , a solução do  $ij$ -ésimo subproblema. Então, o algoritmo de decomposição de Benders sem o uso das restrições de árvore geradora é dado por:

---

**Algoritmo 1:** Algoritmo de decomposição de Benders clássico sem restrições de AG

---

- 1 Faça  $LS = +\infty$ ,  $LI = -\infty$ ,  $GAP = 1$
  - 2 Se  $GAP < \epsilon$ , então pare. Fim da execução, a solução ótima foi obtida.
  - 3 Resolva o PM (3.55)-(3.45), obtendo  $S_{PM}^*$  e o valor ótimo das variáveis  $z_{ik}$  e  $y_{km}$ .
  - 4 Faça  $LI = S_{PM}^*$  e atualize  $z_{ik}$  e  $y_{km}$  no problema dual (3.26)-(3.32).
  - 5 Resolva o SP (3.26)-(3.32) para cada par  $i - j$ :  
 Se o SP for ilimitado, adicione ao PM (3.55)-(3.45) um corte de tipo II usando (3.34). Caso contrário, adicione ao PM um corte de otimalidade usando (3.33).
  - 6 Se todos os SPs forem limitados, então faça:  

$$LS = \min\{LS, \sum_{i < j} SP_{ij}^* + \sum_k f_k z_{kk} + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik}\}.$$
  - 7 Faça  $GAP = \frac{LS-LI}{LS}$  e volte à linha 2.
- 

Na linha 2 do algoritmo acima, é feito o teste para verificar se a solução ótima foi encontrada, parando a execução do mesmo quando o gap de otimalidade for consideravelmente pequeno. Na terceira linha, o PM é resolvido e na quarta linha, o LI é atualizado para o valor da solução ótima do PM e as variáveis inteiras são fixadas. Em seguida, na quinta linha, cada SP (um para cada par  $i - j$ ) é resolvido, adicionando ao PM um corte de viabilidade, se o SP for ilimitado ou um corte de otimalidade, caso o SP seja limitado. Na sexta linha, o LS é atualizado usando a solução dos SP's e os custos associados as variáveis inteiras e por fim, na sétima linha, o  $GAP$  é atualizado usando o LI e o LS.

Conforme já foi ressaltado anteriormente, a adição de restrições que garantam que o problema mestre é sempre viável pode melhorar o desempenho do método. Logo abaixo é apresentado o algoritmo de decomposição usando restrições de AG para garantir a viabilidade do PM.

---

**Algoritmo 2:** Algoritmo de decomposição de Benders clássico com restrições de AG

---

- 1 Faça  $LS = +\infty$ ,  $LI = -\infty$ ,  $GAP = 1$
  - 2 Se  $GAP < \epsilon$ , então pare. Fim da execução, a solução ótima foi obtida.
  - 3 Resolva o PM (3.55)-(3.45), obtendo  $S_{PM}^*$  e o valor ótimo das variáveis  $z_{ik}$  e  $y_{km}$ .
  - 4 Faça  $LI = S_{PM}^*$  e atualize  $z_{ik}$  e  $y_{km}$  no problema dual (3.26)-(3.32).
  - 5 Resolva o SP (3.26)-(3.32) para cada par  $i - j$  e adicione ao PM um novo corte usando (3.33).
  - 6 Faça  $LS = \min\{LS, \sum_{i < j} SP_{ij}^* + \sum_k f_k z_{kk} + \sum_i \sum_{k \neq i} (O_i + D_i) c_{ik} z_{ik}\}$ .
  - 7 faça  $GAP = \frac{LS-LI}{LS}$  e volte à linha 2.
- 

A descrição do algoritmo acima é bem semelhante a descrição do algoritmo anterior. A principal diferença entre ambos os algoritmos, está na quinta linha desse algoritmo, onde é adicionado ao PM apenas cortes de otimalidade, uma vez que a solução do SP é sempre limitada. No entanto, ao utilizar a técnica de pré-processamento proposta por [McDaniel e Devine \(1977\)](#), relaxando as restrições de integralidades das variáveis inteiras do problema mestre, ou ao adicionar cortes nos nós da árvore de *branch-and-bound* (*Benders-and-cut*), as restrições (3.46)-(3.54) não continuam garantindo que as soluções do PM são sempre viáveis. Neste caso, nas primeiras iterações de Benders com as restrições de integralidade relaxadas e nas iterações de *Benders-and-cut*, o Algoritmo 1 é utilizado.

Na próxima seção, são apresentados os resultados de testes computacionais usando instâncias padrão da literatura.

### 3.3 Resultados Computacionais

Para verificar a performance do algoritmo e fazer uma análise da estrutura topológica das soluções no contexto do transporte público, quatro conjuntos de testes são realizados.

O primeiro conjunto de teste comprova que a formulação usada neste trabalho, formulação (3.1)-(3.13), é mais forte que a versão com custo fixo da formulação proposta por [Contreras et al. \(2009\)](#), isto é, formulação formada pela função objetivo (3.1) que minimiza o custo de instalação dos concentradores e o custo de transporte, sujeita às restrições (2.43)-(2.49) oriundas da formulação p-mediana para o THL, formulação (2.41)-(2.49), de [Contreras et al.](#).

O segundo conjunto de teste comprova a contribuição das técnicas de pré-processamento e de *Benders-and-cut* para melhorar a convergência do método de decomposição de

Benders usando as restrições de AG.

O terceiro conjunto de testes analisa como muda a estrutura topológica da rede ao variar os dois principais parâmetros do modelo: fator de desconto e custo de instalação dos concentradores. Nesta etapa, também é feita uma análise da estrutura das soluções no contexto do transporte público.

Finalmente, o último conjunto de teste verifica o desempenho do método de decomposição de Benders com e sem as restrições de árvore geradora frente ao software comercial CPLEX.

Os testes foram feitos usando dois conjuntos de testes padrão da literatura: conjunto AP do serviço postal australiano e o conjunto CAB do Conselho de Aviação Civil dos Estados Unidos da América introduzidos por [Ernst e Krishnamoorthy \(1996\)](#) e [O’Kelly \(1987\)](#), respectivamente. Onde, cada instância AP e CAB são denotadas por  $AP_{n,\alpha}$  e  $CAB_{n,\alpha}$ , respectivamente, onde  $n$  é o número de pontos de demanda e  $\alpha$  pode assumir o valor 2, 4, 6 e 8 para representar um fator de desconto de 0.2, 0.4, 0.6 e 0.8. O conjunto CAB possui instâncias com 10, 15, 20 e 25 pontos de demanda. Enquanto, o conjunto de instâncias AP possuem de 10 a 200 pontos demanda variando de 10 em 10.

Devido ao fato do conjunto de instâncias CAB não possuir os custos de instalação dos concentradores e o conjunto AP possuir apenas os cinquenta primeiros custos de instalação, foram utilizados os custos fixos gerados em [Camargo et al. \(2008\)](#). Estes custos foram gerados através de uma distribuição normal com média igual a  $f_0$  e coeficiente de variação igual a 40, com o objetivo de representar a variação nos valores dos custo de forma mais real possível. O valor de  $f_0$  foi calculado como sugerido por [Ebery et al. \(2000\)](#) e representa a diferença no valor da função objetivo entre o cenário em que há um concentrador virtual localizado no centro de massa e o cenário em que todos os pontos são concentradores ([Camargo et al., 2008](#)). Além disso, devido ao fato da matriz de fluxo das instâncias AP serem fracionárias foi feita uma normalização na matriz de fluxo para que haja uma coerência entre os dois conjuntos de instâncias: AP e CAB.

Além das diferenças citadas acima entre os dois conjuntos de testes, uma outra grande diferença entre eles é o fato da matriz de fluxo das instâncias CAB ser simétrica ao contrário da matriz das instâncias AP.

Todos os testes foram feitos usando um computador com oito núcleos Intel(R) Xeon(R) CPU E5630 com 2.53GHz e 24 GB de memória usando o sistema operacional Linux. O método de decomposição de Benders foi implementado em C++ usando o CPLEX 12 para resolver os problemas de programação linear e problemas de programação linear inteira mista.

Para comprovar a qualidade da relaxação linear da formulação usada neste trabalho,

Tabela 3.1: Resultados computacionais da relaxação linear das formulações para o THL com custo fixo.

Instância	F-THL1		F-THL2	
	Tempo (s)	GAP (%)	Tempo (s)	GAP (%)
AP10.2	0.29	3.01	0.05	0
AP10.4	0.11	0	0.04	0
AP10.6	0.06	0	0.03	0
AP10.8	0.04	0	0.02	0
AP20.2	92.87	10.8	17.30	0
AP20.4	89.63	3.95	16.06	0
AP20.6	76.81	0	9.46	0
AP20.8	16.3	0	5.49	0

foi feito um conjunto de teste usando instâncias AP com 10 e 20 pontos de demanda. A Tabela 3.1 apresenta o resultado da relaxação linear das duas formulações para o THL: formulação F-THL1, versão custo fixo da formulação de Contreras et al. (2009), e formulação F-THL2, formulação usada neste trabalho. De acordo com a tabela, a formulação F-THL2 resolve todas as instâncias propostas com um GAP de otimalidade de 0%, enquanto a formulação F-THL1 possui um GAP médio de otimalidade de 2.21%. Comprovando que a formulação F-THL2 baseada no modelo proposto por Contreras et al. acrescido de algumas restrições adicionais é mais forte que a formulação de Contreras et al. sem estas restrições. Além disso o tempo computacional gasto, para resolver o problema relaxado, usando a formulação F-THL1 é muito maior que o tempo gasto ao usar a formulação F-THL2.

Os demais conjuntos de teste utilizam as implementações dos algoritmos de decomposição de Benders propostos na seção anterior usando as técnicas de pré-processamento e *Benders-and-cut*. Onde, a etapa de *Benders-and-cut* foi implementada adicionando cortes na árvore de *branch-and-bound* sempre que uma nova solução incumbente é encontrada. Para comprovar a eficiência do uso dessas duas técnicas, foi feito um conjunto de testes para comparar o desempenho do método de decomposição de Benders com e sem o uso delas. Nestes testes, a quantidade de iterações de pré-processamento foi fixada como sendo o número de pontos de demanda ( $N$ ). Os resultados dos testes computacionais estão na Tabela 3.2, onde as três variantes de Benders podem ser identificadas da seguinte forma:

- Benders com PP - Versão de Benders usando apenas pré-processamento.
- Benders com B&C- Versão de Benders usando apenas *Benders-and-cut*.
- Benders com PP e B&C - Versão de Benders usando as duas técnicas: pré-processamento e *Benders-and-cut*.

Tabela 3.2: Resultados computacionais com e sem o uso das técnicas HS e B&amp;C.

Instância	Benders com PP e B&C					Benders com PP					Benders com B&C				
	Ciclos		Cortes		Tempo (s)	Ciclos		Cortes		Tempo (s)	Ciclos		Cortes		Tempo (s)
	PP	Int	I	II		PP	Int	I	II		PP	Int	I	II	
AP10.2	10	1	321	129	0.24	10	2	366	129	0.24	0	4	180	0	1.95
AP10.4	10	1	270	180	0.23	10	2	315	180	0.26	0	3	135	0	1.73
AP10.6	10	1	272	178	0.23	10	1	272	178	0.25	0	2	90	0	0.4
AP10.8	10	1	274	176	0.18	10	1	274	176	0.16	0	2	90	0	0.63
AP20.2	20	1	2151	1649	13.10	20	2	2341	1649	36.19	0	6	1140	0	2062.86
AP20.4	20	1	1936	1864	28.74	20	2	2126	1864	53.41	0	5	950	0	2986.42
AP20.6	20	1	2096	1704	8.81	20	1	2096	1704	8.72	0	4	760	0	3600.11
AP20.8	20	1	1908	1892	9.52	20	1	1908	1892	9.39	0	4	760	0	3108.84

A Tabela 3.2 apresenta o número de ciclos (iterações) da etapa de pré-processamento e da etapa inteira, o número de cortes tipo I e de cortes tipo II, e o tempo gasto para encontrar a solução ótima para cada variante. Conforme mencionado anteriormente, pode-se observar na tabela acima que, mesmo com a adição das restrições de AG, cortes tipo II podem ser gerados na etapa de pré-processamento, ou seja, quando as restrições de integralidade do PM estiverem relaxadas. Com relação ao número de iterações inteiras, a variante Benders com PP e B&C é a versão de Benders que necessita de menos iterações para encontrar a solução ótima, ou seja, apenas uma iteração inteira é necessário para todas as instâncias de teste. O desempenho das três variantes pode ser melhor observado através das Figuras 3.1-3.2 que mostra o gráfico com o desempenho do método para as três versões. Ao analisar as figuras, percebe-se que o algoritmo de Benders usando as duas técnicas apresenta o melhor desempenho, enquanto o pior desempenho é apresentado pela variante de Benders que usa apenas *Benders-and-cut*. Comprovando a contribuição da técnica de pré-processamento para um bom desempenho do método.

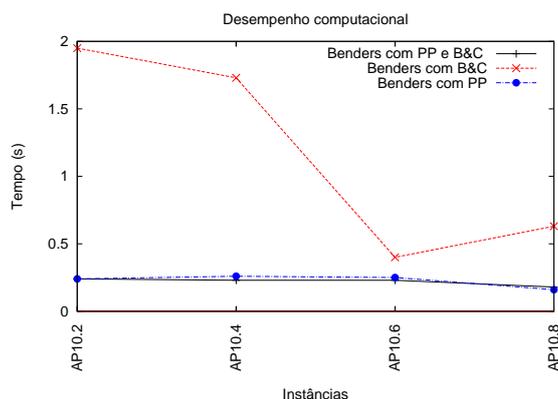


Figura 3.1: Desempenho computacional das versões do método de decomposição de Benders para instâncias AP10.

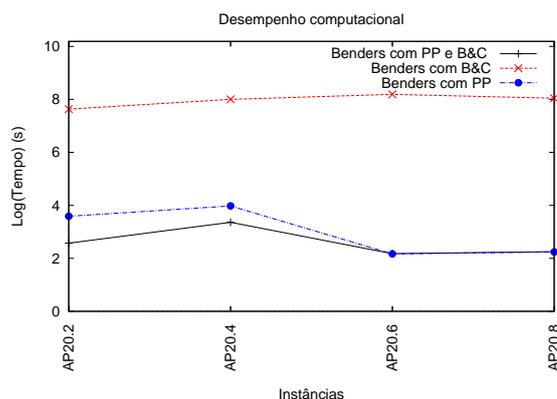


Figura 3.2: Desempenho computacional das versões do método de decomposição de Benders para instâncias AP20.

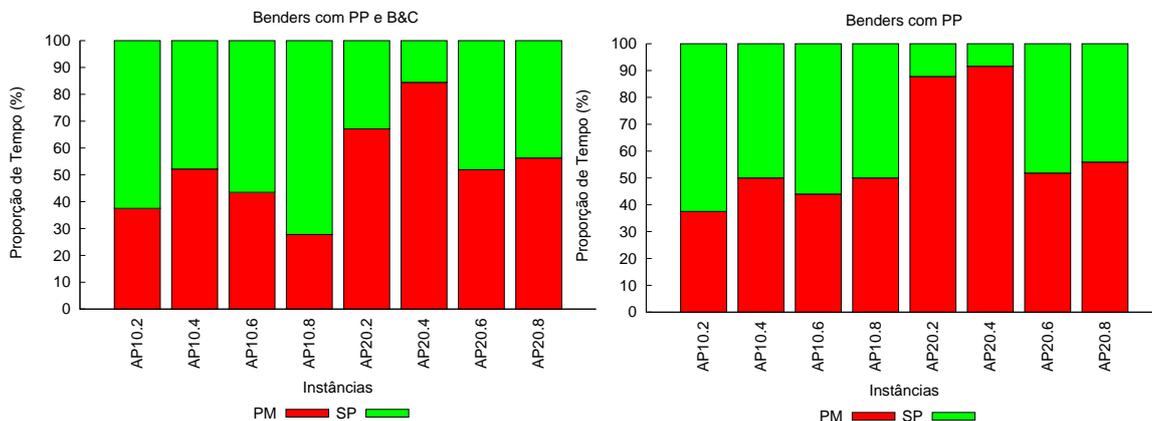


Figura 3.3: Proporção do tempo gasto pela variante Benders com PP e B&C para executar o PM e SP.

Figura 3.4: Proporção do tempo gasto pela variante Benders com PP para executar o PM e SP.

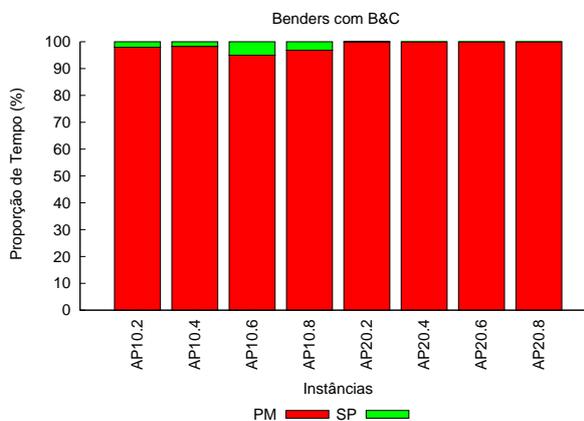


Figura 3.5: Proporção do tempo gasto pela variante Benders com B&C para executar o PM e SP.

Uma justificativa para a diferença de desempenho computacional de cada versão pode ser encontrada nas Figuras 3.3-3.5, que apresentam a proporção do tempo gasto por cada algoritmo para resolver o PM e o SP. De acordo com as figuras, o fraco desempenho da versão de Benders com B&C pode ser justificado pelo fato desta variante gastar uma fração de tempo muito grande resolvendo o PM, que é um problema linear inteiro misto. Ao contrário, das outras duas variantes em que o tempo gasto para resolver os subproblemas e o PM é mais balanceado.

A Figura 3.6 apresenta a evolução do LI e do LS para as versões: Benders com PP e B&C e Benders com PP. O gráfico ilustrado na figura representa a evolução dos limites para as duas variantes usando a instância CAB15.2 com cinco iterações de pré-processamento. De acordo com a figura, pode-se observar que o LS da versão de Benders que usa apenas pré-processamento converge rapidamente para o ótimo, enquanto, o LI gasta um tempo muito grande para convergir. Entretanto, o LI da versão de Benders com PP e B&C converge mais rápido para a solução ótima, dessa forma esta variante encontra a solução ótima mais rápido que a variante Benders com PP. A diferença entre o tempo de convergência do LS para as duas variantes de Benders está no fato de que o tempo de execução de cada iteração inteira sem *Benders-and-cut* é menor que o tempo de execução das iterações inteiras com o uso desta técnica. Sendo assim, apesar de ambos os limites superiores convergirem na terceira iteração, o tempo de convergência do LS para a variante de Benders com PP é menor. No entanto, com o uso da técnica *Benders-and-cut* (variante de Benders com PP e B&C), o número de iterações é bem menor, devido ao fato do LI convergir mais rápido com o uso desta técnica. Confirmando, dessa forma, a eficácia da adição de cortes na árvore

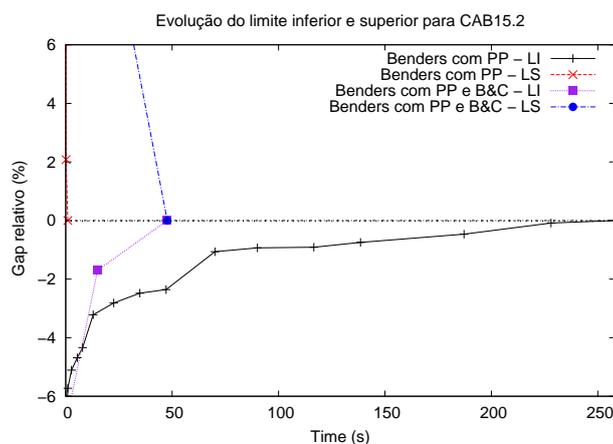


Figura 3.6: Evolução do limite inferior e do limite superior.

de *branch-and-bound* para a convergência do método.

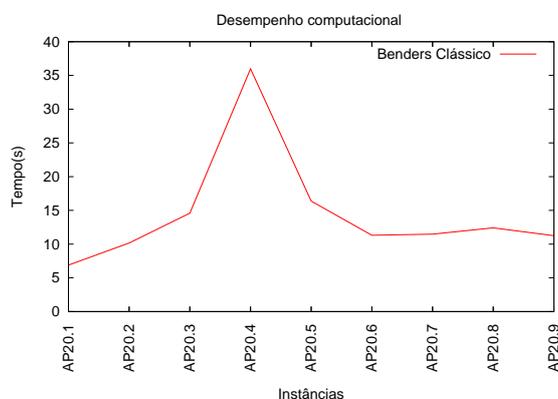


Figura 3.7: Esforço computacional versus a variação no fator de desconto.

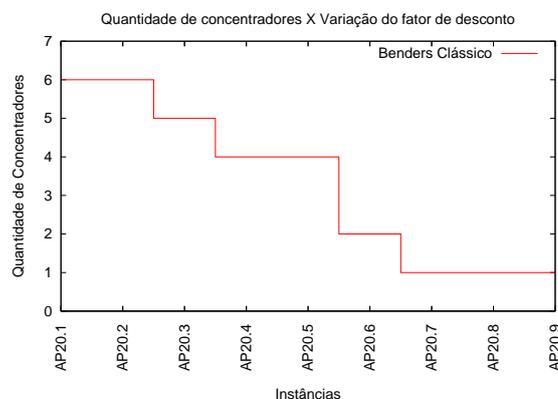
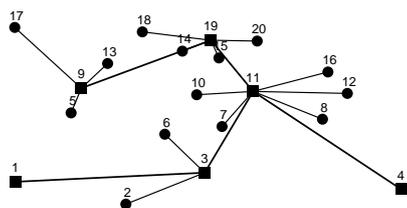
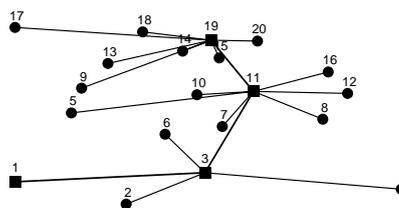
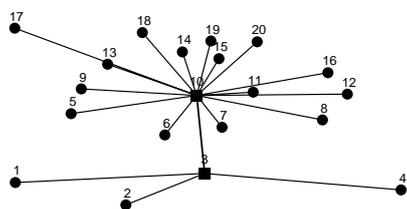
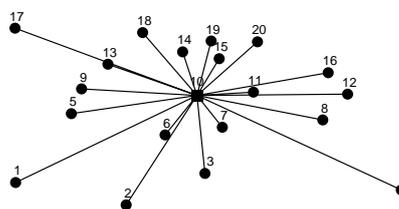


Figura 3.8: Número de concentradores instalados versus a variação no fator de desconto.

Ao observar os resultados anteriores, pode-se perceber que o tempo computacional para encontrar a solução ótima é dependente do fator de desconto. Logo o problema é muito sensível à variação dos seus parâmetros. Para verificar como o desempenho do algoritmo é afetado ao variar o fator de desconto, foram feitos testes variando este fator de 0.1 a 0.9. Para facilitar a análise, estes testes foram feitos usando instâncias AP20, multiplicando o custo de instalação por um fator igual 0.6. A Figura 3.7 mostra o desempenho computacional para cada valor de  $\alpha$ . De acordo com a figura, o algoritmo apresenta o pior desempenho para um fator de desconto  $\alpha = 0.4$ . Além disso, a variação do tempo de execução é bastante significativa, ou seja, varia de 6.89 s ( $\alpha = 0.1$ ) a 35.99 s ( $\alpha = 0.4$ ). Um outro fator a ser considerado ao variar o nível de desconto é a estrutura topológica da solução. A estrutura da rede da solução ótima, para os fatores de desconto 0.2, 0.4, 0.6 e 0.8, é apresentada nas Figuras 3.9-3.12. Para analisar as soluções encontrados no contexto do transporte público, será usado como exemplo uma rede onde os concentradores são estações de metrô. O transporte de pessoas de uma determinada origem que não possui uma estação de metrô até uma estação é feito utilizando outro meio de transporte, por exemplo, ônibus. Considera-se também que o fator preponderante para o projeto deste sistema de transporte é o fator financeiro, ou seja, aceita-se fazer viagens um pouco mais longas ou com muitas baldeações, a um custo menor.

Ao observar as figuras, percebe-se que quanto menor o fator de desconto, ou seja, maior economia de escala, é conveniente instalar mais concentradores. No entanto, conforme o desconto diminui o número de concentradores a serem instalados também

Figura 3.9: Configuração da rede para  $\alpha = 2$ .Figura 3.10: Configuração da rede para  $\alpha = 4$ .Figura 3.11: Configuração da rede para  $\alpha = 6$ .Figura 3.12: Configuração da rede para  $\alpha = 8$ .

diminui (Figura 3.8). No contexto do transporte público, ao obter uma quantidade considerável de desconto pode ser conveniente uma viagem longa, passando por várias estações que geralmente estarão dispersas na rede. No entanto, conforme o nível de desconto diminui dá-se preferência a viagens mais curtas, instalando uma quantidade menor de estações localizadas mais próximas umas das outras ou em pontos isolados.

Um outro parâmetro importante que influencia a solução computacional do problema e a estrutura da rede é o custo de instalação. Foram feitos testes variando o custo de instalação, multiplicando-o por um fator  $ff$  que varia de 20% à 300% com incrementos de 10%. Nas Figuras 3.13 e 3.14 estão representados o desempenho computacional e o número de concentradores instalados ao variar o custo de instalação para a instância AP20.2. Observando-se as figuras, pode-se perceber que quanto maior o custo de instalação dos concentradores menos concentradores são instalados e menor é o tempo computacional gasto para encontrar a solução ótima. A variação na estrutura topológica das soluções são apresentadas nas Figuras 3.15-3.18 para  $ff=50, 100, 200$  e 300%. A primeira figura (Figura 3.15) corresponde à Figura 3.9, que é primeira configuração apresentada variando o fator de desconto. Conforme o custo de instalação aumenta, este custo se torna o fator preponderante para a decisão de localização e

alocação. Isto fica mais claro ao comparar as Figuras 3.12 e 3.18. Em ambas figuras há apenas um concentrador instalado, logo, não existe economia escala. Quando o custo de instalação é pequeno (Figura 3.12), o custo de transporte é o fator que mais influencia a estrutura da rede, ou seja, é instalado o concentrador que está mais próximo do centro de massa da rede. Entretanto, quando o custo de instalação é grande (Figura 3.18), o concentrador com o menor custo de instalação é localizado.

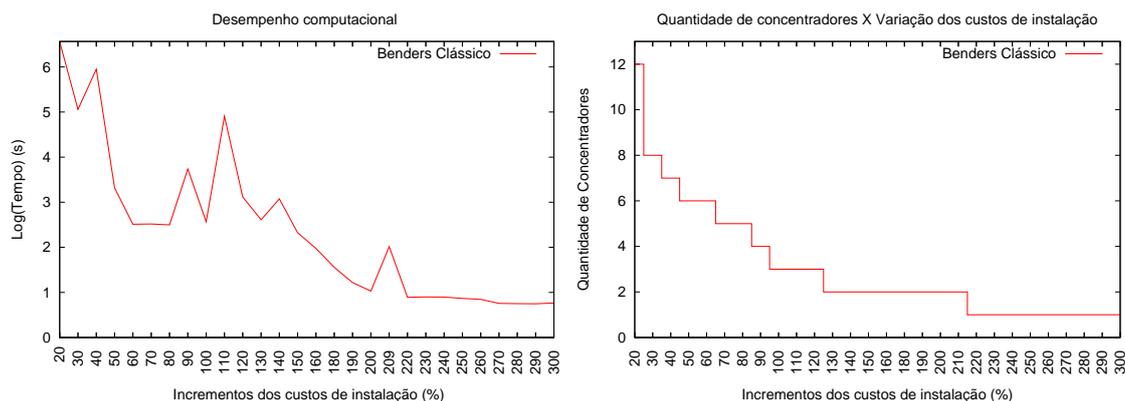


Figura 3.13: Esforço computacional versus a variação nos custos de instalação.

Figura 3.14: Número de concentradores instalados versus a variação nos custos de instalação.

Usando o custo de instalação original de cada instâncias ( $f=1$ ) e os valores 0.2, 0.4, 0.6 e 0.8 para o fator de desconto, foi feito um conjunto de testes para comparar o desempenho do método de decomposição de Benders com o aplicativo comercial CPLEX. Foram feitos testes para as duas versões do Benders: sem restrições de árvore geradora e com estas restrições. Ambas as versões foram implementadas usando pré-processamento, executado até que a solução ótima do problema original relaxado é encontrada, e usando a técnica *Benders-and-cut*. Para cada teste feito foi estabelecido como critério de parada um tempo limite de 10h, ou seja, 36000 segundos.

Os resultados dos testes computacionais para instâncias CAB e AP são apresentados nas tabelas 3.3-3.4. Onde, as colunas Ciclo de Benders PP e Int mostram a quantidade de iterações na etapa de pré-processamento e na etapa inteira do método, respectivamente. Enquanto, as colunas Tipo de Corte I e II apresentam a quantidade total de cortes de otimalidade e de cortes de viabilidade. E por fim, a coluna Tempo PP apresenta o tempo gasto no pré-processamento e a coluna Tempo Total apresenta o tempo total gasto para encontrar a solução ótima. De acordo com as tabelas, as duas variantes de Benders conseguem resolver todas as instâncias propostas dentro do tempo limite, ao contrário do CPLEX que ultrapassa o tempo limite ao resolver a instância

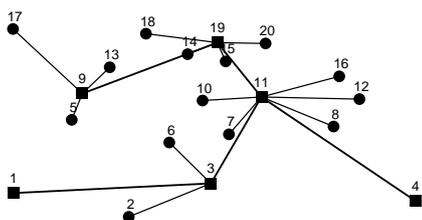


Figura 3.15: Configuração da rede para  $ff = 50\%$ .

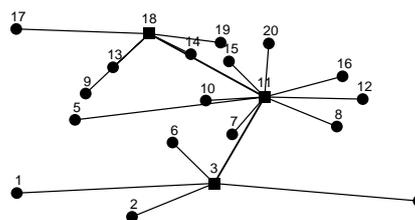


Figura 3.16: Configuração da rede para  $ff = 100\%$ .

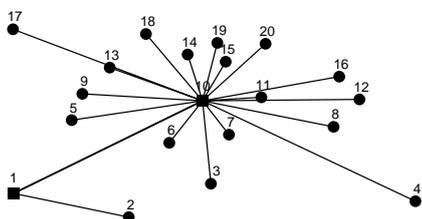


Figura 3.17: Configuração da rede para  $ff = 200\%$ .

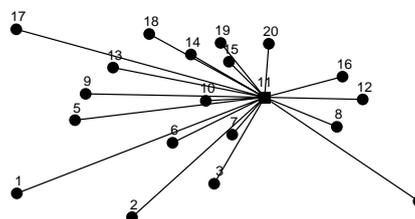


Figura 3.18: Configuração da rede para  $ff = 300\%$ .

AP40.4.

Tabela 3.3: Resultados computacionais usando instâncias CAB - CPLEX versus versões do Benders clássico.

Instância	Cplex Tempo (s)	Benders Clássico - Arvore Geradora						Benders Clássico - Sem Arvore Geradora					
		Ciclos de Benders		Tipo de corte		Tempo (s)		Ciclos de Benders		Tipo de corte		Tempo (s)	
		PP	Int	I	II	PP	Total	PP	Int	I	II	PP	Total
CAB10.2	0.25	13	1	347	283	0.29	0.31	12	1	370	215	0.29	0.33
CAB10.4	0.27	15	1	376	344	0.26	0.34	12	1	367	218	0.22	0.24
CAB10.6	0.27	15	1	414	306	0.25	0.28	11	1	332	208	0.17	0.23
CAB10.8	0.27	12	1	333	252	0.20	0.23	22	1	575	460	0.43	0.46
CAB15.2	1.07	15	1	1153	527	1.45	1.55	18	1	1377	618	1.65	1.76
CAB15.4	1.27	19	1	1435	665	2.21	2.33	28	1	2116	929	3.56	3.73
CAB15.6	1.50	27	1	2176	764	3.86	4.02	25	1	1707	1023	2.81	2.95
CAB15.8	1.35	30	1	1799	1456	5.12	5.29	23	1	1457	1063	2.84	2.93
CAB20.2	2.12	13	1	1980	680	4.12	4.51	18	1	2382	1228	5.90	6.16
CAB20.4	4.53	24	1	2873	1877	11.76	12.41	18	1	2281	1329	7.18	7.42
CAB20.6	5.03	19	1	2312	1488	8.08	8.40	41	1	5125	2855	30.00	30.87
CAB20.8	11.53	38	1	4632	2778	33.26	34.25	45	1	5378	3362	39.40	40.38
CAB25.2	7.87	21	1	4072	2528	26.09	27.24	14	1	3069	1431	14.61	15.27
CAB25.4	13.39	28	1	5150	3550	49.11	50.66	35	2	6907	4193	68.59	79.39
CAB25.6	19.29	30	1	5480	3820	70.75	72.65	18	1	3527	2173	21.39	21.96
CAB25.8	22.89	38	1	7292	4408	84.98	87.39	59	1	11649	6351	215.93	219.22

Tabela 3.4: Resultados computacionais usando instâncias AP - CPLEX versus Benders clássico.

Instância	Cplex Tempo (s)	Benders Clássico - Arvore Geradora						Benders Clássico - Sem Arvore Geradora					
		Ciclos de Benders		Tipo de corte		Tempo (s)		Ciclos de Benders		Tipo de corte		Tempo (s)	
		PP	Int	I	II	PP	Int	PP	Int	I	II	PP	Int
AP10.2	0.12	12	1	443	142	0.29	0.31	10	1	355	140	0.14	0.19
AP10.4	0.12	11	1	337	203	0.17	0.19	11	1	399	141	0.23	0.26
AP10.6	0.15	11	1	335	205	0.17	0.18	9	1	295	155	0.19	0.22
AP10.8	0.15	9	1	274	176	0.13	0.14	12	1	361	224	0.18	0.20
AP20.2	17.35	22	1	2594	1776	10.51	10.80	24	1	2820	1930	12.80	13.07
AP20.4	11.55	22	1	2358	2012	10.04	10.54	23	2	2547	2203	11.29	78.59
AP20.6	6.74	19	1	2096	1704	7.48	7.63	27	1	2686	2634	14.96	15.16
AP20.8	5.94	19	1	1908	1892	8.08	8.25	20	1	2128	1862	8.19	8.34
AP30.2	745.71	25	3	8329	3851	88.11	603.30	24	1	7096	3779	86.45	153.61
AP30.4	247.13	22	2	6340	4100	71.68	85.39	31	1	8670	5250	135.42	136.55
AP30.6	182.35	35	1	10340	5320	188.61	190.82	31	1	9074	4846	156.36	157.47
AP30.8	185.29	40	1	11794	6041	385.56	395.73	28	1	7855	4760	113.54	114.56
AP40.2	27467.69 <sup>a</sup>	34	1	17235	10065	901.97	1326.77	35	1	16974	11106	896.51	5881.34
AP40.4		36	1	18601	10259	1122.94	1549.38	35	1	16833	11247	914.21	1573.93
AP40.6	19680.65	56	1	27112	17348	5860.01	5870.30	36	1	16833	12027	1102.31	1197.44
AP40.8	5461.26	58	1	30609	15411	7240.57	7264.60	51	1	26415	14145	2889.17	4508.52

<sup>a</sup> Tempo limite excedido

Os desempenhos dos algoritmos podem ser melhor observados através das Figuras 3.19 e 3.20. É importante ressaltar, que apesar do CPLEX não ter resolvido a instância AP40.4, para efeito de comparação considera-se que o tempo de resolução desta instância pelo CPLEX é igual ao tempo limite de 36000s. De acordo com a Figura 3.19, pode-se perceber que as variantes de Benders são bastante competitivas, no entanto, o CPLEX se destaca na resolução de instâncias CAB. Já, para instâncias AP (Figura 3.20) as versões de Benders apresentam o melhor desempenho para instâncias com mais de 20 nós.

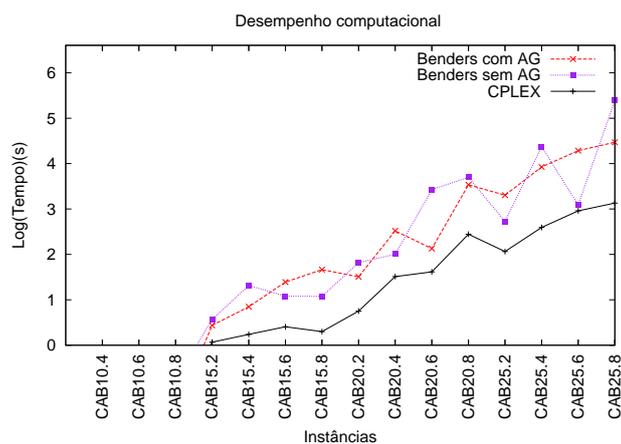


Figura 3.19: Esforço computacional usando instâncias CAB - CPLEX versus versões do Benders clássico.

Para comparar as duas variantes de Benders, ou seja, verificar a eficácia do uso

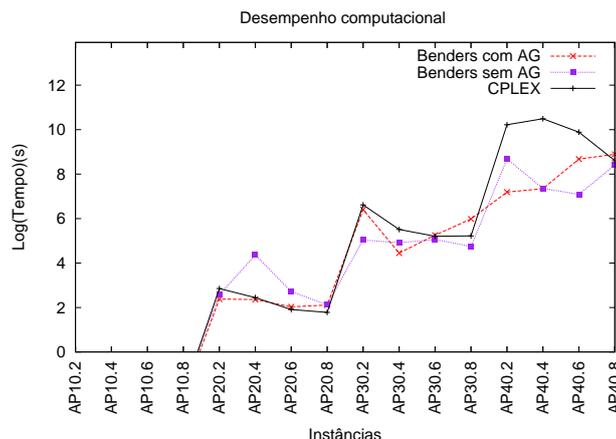


Figura 3.20: Esforço computacional usando instâncias AP - CPLEX versus versões do Benders clássico.

Tabela 3.5: Número médio de iterações de pré-processamento e tempo médio gasto nas iterações inteiras em escala relativa.

N	Número Médio - Iterações de PP		Tempo Médio - Iterações Inteiras	
	Benders com AG	Benders Sem AG	Benders com AG	Benders Sem AG
10	1.02	1	1	2.17
20	1	1.15	1	61.19
30	1.07	1	7.69	1
40	1.17	1	1	8.31

de restrições de AG para garantir a viabilidade do PM, é apresentado na Tabela 3.5 o número médio de iterações de pré-processamento e o tempo médio gasto na etapa inteira para instâncias AP com 10, 20, 30 e 40 pontos de demanda em escala relativa. Analisando esta tabela, pode-se observar que a versão de Benders com AG necessita de mais iterações de pré-processamento para encontrar a solução ótima do problema original relaxado que versão sem AG, gastando mais tempo nesta etapa. No entanto, apesar das duas versões necessitar, geralmente, de apenas uma iteração para encontrar a solução ótima na etapa inteira, a versão de Benders sem as restrições de árvore geradora gasta em média mais tempo para resolver as iterações inteiras.

Uma outra forma de comparar as duas versões de Benders é analisar os cortes gerados por cada algoritmo. Após as análises anteriores, o número de cortes não é um fator muito interessante para se analisar, uma vez que ele está correlacionado com o número de iterações, isto é, a cada iteração uma quantidade fixa de cortes é gerado. Sendo assim, um parâmetro interessante para analisar os cortes gerados pelos dois algoritmos é a proporção média dos dois tipos de cortes. A Figura 3.21 apresenta a proporção média de cortes tipo I e de cortes tipo II para a versão de Benders com AG e para a versão sem AG. De acordo com a figura, pode-se observar que ambos

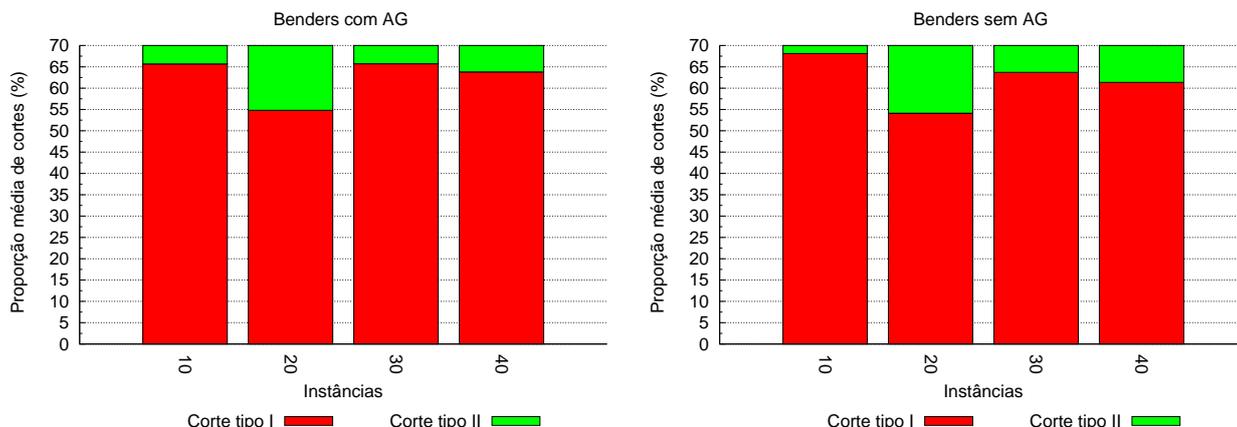


Figura 3.21: Proporção média de cortes tipo I e de cortes tipo II.

os algoritmos apresentam em média uma maior proporção de cortes tipo I que cortes tipo II. Além disso, a versão de Benders com AG apresenta uma proporção média de cortes tipo I levemente maior que a versão sem AG. No entanto, ao contrário do que se esperava, esta diferença é muito pequena. Logo, a adição de restrições de AG não reduz de forma considerável a proporção de cortes do tipo II. Uma possível explicação para isto é o fato de que a maior parte dos cortes do tipo II são gerados na etapa de pré-processamento e nesta etapa as restrições de AG não são capazes de evitar os cortes do tipo II.

Apesar das duas versões do método de decomposição de Benders terem apresentado bons resultados frente ao software comercial CPLEX, os resultados apresentados não foram satisfatórios. No próximo capítulo serão apresentadas quatro técnicas para melhorar o desempenho do método, com o objetivo de resolver instâncias maiores com tempos computacionais menores.

### 3.4 Comentários

Neste capítulo o método de decomposição de Benders foi utilizado para resolver o PLCATP. Foram feitos testes computacionais para verificar o desempenho de duas versões do método de decomposição Benders clássico, versão com e sem o uso de restrições de árvore geradora, frente ao aplicativo comercial CPLEX. Ao analisar os resultados obtidos pode-se concluir que o método apresentou um bom resultado obtendo soluções para instâncias de tamanho considerável dentro do tempo limite. Vale ressaltar que soluções para instâncias maiores só foram resolvidas em [Contreras et al. \(2009\)](#) usando

relaxação lagrangeana, com um *gap* de otimalidade relativamente alto. No próximo capítulo são apresentadas outras variante do método de decomposição de Benders para resolver instâncias de grande porte.

# Capítulo 4

## Variantes do Método de Decomposição de Benders

*“Todos querem o perfume das flores, mas poucos sujam as suas mãos para cultivá-las.”*

Augusto Cury

Apesar do método de decomposição de Benders ter sido aplicado eficientemente por alguns autores, dentre eles, [Geoffrion e Graves \(1974\)](#); [Camargo et al. \(2008\)](#); [Gelareh \(2008\)](#); [Camargo et al. \(2009\)](#), geralmente, o método possui uma convergência muito lenta na prática. Em outras palavras, são necessárias uma grande quantidade de iterações para se obter a solução ótima. Por isso, várias técnicas para melhorar a convergência do método são encontradas na literatura.

No presente capítulo são introduzidas várias metodologias para acelerar a convergência do método de decomposição de Benders e uma nova técnica é proposta na última seção. Na próxima seção, é introduzido o conceito de cortes pareto-ótimo e dois algoritmos, para o Benders com e sem restrições de AG, utilizando este tipo de corte são apresentados. Na seção [4.2](#) é apresentada a técnica proposta por [Fischetti et al. \(2008\)](#) que utiliza o Subsistema de Inviabilidade Mínimo (MIS) para gerar um único tipo de corte. Um algoritmo usando cortes do tipo MIS é apresentado. Uma outra técnica usando o Sistema de Viabilidade Máximo (MFS) proposta por [Saharidis e Ierapetritou \(2010\)](#) é apresentada na seção [4.3](#), juntamente com um algoritmo de Benders usando cortes baseado no MFS. Finalmente, na seção [4.4](#), é proposta uma nova técnica, nomeada  $\lambda$ -ótimo, para garantir a eficiência dos cortes pareto-ótimo quando o problema mestre gerar soluções inviáveis.

## 4.1 Geração de Cortes Pareto-Ótimo

Magnanti e Wong (1981) propõem uma nova metodologia para acelerar a convergência do método de decomposição de Benders. Eles observaram que quando o subproblema de Benders é degenerado, pode existir mais de uma solução ótima, logo, vários cortes diferentes podem ser gerados. A idéia básica da técnica de Magnanti e Wong (1981) é: a cada iteração do método, adicionar ao problema mestre cortes pareto-ótimo, ou seja, que não são dominados por nenhum outro corte. Mantendo a notação utilizada na seção 2.4, segue abaixo uma definição de corte pareto-ótimo:

**Definição 1.** Um corte é dito ser pareto-ótimo se ele domina todos os outros cortes. Seja  $U = \{u \geq 0 : Au \leq c\}$ , o conjunto de valores viáveis para as variáveis duais  $u$ . Então o corte de Benders associado as variáveis  $u^1 \in U$  domina o corte associado a  $u^2 \in U$ , se:

$$(b - Bz)^\top u^1 \geq (b - Bz)^\top u^2 \quad (4.1)$$

Para construir o problema de geração de cortes pareto-ótimo, é utilizado o conceito de *core point*. Segue abaixo uma definição de *core point*.

**Definição 2.** Um ponto  $z^0$  é um *core point* se ele pertence ao interior relativo da casca convexa, ou seja, se  $z^0 \in \text{ir}(Z^c)$ , onde  $\text{ir}(Z^c)$  é o interior relativo da casca convexa  $Z^c$  de  $Z$ .

Seja  $u_h$  e  $f(u_h)$ , a solução ótima e o valor da função objetivo do SP, associado à solução  $z_h$  do problema mestre. Então o subproblema de Magnanti e Wong (SMW) é dado por:

$$\max (b - Bz^0)^\top u \quad (4.2)$$

$$\text{s. a: } (b - Bz_h)^\top u = f(u_h) \quad (4.3)$$

$$A^\top u \leq c \quad (4.4)$$

$$u \geq 0 \quad (4.5)$$

O subproblema acima gera o seguinte corte pareto-ótimo a cada iteração:

$$\eta \geq (b - Bz)^\top u_h \quad (4.6)$$

O algoritmo original de Magnanti e Wong (1981) resolve dois subproblemas a cada iteração do método de decomposição de Benders: o SP, associado a solução corrente do PM e o SMW, associado a um dado *core-point*  $z^0$ , a solução do SP e a solução

corrente do PM. Logo existe uma dependência do SMW com relação à solução do SP representada pela restrição (4.3). Esta dependência pode prejudicar o desempenho do algoritmo, uma vez que, caso o SP seja um problema difícil de se resolver, soluções sub-ótimas não poderão ser utilizados, além disso, esta restrição torna o problema propenso à instabilidade numérica.

Um outro fator desfavorável na técnica de Magnanti e Wong é a dificuldade de obtenção de um novo *core point* a cada iteração, uma vez que encontrar um *core point* pode ser uma tarefa muito difícil (Papadakos, 2008; Mercier et al., 2005).

Papadakos (2008) propõe a utilização de um problema alternativo ao proposto por Magnanti e Wong (1981) para encontrar os cortes pareto-ótimo. Este problema, nomeado problema independente de Magnanti e Wong (SIMW), independe da solução do SP. Seja  $z^0 \in \text{ir}(Z^c)$  um *core point* válido, então o SIMW é dado por:

$$\max (b - Bz^0)^\top u \quad (4.7)$$

$$\text{s. a: restrição (4.4)} \quad (4.8)$$

$$u \geq 0 \quad (4.9)$$

Além de propor um problema mais simples e eficaz para a geração de cortes pareto-ótimo, ao excluir a restrição (4.3), Papadakos (2008) apresenta ainda uma nova forma de atualização dos *core points* a cada iteração. Ele mostra que a combinação convexa entre um *core point* e o ponto inteiro viável referente a solução do PM também é um *core point* válido. Seja  $z_h \in Z$ , a solução ótima do problema mestre na iteração  $h$ , então, o *core point* na iteração  $h$  pode ser atualizado da seguinte forma:

$$z_{h+1}^0 = \lambda z_h^0 + (1 - \lambda)z_h \quad (4.10)$$

Onde,  $0 \leq \lambda \leq 1$ . Portanto, dado um *core point* inicial, a atualização dos *core points* a cada iteração é trivial.

Um desvantagem na teoria de Papadakos (2008) é que ele considera que o PM fornece somente soluções viáveis. No entanto, como já foi mostrado no capítulo anterior, não existe garantia de viabilidade das soluções do PM quando as restrições de AG não são utilizadas e quando as restrições de integralidade do PM estão relaxadas (pré-processamento e *Benders-and-cut*). Uma alternativa para aplicar a teoria de Papadakos (2008) neste trabalho é fazer a atualização dos *core points* usando a equação 4.10 mesmo quando a solução do PM for inviável. Dessa forma, ao resolver o SIMW pode-se gerar

os dois tipos de cortes: corte tipo I e corte tipo II.

Vale ressaltar que adotando a estratégia acima não se pode garantir que os pontos obtidos nesta atualização sejam *core points* válidos, ou seja, garantir a qualidade dos cortes gerados. Uma alternativa mais eficiente para esta atualização, será apresenta na seção 4.4.

### 4.1.1 Subproblema Independente de Magnanti-Wong

A formulação para SIMW decomposta para cada par  $i - j$ ,  $i < j$  é dada por :

$$\max \sum_m ((z_{jm}^0 - z_{im}^0)u_{ijm} - z_{mm}^0(s_{ijm} + t_{ijm})) - \sum_k \sum_{\substack{m \\ m > k}} y_{km}^0 e_{ijkm} \quad (4.11)$$

$$\text{s. a: } u_{ijm} - u_{ijk} - e_{ijkm} - s_{ijk} - t_{ijm} \leq \alpha(c_{km}w_{ij} + c_{mk}w_{ji}) \quad \forall k, m : k < m \quad (4.12)$$

$$u_{ijm} - u_{ijk} - e_{ijmk} - s_{ijk} - t_{ijm} \leq \alpha(c_{km}w_{ij} + c_{mk}w_{ji}) \quad \forall k, m : k < m \quad (4.13)$$

$$u_{ijm} \in \mathcal{R} \quad \forall m \quad (4.14)$$

$$s_{ijm} \geq 0 \quad \forall m \quad (4.15)$$

$$t_{ijm} \geq 0 \quad \forall m \quad (4.16)$$

$$e_{ijkm} \geq 0 \quad \forall k, m : k < m \quad (4.17)$$

Onde  $(z^0, y^0) \in \mathbf{ir}(\mathcal{Z}^c)$  é um *core point*. Conforme já dito anteriormente, encontrar o *core point* inicial não é uma tarefa simples, pois, não existe nenhum método prático para obtê-lo. Neste trabalho, o seguinte ponto é utilizado como *core point* inicial:

$$z_{kk}^0 = 0.5 \quad (4.18)$$

$$z_{ik}^0 = \frac{0.5}{n-1} \quad \text{para } i \neq k \quad (4.19)$$

$$y_{km}^0 = 2 \frac{(0.5n-1)}{(n^2-n)} \quad (4.20)$$

**Proposição 1.** O ponto descrito por (4.18)-(4.20) é um *core point* válido.

*Demonstração.* O valor inicial de  $(z^0, y^0)$  pode ser construído pela combinação convexa de  $n+1$  soluções viáveis:  $n$  soluções com apenas um concentrador instalado e uma solu-

ção em que todos os pontos são concentradores. O vetor  $z^0$  pode ser obtido atribuindo valores adequados para os pesos de forma que as restrições e a combinação convexa seja respeitada. Logo um peso igual a  $\frac{1}{2(n-1)}$  é atribuído as primeiras  $n$  soluções e um peso igual a  $\frac{1}{2} - \frac{1}{2(n-1)}$  é dado para a solução com  $n$  concentradores instalados.

O vetor  $y^0$  pode ser calculado da mesma maneira, recordando que as  $n$  primeiras soluções tem vetor  $y^0$  nulo. Então o vetor  $y^0$  pode ser construído observando que é possível combinar as  $n$  árvore de caminho mais curto, cada uma composta por  $n$  concentradores, da solução com  $n$  concentradores com peso  $\frac{1}{2} - \frac{1}{2(n-1)}$ . Como cada aresta aparece duas vezes, cada componente do vetor  $y$  assume  $\frac{2}{n}$  valores. Então, para cada  $k < m$ ,  $y_{km}^0 = (\frac{1}{2} - \frac{1}{2(n-1)})\frac{2}{n}$ , Portanto  $y_{km}^0 = \frac{n-2}{n(n-1)}$ .  $\square$

Para a atualização dos *core points* a cada iteração é utilizado o valor de  $\lambda = 0.5$ , pois segundo Papadakos (2008) este valor fornece os melhores resultados empiricamente.

### 4.1.2 Algoritmo de Decomposição de Benders Usando Cortes Pareto-Ótimo

Seja  $S_{PM}^*$ , a solução ótima do problema mestre, e  $SP_{ij}^*$ , a solução do  $ij$ -ésimo sub-problema. Então, o algoritmo de decomposição de Benders com cortes pareto-ótimo é dado por:

---

**Algoritmo 3:** Algoritmo de decomposição de Benders com cortes pareto-ótimo

---

- 1 Faça  $LS = +\infty$ ,  $LI = -\infty$ ,  $GAP = 1$
  - 2 Se  $GAP < \epsilon$ , então pare. Fim da execução, a solução ótima foi obtida.
  - 3 Resolva o SIMW (4.11)-(4.13) para cada par  $i - j$ :  
Se o SIMW for ilimitado, adicione ao PM (3.55)-(3.45) um corte tipo II usando (3.34). Caso contrário, adicione ao PM um corte tipo I usando (3.33).
  - 4 Resolva o PM (3.55)-(3.45), obtendo o valor ótimo das variáveis  $z_{ik}$  e  $y_{km}$ .
  - 5 Atualize  $z_{ik}$  e  $y_{km}$  no SP (3.26)-(3.32).
  - 6 Resolva o SP (3.26)-(3.32) para cada par  $i - j$ : .
  - 7 Se o SP for ilimitado, adicione ao PM (3.55)-(3.45) um corte tipo II usando (3.34). Caso contrário, adicione ao PM um corte tipo I usando (3.33).
  - 8 Se todos os SPs forem limitados, então faça:  
$$LS = \min\{LS, \sum_{i < j} SP_{ij}^* + \sum_k f_k z_{kk} + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik}\}.$$
  - 9 Atualize os *core points* usando 4.10, faça  $GAP = \frac{LS-LI}{LS}$  e volte à linha 2.
- 

O algoritmo acima, segue o mesmo padrão do Algoritmo 1. A principal diferença entre ambos é o fato de que se resolve o SIMW antes do PM gerando um corte pareto-ótimo de qualquer um dos dois tipos (corte tipo I ou corte tipo II) e na nona linha, o *core point* é atualizado.

Ao acrescentar as restrições de AG ao PM, o SIMW e o SP serão sempre limitados. Dessa forma, serão gerados somente cortes de viabilidade (corte tipo I). Logo abaixo é formalizado o algoritmo de decomposição de Benders com cortes pareto-ótimo usando restrições de AG.

---

**Algoritmo 4:** Algoritmo de decomposição de Benders com cortes pareto-ótimo usando restrições de AG

---

- 1 Faça  $LS = +\infty$ ,  $LI = -\infty$ ,  $GAP = 1$
  - 2 Se  $GAP < \epsilon$ , então pare. Fim da execução, a solução ótima foi obtida.
  - 3 Resolva o SIMW (4.11)-(4.13) para cada par  $i - j$ : adicione ao PM um novo corte usando (3.33).
  - 4 Resolva o PM (3.55)-(3.45). Obtendo  $S_{PM}^*$  e o valor ótimo das variáveis  $z_{ik}$  e  $y_{km}$ .
  - 5 Faça  $LB = S_{PM}^*$  e atualize  $z_{ik}$  e  $y_{km}$  no problema dual (3.26)-(3.32).
  - 6 Resolva o SP (3.26)-(3.32) para cada par  $i - j$ : Adicione ao PM um novo corte usando (3.33).
  - 7 Faça  $UB = \min\{UB, \sum_{i < j} SP_{ij}^* + \sum_k f_k z_{kk} + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik}\}$ .
  - 8 Atualize os *core points* usando 4.10 e volte à linha 2.
- 

Novamente, ao relaxar as restrições de integralidade do problema mestre nas primeiras iterações ou ao utilizar o *Benders-and-cut* para resolver o PM, podem ser gerados tanto corte tipo I quanto corte tipo II. Nestes casos, mesmo com a utilização das restrições de AG, o Algoritmo 3 será utilizado.

Nas próximas seções são apresentadas outras técnicas para melhorar a convergência do método de decomposição de Benders. Apesar da maior parte das técnicas para acelerar a convergência do método serem propostas para problemas que geram apenas um tipo de corte, as técnicas que serão apresentadas em seguida foram desenvolvidas para serem aplicadas a problemas em que os dois tipos de cortes podem ser gerados, como é o caso do problema utilizado neste trabalho.

## 4.2 Subsistema de Inviabilidade Mínimo

Ao aplicar o método de decomposição de Benders em problemas que geram cortes tipo I e cortes tipo II, pode-se observar algumas diferenças entre os dois tipos de cortes, tais como: contribuição para a melhoria dos limites, instabilidade numérica e densidade de cada um dos dois tipos. Para se ter uma maior controle da qualidade dos cortes gerados, Fischetti et al. (2008) propõem a geração de um único tipo de corte. Eles partem do princípio de que a solução do SP gera um corte, de otimalidade ou viabilidade, se um dado problema alternativo for inviável. Ou seja, é possível gerar um corte a partir da solução do SP se, e somente se, o problema abaixo for ilimitado, ou seja, se o primal

deste problema for inviável:

$$\max (b - Bz_h)^\top u - \pi_0 \eta_h \quad (4.21)$$

$$\text{s. a: } A^\top u \leq \pi_0 c \quad (4.22)$$

$$u \geq 0 \quad (4.23)$$

$$\pi_0 \geq 0 \quad (4.24)$$

Onde  $(z_h, \eta_h)$  é solução do PM na iteração  $h$ , tal que,  $z_h$  é o vetor de variáveis inteiras e  $\eta_h$ , a variável contínua adicional.

Utilizando o problema acima, o seguinte corte pode ser gerado a cada iteração:

$$(b - Bz)^\top u_h - \pi_{0h} \eta \leq 0 \quad (4.25)$$

Na prática, o objetivo é gerar um novo tipo de corte baseado no Subsistema de Inviabilidade Mínimo (MIS), ou seja, uma fonte mínima de inviabilidade. O que é equivalente, a encontrar a solução (vértice) do problema:

$$\max (b - Bz_h)^\top u - \pi_0 \eta \quad (4.26)$$

$$\text{s. a: } A^\top u \leq \pi_0 c \quad (4.27)$$

$$\sum_i^m u_i + \pi_0 = 1 \quad (4.28)$$

$$u \geq 0 \quad (4.29)$$

$$\pi_0 \geq 0 \quad (4.30)$$

Onde, as variáveis  $u_i, \dots, u_m$  são os elementos do vetor coluna  $u$  de dimensão  $m$  e  $\pi_0$  é uma variável contínua. É importante ressaltar que existe uma correspondência entre os vértices do conjunto convexo formado pelas restrições 4.27-4.30 e os raios extremos (se  $\pi_0 = 0$ ) e pontos extremos (se  $\pi_0 \neq 0$ ) do conjunto de soluções viáveis do SP formado pelas restrições 2.62-2.63. Sendo assim, esta reformulação não altera o conjunto de cortes de Benders, mas ela é útil na medida que permite uma escolha mais inteligente dos cortes violados (Fischetti et al., 2008), apesar da restrição de normalização (4.28) ser bastante densa e propícia à instabilidade numérica.

### 4.2.1 Subproblema para Gerar o MIS

A formulação do subproblema MIS para cada par  $i - j$ ,  $i < j$  é dada por :

$$\max \sum_m ((z_{jm}^h - z_{im}^h)(u_{ijm}^1 - u_{ijm}^2) - z_{mm}^h(s_{ijm} + t_{ijm})) - \sum_k \sum_{\substack{m \\ m > k}} y_{km}^h e_{ijkm} - \pi_{ij}^0 \eta_{ij} \quad (4.31)$$

$$\text{s. a: } (u_{ijm}^1 - u_{ijm}^2) - (u_{ijk}^1 - u_{ijk}^2) - e_{ijkm} - s_{ijk} - t_{ijm} \leq \pi_{ij}^0 \alpha (c_{km} w_{ij} + c_{mk} w_{ji}) \quad \forall k, m : k < m \quad (4.32)$$

$$(u_{ijm}^1 - u_{ijm}^2) - (u_{ijk}^1 - u_{ijk}^2) - e_{ijmk} - s_{ijk} - t_{ijm} \leq \pi_{ij}^0 \alpha (c_{km} w_{ij} + c_{mk} w_{ji}) \quad \forall k, m : k > m \quad (4.33)$$

$$\sum_m (u_{ijm}^1 - u_{ijm}^2 + s_{ijm} + t_{ijm}) + \sum_k \sum_{\substack{m \\ m > k}} e_{ijkm} + \pi_{ij}^0 = 1 \quad (4.34)$$

$$u_{ijm}^1 \geq 0 \quad \forall m \quad (4.35)$$

$$u_{ijm}^2 \geq 0 \quad \forall m \quad (4.36)$$

$$s_{ijm} \geq 0 \quad \forall m \quad (4.37)$$

$$t_{ijm} \geq 0 \quad \forall m \quad (4.38)$$

$$e_{ijkm} \geq 0 \quad \forall k, m : k < m \quad (4.39)$$

$$\pi_{ij}^0 \geq 0 \quad (4.40)$$

O subproblema acima gera a cada iteração o seguinte tipo de corte, nomeado, corte tipo MIS:

$$\eta_{ij} \pi_{ij}^{0h} \geq \sum_m ((z_{jm}^h - z_{im}^h)(u_{ijm}^{1h} - u_{ijm}^{2h}) - z_{mm}^h(s_{ijm}^h + t_{ijm}^h)) - \sum_k \sum_{\substack{m \\ m > k}} y_{km}^h e_{ijkm}^h \quad (4.41)$$

### 4.2.2 Algoritmo de Decomposição de Benders Usando MIS

Seja  $SP_{ij}^*$ , a solução do  $ij$ -ésimo subproblema MIS. Então, o algoritmo de decomposição de Benders usando corte tipo MIS é dado por:

---

**Algoritmo 5:** Algoritmo de decomposição de Benders usando cortes baseados no MIS

---

- 1 Faça  $GAP = 1$
  - 2 Se  $GAP < \epsilon$ , então pare. Fim da execução, a solução ótima foi obtida.
  - 3 Resolva o PM (3.55)-(3.45), obtendo o valor ótimo das variáveis  $z_{ik}$ ,  $y_{km}$  e  $\eta_{ij}$ .
  - 4 Atualize  $z_{ik}$ ,  $y_{km}$  e  $\eta_{ij}$  no problema MIS (4.31)-(4.40).
  - 5 Resolva o problema MIS para cada par  $i - j$  e adicione ao PM um novo corte usando (4.41).
  - 6 Faça  $GAP = \sum_{i < j} SP_{ij}^*$  e volte ao passo 2.
- 

A segunda linha do algoritmo estabelece o critério de parada, isto é, o algoritmo para a execução quando o valor da função objetivo do subproblema MIS for igual a zero. A justificativa para este critério de parada é o fato de que o subproblema MIS é sempre limitado, sendo assim, é possível gerar cortes violados se, e somente se, este problema for estritamente positivo. Na terceira linha, o PM é resolvido e em seguida o problema MIS com as variáveis do PM fixas é resolvido para cada par  $i - j$  e um novo corte é gerado. Por fim, o gap é atualizado na última linha do algoritmo.

Na próxima seção é introduzida a técnica proposta por [Saharidis e Ierapetritou \(2010\)](#) para gerar cortes utilizando o Sistema de Viabilidade Máximo.

### 4.3 Sistema de Viabilidade Máximo

Conforme já foi ressaltado nas seções anteriores, os cortes de otimalidade são mais efetivos para melhorar os limites, inferior e superior, que cortes de viabilidade. [Saharidis e Ierapetritou \(2010\)](#) propõe a geração de um corte tipo I adicional sempre que o SP for ilimitado. No entanto, devido ao esforço computacional extra para gerar este tipo de corte, os autores propõem a utilização do Sistema de Viabilidade Máximo (MFS), ou seja, o maior conjunto de restrições do conjunto (3.22)-(3.25) que são viáveis, para gerar o novo corte tipo I. A estratégia é a seguinte: sempre que o SP for ilimitado, além de gerar um corte tipo II, é gerado um corte tipo I adicional em dois passos.

Primeiramente, utiliza-se o subproblema abaixo para gerar o MFS:

$$\min \sum_{i=0}^m w_i \quad (4.42)$$

$$\text{s. a: } Ax + MIw \geq b - Bz_h \quad (4.43)$$

$$x \geq 0 \quad (4.44)$$

$$w_i \in \{0, 1\} \quad \forall i = 1 \dots m \quad (4.45)$$

Onde  $M$  é um inteiro consideravelmente grande e  $w$  é um vetor de dimensão  $m$  composto pelos elementos  $w_i$ , para  $i = 1 \dots m$ . Após encontrar o MFS temos o seguinte problema primal viável:

$$\min c^\top x \quad (4.46)$$

$$\text{s. a: } Ax \geq b - Bz_h - MIw_h \quad (4.47)$$

$$x \geq 0 \quad (4.48)$$

Cujo dual é:

$$\max (b - Bz_h)^\top u - Mw_h^\top u \quad (4.49)$$

$$\text{s. a: } A^\top u \leq c \quad (4.50)$$

$$u \geq 0 \quad (4.51)$$

O subproblema acima é utilizado para gerar um corte extra de otimalidade. O principal ponto fraco deste método é o fato de que a cada iteração um problema inteiro (problema 4.42-4.45) deve ser resolvido para cada par  $i - j$ . Logo, conforme o tamanho do problema cresce, ou seja, o número de pontos de demanda ( $N$ ) aumenta, o esforço computacional para encontrar a solução deste problema aumentará consideravelmente.

Sendo assim, esta técnica só é conveniente quando a quantidade de cortes tipo I for bem maior que a quantidade de cortes tipo II. No caso do problema alvo neste trabalho em média mais de 60% dos cortes gerados são cortes tipo I (Figura 3.21).

### 4.3.1 Subproblema para Gerar o MFS

A seguinte formulação é utilizada para encontrar o MFS:

$$\min \sum_m (w_m^1 + w_m^2 + w_m^3 + w_m^4) + \sum_k \sum_{m>k} w_{km}^5 \quad (4.52)$$

$$\text{s.a.: } - \left( \sum_{\substack{k \\ k \neq m}} x_{ijkm} - \sum_{\substack{r \\ r \neq m}} x_{ijmr} \right) - Mw_{1m} \leq -(z_{jm}^h - z_{im}^h) \quad \forall m \quad (4.53)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijkm} - \sum_{\substack{r \\ r \neq m}} x_{ijmr} - Mw_{2m} \leq z_{jm}^h - z_{im}^h \quad \forall m \quad (4.54)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijmk} - Mw_{3m} \leq z_{mm}^h \quad \forall m \quad (4.55)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijkm} - Mw_{4m} \leq z_{mm}^h \quad \forall m \quad (4.56)$$

$$x_{ijkm} + x_{ijmk} - Mw_{5km} \leq y_{km}^h \quad \forall k, m : k < m \quad (4.57)$$

$$x_{ijkm} \geq 0 \quad \forall k, m : k \neq m \quad (4.58)$$

$$w_{1m}, w_{2m}, w_{3m}, w_{4m} \in \{0, 1\} \quad \forall m \quad (4.59)$$

$$w_{5km} \in \{0, 1\} \quad \forall k, m : k < m \quad (4.60)$$

O segundo passo consiste em gerar um corte tipo I baseado na solução do subproblema abaixo:

$$\begin{aligned} \max \sum_m ((z_{jm}^h - z_{im}^h)(u_{ijm} - v_{ijm}) - (z_{mm}^h)(s_{ijm} + t_{ijm})) - \sum_k \sum_{m>k} (y_{km}^h) e_{ijkm} \\ - M \left( \sum_m (w_{1m}^h u_{ijm} + w_{2m}^h v_{ijm} + w_{3m}^h s_{ijm} + w_{4m}^h t_{ijm}) + \sum_k \sum_{m>k} w_{5km}^h e_{ijkm} \right) \end{aligned} \quad (4.61)$$

$$\text{s. a: } (u_{ijm} - v_{ijm}) - (u_{ijk} - v_{ijk}) - e_{ijkm} - s_{ijk} - t_{ijm} \leq \alpha(c_{km}w_{ij} + c_{mk}w_{ji}) \quad \forall k, m : k < m \quad (4.62)$$

$$(u_{ijm} - v_{ijm}) - (u_{ijk} - v_{ijk}) - e_{ijmk} - s_{ijk} - t_{ijm} \leq \alpha(c_{km}w_{ij} + c_{mk}w_{ji}) \quad \forall k, m : k > m \quad (4.63)$$

$$u_{ijm}, v_{ijm}, s_{ijm}, t_{ijm} \geq 0 \quad \forall m \quad (4.64)$$

$$e_{ijkm} \geq 0 \quad \forall k < m \quad (4.65)$$

O subproblema acima gera um corte tipo I usando (3.33).

### 4.3.2 Algoritmo de Decomposição de Benders Usando o MFS

Seja  $S_{PM}^*$ , a solução ótima do problema mestre, e  $SP_{ij}^*$ , a solução do  $ij$ -ésimo subproblema. Então, o algoritmo de decomposição de Benders usando o MFS é dado por:

---

**Algoritmo 6:** Algoritmo de decomposição de Benders usando cortes baseados no MFS

---

- 1 Faça  $LS = +\infty$ ,  $LI = -\infty$ ,  $GAP = 1$
  - 2 Se  $GAP < \epsilon$ , então pare. Fim da execução, a solução ótima foi obtida.
  - 3 Resolva o SIMW (4.11)-(4.13) para cada par  $i - j$ :  
 Se o SIMW for ilimitado, adicione ao PM (3.55)-(3.45) um corte tipo II usando (3.34). Em seguida, encontre o MFS usando (4.52)-(4.60) e adicione um corte tipo I adicional usando o subproblema (4.61)-(4.65).  
 Caso contrário, adicione ao PM um corte tipo I usando (3.33).
  - 4 Resolva o PM, obtendo o valor ótimo das variáveis  $z_{ik}$  e  $y_{km}$ .
  - 5 Atualize  $z_{ik}$  e  $y_{km}$  no problema dual (3.26)-(3.32).
  - 6 Resolva o SP (3.26)-(3.32) para cada par  $i - j$ :  
 Se o problema dual for ilimitado, adicione ao PM (3.55)-(3.45) um corte tipo II usando (3.34). Em seguida, encontre o MFS usando (4.52)-(4.60) e adicione um corte tipo I adicional usando o subproblema (4.61)-(4.65).  
 Caso contrário, adicione ao PM um corte tipo I usando (3.33)
  - 7 Se todos os SPs forem limitados, então faça:  
 $LS = \min\{LS, \sum_{i < j} SP_{ij}^* + \sum_k f_k z_{kk} + \sum_i \sum_{k \neq i} (O_i + D_i) c_{ik} z_{ik}\}$ .
  - 8 Faça  $GAP = \frac{LS - LI}{LS}$  e volte à linha 2.
- 

Na próxima seção, uma nova técnica é proposta para gerar cortes pareto-ótimo mais eficiente quando o solução do subproblema for inviável.

## 4.4 Seleção de Cortes $\lambda$ -Ótimo

Conforme já foi mencionado na seção 4.1, Papadakos (2008) prova que a combinação convexa entre um *core point* e a solução ótima viável do problema mestre é um *core point* válido. Ou seja, em uma dada iteração  $h$ , o *core point*  $(z_{h+1}^0, y_{h+1}^0)$  pode ser atualizado da seguinte forma:

$$z_{h+1}^0 = (1 - \lambda)z_h^0 + \lambda z_h \quad (4.66)$$

$$y_{h+1}^0 = (1 - \lambda)y_h^0 + \lambda y_h \quad (4.67)$$

Onde  $0 \leq \lambda \leq 1$ .

Esta técnica de atualização é válida somente se a solução  $(z_h, y_h) \in \mathcal{Z}$  for uma solução viável. No entanto, sem o uso das restrições de AG, na etapa de pré-processamento ou ao utilizar a técnica *Benders-and-cut*, o PM pode gerar tanto soluções viáveis como soluções inviáveis. Sendo assim, ao atualizar os *core points* usando a estratégia proposta na seção 4.1, ou seja, usando um valor fixo para  $\lambda$ , pode-se encontrar pontos que estão fora do interior relativo da casca convexa de  $\mathcal{Z}$ . Em outras palavras, pode-se gerar pontos inviáveis e dessa forma não é mais possível garantir a efetividade dos cortes gerados.

Uma forma de manter a eficácia dos cortes pareto-ótimo é encontrar a cada iteração um valor para  $\lambda$  tal que a atualização usando as equações 4.66 e 4.67 gere *core points* válidos, ou seja, que sejam soluções viáveis. Logo, o objetivo é encontrar valores para  $\lambda$  tal que o sistema abaixo seja viável.

$$Ax \geq b - B((1 - \lambda)z_h^0 + \lambda z_h) \quad (4.68)$$

$$x \geq 0 \quad (4.69)$$

$$0 \leq \lambda \leq 1 \quad (4.70)$$

Conforme, já foi comentado na seção 4.1 o valor  $\lambda = \frac{1}{2}$  fornece os melhores resultados empiricamente. Logo, caso a solução do PM seja inteira, este valor é o utilizado para atualização do *core point*, caso contrário seria conveniente utilizar o maior valor de  $\lambda$  no intervalo  $[0, \frac{1}{2}]$  que gere um ponto viável para atualização. O valor para  $\lambda$  encontrado dessa forma é denominado  $\lambda$ -ótimo. Logo, o subproblema para encontrar o  $\lambda$ -ótimo é dado por:

$$\max \lambda \quad (4.71)$$

$$\text{s. a: restrições (4.68)-(4.69)} \quad (4.72)$$

$$0 \leq \lambda \leq \frac{1}{2} \quad (4.73)$$

Encontrar o  $\lambda$ -ótimo é uma estratégia inovadora que ainda não foi proposta na literatura. Esta nova metodologia tem como objetivo contribuir para a melhoria no desempenho do método de decomposição de Benders para problemas que possam gerar os dois tipos de corte. Uma vez que, existem várias técnicas na literatura para acelerar a convergência do método, no entanto, poucas técnicas foram propostas para este caso.

#### 4.4.1 Subproblema para Cálculo do $\lambda$ -ótimo

A formulação do subproblema para encontrar o  $\lambda$ -ótimo para cada par  $i - j$  é dada por:

$$\max \lambda_{ij} \quad (4.74)$$

$$\text{s.a: } \sum_{\substack{k \\ k \neq m}} x_{ijkm} - \sum_{\substack{r \\ r \neq m}} x_{ijmr} = (1 - \lambda_{ij})(z_{jm}^0 - z_{im}^0) + \lambda_{ij}(z_{jm}^h - z_{im}^h) \quad \forall m \quad (4.75)$$

$$x_{ijkm} + x_{ijmk} \leq (1 - \lambda)y_{km}^0 + \lambda_{ij}y_{km}^h \quad \forall k, m : k < m \quad (4.76)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijkm} \leq (1 - \lambda_{ij})z_{mm}^0 + \lambda_{ij}z_{mm}^h \quad \forall m \quad (4.77)$$

$$\sum_{\substack{k \\ k \neq m}} x_{ijkm} \leq (1 - \lambda_{ij})z_{mm}^0 + \lambda_{ij}z_{mm}^h \quad \forall m \quad (4.78)$$

$$x_{ijkm} \geq 0 \quad \forall k, m : k \neq m \quad (4.79)$$

$$\lambda_{ij} \in (0, 0.5) \quad (4.80)$$

#### 4.4.2 Algoritmo de Decomposição de Benders Usando $\lambda$ -Ótimo

Seja  $SP_M^*$ , a solução ótima do problema mestre, e  $SP_{ij}^*$ , a solução do  $ij$ -ésimo subproblema. Então, o algoritmo de decomposição de Benders usando o  $\lambda$ -Ótimo é dado por:

**Algoritmo 7:** Algoritmo de decomposição de Benders usando  $\lambda$ -Ótimo

- 1 Faça  $LS = +\infty$ ,  $LI = -\infty$ ,  $GAP = 1$
- 2 Se  $GAP < \epsilon$ , então pare. Fim da execução, a solução ótima foi obtida.
- 3 Resolva o SIMW (4.11)-(4.13) para cada par  $i - j$ :  
Se o SIMW for ilimitado, adicione ao PM (3.55)-(3.45) um corte tipo II usando (3.34). Caso contrário, adicione ao PM um corte tipo I usando (3.33).
- 4 Resolva o PM (3.55)-(3.45), obtendo o valor ótimo das variáveis  $z_{ik}$  e  $y_{km}$ .
- 5 Atualize  $z_{ik}$  e  $y_{km}$  no SP (3.26)-(3.32).
- 6 Resolva o SP (3.26)-(3.32) para cada par  $i - j$ : . Se o SP for ilimitado, adicione ao PM (3.55)-(3.45) um corte tipo II usando (3.34). Caso contrário, adicione ao PM um corte tipo I usando (3.33).
- 7 Se todos os SPs forem limitados, então faça:  
 $LS = \min\{LS, \sum_{i < j} SP_{ij}^* + \sum_k f_k z_{kk} + \sum_i \sum_{k \neq i} (O_i + D_i) c_{ik} z_{ik}\}$ . Caso contrário, resolva o problema  $\lambda$ -ótimo (4.74)-(4.80) para cada par  $i - j$ .
- 8 Atualize os *core points* usando (4.66) e (4.67) com  $\lambda = \min\{\lambda_{ij}\}$  e volte ao passo 2 .

Onde,  $\lambda_{ij}$  é o valor ótimo do  $ij$ -ésimo subproblema para cálculo do  $\lambda$ -ótimo. A diferença entre o algoritmo acima e o Algoritmo 2 é que sempre que a solução do PM for ilimitada é adicionado um corte tipo II e calculado o valor do  $\lambda$ -ótimo para a atualização dos *core points*. Como o subproblema para cálculo do  $\lambda$ -ótimo está decomposto em um subproblema para cada par  $i - j$ , calcula-se um  $\lambda_{ij}$  para cada subproblema e o valor do  $\lambda$ -ótimo será o menor valor entre os  $\lambda_{ij}$ .

No próximo capítulo, os resultados dos testes computacionais são apresentados.

# Capítulo 5

## Resultados Computacionais

*“Não há problema que não possa ser solucionado pela paciência.”*

Chico Xavier

Os testes computacionais foram feitos usando os mesmos conjuntos de testes da seção 3.3: instâncias AP e CAB. Todos os testes foram executados em um computador com oito núcleos Intel(R) Xeon(R) CPU E5630 com 2.53GHz e 24 GB de memória usando o sistema operacional Linux. O método de decomposição de Benders foi implementado em C++ usando o CPLEX 12 para resolver os problemas lineares e linear inteiro misto.

Para comparar as variantes de Benders foram feitos testes computacionais usando instâncias com até 50 pontos de demanda. Sendo estabelecido como critério de parada um tempo limite de 36000s, ou seja, 10 horas de processamento. As variantes de Benders serão referidas neste capítulo por:

- Benders CII - variante usando um  $\lambda$  fixo ( $\lambda = 0.5$ ) para atualização dos *core points*;
- Benders MIS - variante usando cortes baseado no Subsistema de Inviabilidade Mínimo;
- Benders MFS - variante usando um corte adicional de otimalidade baseado no Sistema de Viabilidade Máximo;
- Benders LO - variante usando o  $\lambda$ -ótimo para a atualização dos *core points*.

Foram feitos testes para cada variante com e sem o acréscimo das restrições de AG. As variantes foram implementadas da seguinte forma: nas iterações inteiras com restrição de AG, onde somente cortes tipo I são gerados, é utilizado o Algoritmo 3; nas

demais etapas (iterações inteiras sem restrição de AG, pré-processamento e *Benders-and-cut*), os Algoritmos 4, 5, 6 e 7 são usados para as variantes Benders CII, Benders MIS, Benders MFS e Benders LO, respectivamente.

O número de iterações de pré-processamento ( $pp$ ) é fixo e foi escolhido empiricamente para cada versão; ao contrário dos testes para a versão de Benders clássica, onde, a etapa de pré-processamento é executada até que a solução do problema original relaxado é encontrada. A motivação da escolha de um número pequeno de iterações de pré-processamento para as variantes de Benders propostas no capítulo anterior pode ser encontrado através da Figura 5.1. Esta figura apresenta a variação do tempo gasto para encontrar a solução ótima por cada variante de Benders ao variar o valor de  $pp$ . De acordo com a figura, o melhor desempenho da versão de Benders clássica é observado após uma grande quantidade de iterações de pré-processamento. Enquanto as outras variantes de Benders necessitam de poucas iterações de pré-processamento para obter um bom desempenho, não sendo necessário que esta etapa seja executada até a otimalidade.

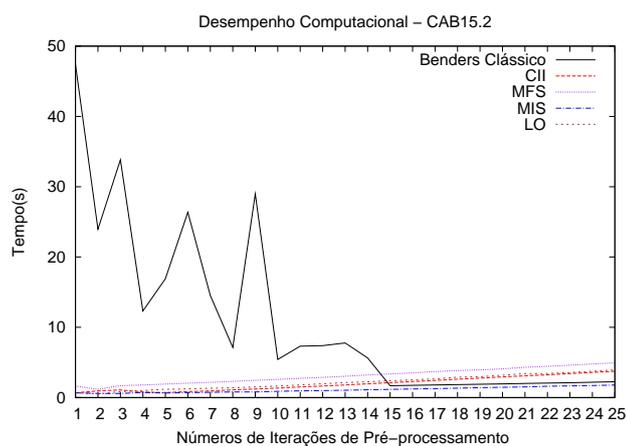


Figura 5.1: Esforço computacional ao variar o número de iterações de pré-processamento para instância CAB15.2.

Os resultados dos testes computacionais, para as versões das variantes de Benders com e sem a adição de restrição de AG, encontram-se nas Tabelas 5.1-5.4.

Tabela 5.1: Resultados computacionais dos algoritmos usando instâncias CAB sem restrições de AG.

Instância	$f_k$	LO					CII					MIS					MFS				
		N° de ciclos		N° de corte		Tempo (s)	N° de ciclos		N° de corte		Tempo (s)	N° de ciclos		N° de corte		Tempo (s)	N° de ciclos		N° de corte		Tempo (s)
		LP	Int	I	II		LP	Int	I	II		LP	Int	I	II		LP	Int	I	II	
CAB10.2	1	1	1	180	0	0.06	1	1	270	0	0.05	1	4	358	137	0.25	1	1	270	0	0.06
CAB10.4	1	1	1	180	0	0.07	1	1	228	42	0.10	2	4	384	201	0.33	1	1	270	0	0.08
CAB10.6	1	1	1	147	33	0.15	1	1	237	33	0.11	1	5	483	327	0.56	1	1	270	33	0.29
CAB10.8	1	1	1	180	0	0.06	1	1	270	0	0.09	1	3	273	132	0.28	1	1	270	0	0.09
CAB15.2	1	1	2	568	62	0.59	1	1	626	214	0.87	2	6	1350	750	2.09	1	1	840	140	3.86
CAB15.4	1	3	1	763	77	0.66	3	1	850	200	0.52	6	3	609	756	1.78	3	1	1050	160	2.98
CAB15.6	1	2	1	470	160	1.00	2	1	680	160	0.41	4	5	1162	728	6.15	2	1	840	149	2.45
CAB15.8	1	2	1	481	149	0.98	2	1	681	159	0.39	4	5	977	1123	4.55	2	1	840	154	7.29
CAB20.2	1	3	1	1469	51	1.96	3	1	1849	51	1.60	6	2	658	862	1.76	3	1	1900	51	1.91
CAB20.4	1	3	1	1409	111	2.66	3	2	3223	577	5.55	6	2	696	1014	3.64	3	1	1900	146	3.36
CAB20.6	1	3	1	1372	148	2.75	3	1	1910	370	2.81	6	2	799	1101	4.89	3	1	1900	427	6.86
CAB20.8	1	3	1	1262	258	2.71	3	2	2834	966	6.60	6	2	870	1030	5.20	3	1	1900	422	10.77
CAB25.2	1	2	1	1670	130	3.73	3	1	2870	130	3.72	6	2	1079	1321	4.75	3	1	3000	130	5.59
CAB25.4	1	3	1	2284	116	4.68	3	1	2884	116	4.08	6	4	2000	2500	14.09	3	1	3000	302	9.87
CAB25.6	1	3	1	2112	288	5.32	3	1	2712	288	3.84	6	2	1163	1237	7.65	3	1	3000	364	12.07
CAB25.8	1	3	1	2144	256	6.59	3	1	2680	320	4.98	6	2	1430	970	8.91	3	1	3000	312	9.96

Tabela 5.2: Resultados computacionais dos algoritmos usando instâncias CAB com restrições de AG.

Instância	$f_k$	LO					CII					MIS					MFS				
		N° de ciclos		N° de corte		Tempo (s)	N° de ciclos		N° de corte		Tempo (s)	N° de ciclos		N° de corte		Tempo (s)	N° de ciclos		N° de corte		Tempo (s)
		LP	Int	I	II		LP	Int	I	II		LP	Int	I	II		LP	Int	I	II	
CAB10.2	1	1	1	180	0	0.04	1	1	270	0	0.09	1	2	198	27	0.10	1	1	270	0	0.10
CAB10.4	1	1	1	180	0	0.06	1	1	228	42	0.08	1	3	266	139	0.23	1	1	270	0	0.08
CAB10.6	1	1	1	147	33	0.16	1	1	237	33	0.07	1	2	179	46	0.13	1	1	270	33	0.17
CAB10.8	1	1	1	180	0	0.05	1	1	270	0	0.06	1	2	177	48	0.09	1	1	270	0	0.06
CAB15.2	1	1	2	568	62	0.61	1	1	626	214	0.85	1	3	635	205	0.53	1	1	630	91	1.01
CAB15.4	1	2	1	553	77	0.53	3	1	855	195	0.53	2	3	846	309	0.88	2	1	840	160	2.88
CAB15.6	1	2	1	468	162	1.03	2	1	888	162	0.52	2	2	535	305	0.94	2	1	1050	138	2.21
CAB15.8	1	2	1	481	149	0.97	2	1	681	159	0.41	2	2	451	179	0.97	2	1	840	164	7.83
CAB20.2	1	2	1	1089	51	1.77	3	1	1849	51	1.66	3	2	1053	277	1.94	3	1	1900	51	1.96
CAB20.4	1	3	1	1433	87	2.59	3	2	2840	580	4.95	3	2	933	397	2.14	3	1	1900	161	3.72
CAB20.6	1	3	1	1301	219	3.02	3	1	1931	349	2.90	3	3	1400	690	3.42	3	1	1900	346	6.36
CAB20.8	1	3	1	1247	273	2.75	3	2	2826	974	14.14	3	3	1390	700	4.80	3	1	1900	346	9.49
CAB25.2	1	2	1	1670	130	3.49	3	1	2870	130	4.01	3	2	1334	466	3.76	3	1	3000	130	5.58
CAB25.4	1	3	1	2202	198	5.07	3	1	2802	198	4.37	3	3	2275	1025	8.35	3	1	3000	294	9.97
CAB25.6	1	3	1	2134	266	6.74	3	1	2742	258	5.12	3	2	1606	794	7.41	3	1	3000	141	7.66
CAB25.8	1	3	1	2139	261	5.63	3	1	2684	316	5.20	3	3	2118	882	8.65	3	1	3000	292	11.53

Tabela 5.3: Resultados computacionais dos algoritmos usando instâncias AP sem restrições de AG.

Instância	LO					CII					MIS					MFS				
	Nº de ciclos		Nº de corte		Tempo (s)	Nº de ciclos		Nº de corte		Tempo (s)	Nº de ciclos		Nº de corte		Tempo (s)	Nº de ciclos		Nº de corte		Tempo (s)
	LP	Int	I	II		LP	Int	I	II		LP	Int	I	II		LP	Int	I	II	
AP10.2	3	1	246	24	0.18	3	1	426	24	0.11	5	3	279	171	0.22	3	1	450	46	0.40
AP10.4	3	1	250	20	0.20	3	1	430	20	0.13	5	2	180	90	0.07	3	1	450	16	0.16
AP10.6	3	1	255	15	0.18	3	1	423	27	0.16	5	2	270	0	0.06	3	1	450	0	0.10
AP10.8	1	1	180	0	0.09	3	1	450	0	0.13	5	2	270	0	0.06	3	1	450	0	0.10
AP20.2	2	2	1205	315	5.54	5	1	1990	670	2.79	9	3	2070	590	4.68	5	1	2660	540	21.16
AP20.4	2	1	833	307	2.83	5	1	2247	413	2.15	9	2	1318	582	1.86	5	1	2660	505	29.60
AP20.6	2	1	1002	138	1.74	5	1	2480	180	1.95	9	2	1899	1	1.67	5	1	2660	232	27.41
AP20.8	2	1	992	148	1.61	5	1	2476	184	1.79	9	2	1899	1	1.46	5	1	2660	153	9.16
AP30.2	4	1	3302	1048	20.56	7	1	6366	2334	39.74	14	2	4613	1912	22.26	6	1	6960	1230	56.06
AP30.4	4	1	3358	992	15.56	7	1	6590	1240	16.31	14	2	5361	1164	17.49	6	1	6960	1194	69.47
AP30.6	4	1	3633	717	12.52	7	1	6897	933	13.84	14	2	7193	1072	34.29	6	1	6960	943	180.80
AP30.8	4	1	3294	1056	15.81	7	1	6762	1068	14.76	14	2	4789	1736	15.40	6	1	6960	795	533.55
AP40.2	5	1	7136	2224	111.41	7	1	13003	2597	108.69	16	3	13222	3158	229.45	6	1	14040	2820	398.94
AP40.4	5	1	6626	2734	206.26	7	2	14057	6223	524.84	16	6	17108	8632	779.18	6	1	14040	4192	31065.62
AP40.6	5	1	6204	3156	80.57	7	1	9144	8016	293.75	16	5	19877	3523	471.01	a	a	a	a	a
AP40.8	5	1	5770	3590	109.67	7	1	10810	3230	66.36	16	2	11739	1521	88.61	a	a	a	a	a
AP50.2	5	2	12506	4644	407.07	7	3	25011	9289	1112.54	16	2	19692	2358	487.42	6	1	19600	5494	1548.52
AP50.4	5	3	14585	5015	1988.91	7	3	36540	14910	9657.01	16	13	49156	24344	9660.03	a	a	a	a	a
AP50.6	5	2	12827	4323	346.26	7	3	26767	17333	14082.91	16	6	29937	8038	1835.23	a	a	a	a	a
AP50.8	5	1	11371	3329	218.78	7	2	17699	16601	3421.69	16	2	19336	1489	296.98	a	a	a	a	a

<sup>a</sup> Tempo limite excedido

Tabela 5.4: Resultados computacionais dos algoritmos usando instâncias AP com restrições de AG.

Instância	LO					CII					MIS					MFS				
	Nº de ciclos		Nº de corte		Tempo (s)	Nº de ciclos		Nº de corte		Tempo (s)	Nº de ciclos		Nº de corte		Tempo (s)	Nº de ciclos		Nº de corte		Tempo (s)
	LP	Int	I	II		LP	Int	I	II		LP	Int	I	II		LP	Int	I	II	
AP10.2	1	2	270	0	0.10	3	1	426	24	0.12	5	2	328	32	0.12	4	1	450	46	0.37
AP10.4	1	2	270	0	0.16	3	1	430	20	0.16	5	2	264	51	0.09	4	1	450	16	0.26
AP10.6	1	1	165	15	0.20	3	1	423	27	0.17	5	2	315	0	0.11	4	1	450	0	0.10
AP10.8	1	1	180	0	0.06	3	1	450	0	0.15	5	2	315	0	0.11	4	1	450	0	0.10
AP20.2	2	1	825	315	3.63	5	1	1990	670	2.88	9	3	2490	360	4.18	6	1	2660	583	25.58
AP20.4	2	1	833	307	2.57	5	1	2339	321	2.26	9	2	1712	378	2.06	6	1	2660	622	38.33
AP20.6	2	1	1002	138	1.56	5	1	2480	180	2.00	9	2	2089	1	1.90	6	1	2660	232	27.66
AP20.8	2	1	992	148	1.67	5	1	2476	184	1.83	9	2	2089	1	1.65	6	1	2660	153	9.38
AP30.2	4	1	3319	1031	19.00	6	1	6388	2312	48.52	14	2	6295	665	25.60	7	1	6960	1536	64.13
AP30.4	4	1	3523	827	15.15	6	1	5745	1215	15.20	14	2	5376	1584	18.01	7	1	6960	1234	135.42
AP30.6	4	1	3695	655	12.70	6	1	6121	839	12.48	14	2	6047	913	17.29	7	1	6960	939	146.05
AP30.8	4	1	3211	1139	16.68	6	1	5780	1180	13.70	14	2	5683	1277	16.44	7	1	6960	788	7554.22
AP40.2	5	1	7012	2348	155.11	6	1	8937	3543	393.52	16	3	16991	1729	236.66	7	1	14040	2852	476.32
AP40.4	5	2	8024	2896	344.67	6	3	19606	6914	1667.37	16	4	20179	2441	466.38	a	a	a	a	a
AP40.6	5	1	6144	3216	87.10	6	1	9041	4999	242.30	16	3	18094	1406	269.73	7	1	12480	3018	19266.94
AP40.8	5	1	7210	2150	62.37	6	1	7641	4839	77.77	16	3	15057	543	119.59	a	a	a	a	a
AP50.2	5	2	12516	4634	407.15	6	3	26579	12621	1819.99	16	2	22258	1017	491.27	7	1	19600	5228	1527.88
AP50.4	5	2	12030	5120	2032.92	6	2	24535	12215	8561.10	16	4	35044	2931	2691.42	a	a	a	a	a
AP50.6	5	1	10298	4402	374.44	6	1	19411	9989	4684.97	16	3	26914	1261	815.65	a	a	a	a	a
AP50.8	5	1	10266	4434	264.01	6	2	17260	14590	9038.53	16	3	25076	1874	498.42	a	a	a	a	a

<sup>a</sup> Tempo limite excedido

Nestas tabelas são apresentadas a quantidade de iterações (ciclos) de pré-processamento (LP) e de iterações inteiras (Int), o número de cortes tipo I e de cortes tipo II, e o tempo gasto para encontrar a solução ótima para cada variante de Benders. De acordo com as tabelas, o método de decomposição de Benders com MFS apresenta o pior desempenho, não sendo capaz de resolver todas as instâncias dentro do tempo limite. Nos casos em que esta variante não consegue obter a solução ótima, ela também não é capaz de resolver nenhuma iteração inteira. Logo, não é possível obter *gap* de otimalidade quando o tempo limite de 36000s é excedido.

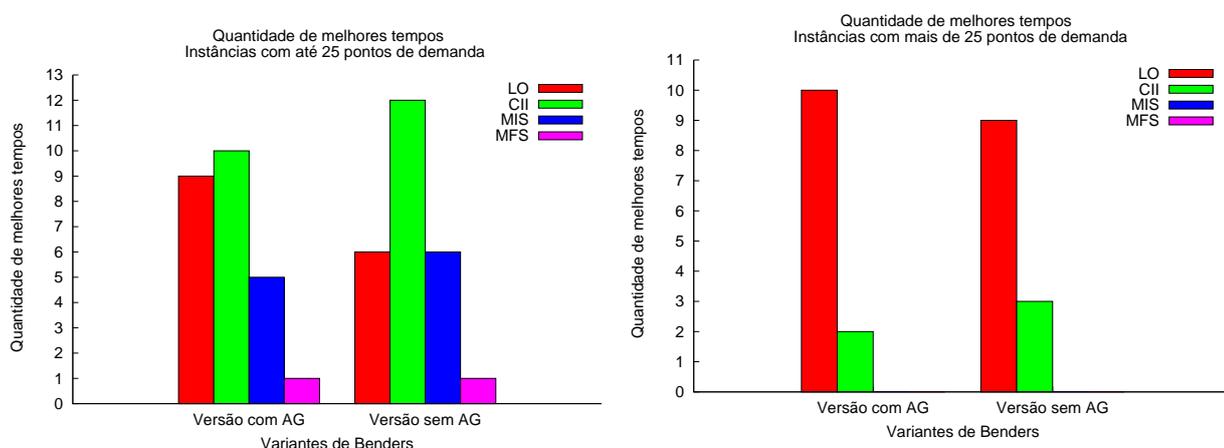


Figura 5.2: Quantidade de melhores de tempos de cada variantes com e sem restrições de AG.

A Figura 5.2 apresenta a quantidade de melhores tempos de cada variante de Benders para as versões com e sem restrições de AG. Estes dados foram divididos em dois gráficos: o primeiro gráfico mostra a quantidade de melhores alcançados por cada variante ao resolver instâncias com até 25 pontos de demandas (instâncias CAB e instâncias AP10 e AP20), enquanto o segundo gráfico apresenta os resultados para instâncias com mais de 25 nós (AP30, AP40 e AP50). De acordo com a figura, a variante Benders CII resolve o maior número de instâncias com até 25 pontos de demanda. No entanto, conforme o tamanho do problema aumenta, instâncias com mais de 25 pontos de demanda, a variante de Benders LO apresenta o melhor desempenho.

A Figura 5.3 apresenta um gráfico com a quantidade de melhores tempos comparando todas as versões de Benders: as quatro variantes de Benders com e sem restrição de AG. Sendo assim, a versão de Benders LO apresenta no total a maior quantidade de melhores tempos, resolvendo uma maior quantidade de instâncias (10) com o menor tempo ao usar restrições de AG. Já as variantes Benders CII e Benders MIS apresentam

uma maior quantidade de melhores tempos ao utilizar a versão sem restrições de AG.

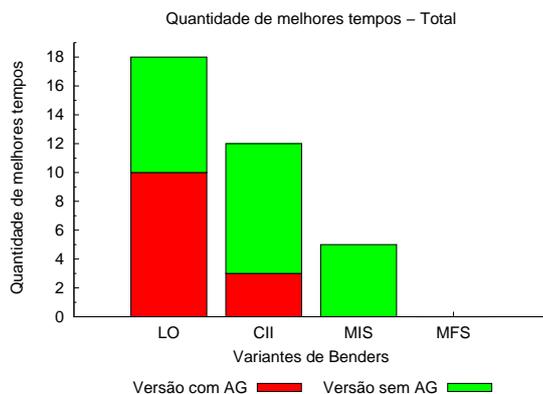


Figura 5.3: Quantidade de melhores de tempos entre todas as versões de Benders.

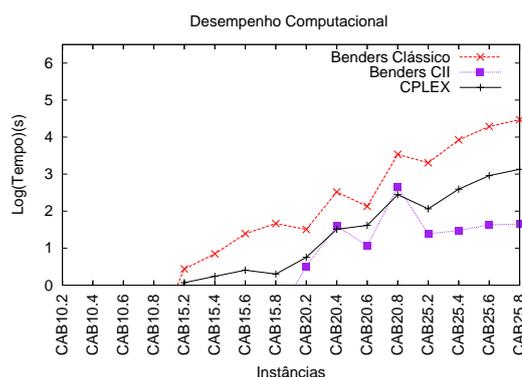


Figura 5.4: Esforço computacional usando instâncias CAB: CPLEX e Benders Clássico versus Benders CII.

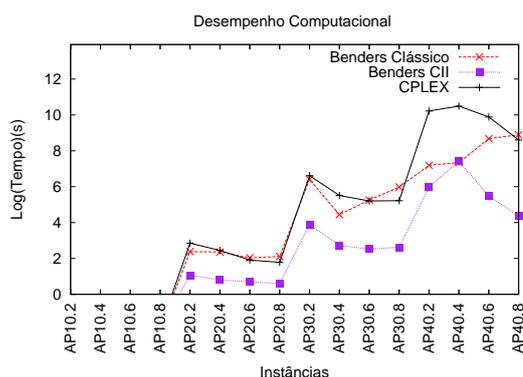


Figura 5.5: Esforço computacional usando instâncias AP: CPLEX e Benders Clássico versus Benders CII.

O desempenho computacional das variantes de Benders apresentadas no capítulo anterior pode ser melhor observado através das Figuras 5.4-5.9.

As Figuras 5.4 e 5.5 apresentam uma comparação do desempenho computacional da variante Benders CII frente à versão de Benders clássico e ao pacote comercial CPLEX. A variante Benders CII se mostra muito mais eficiente que o CPLEX e o método de decomposição de Benders clássico tanto para instâncias AP quanto para instâncias CAB, confirmando a eficácia dos cortes pareto-ótimo para melhorar a convergência do método.

Nas Figuras 5.6-5.9 são apresentados os gráficos com o desempenho computacional das variantes Benders CII, Benders LO, Benders MIS e Benders MFS, com e sem

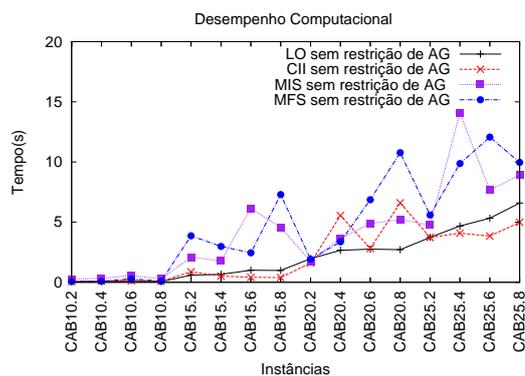


Figura 5.6: Esforço computacional usando instâncias CAB sem restrições de AG.

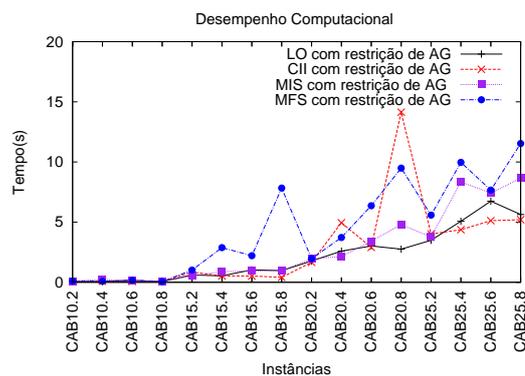


Figura 5.7: Esforço computacional usando instâncias CAB com restrições de AG.

restrições de AG, para os testes usando instâncias CAB e instâncias AP. Nestes gráficos, apenas para efeito de comparação, considera-se que o tempo de resolução das instâncias que excederam o tempo limite, como sendo o próprio tempo limite. De acordo com as figuras, pode-se observar que as variantes Benders CII e Benders LO são bastante competitivas quando testadas em instâncias pequenas, enquanto a versão Benders LO se destaca isoladamente ao resolver instâncias com mais de 30 pontos de demanda. O pior desempenho é apresentado pela variante Benders MFS que consome a maior quantidade de tempo ao resolver as instâncias CAB e AP. Enquanto a variante Benders MIS, que apesar de não superar o Benders LO, é bastante competitiva, principalmente, quando testada em instâncias com mais de 30 pontos de demanda.

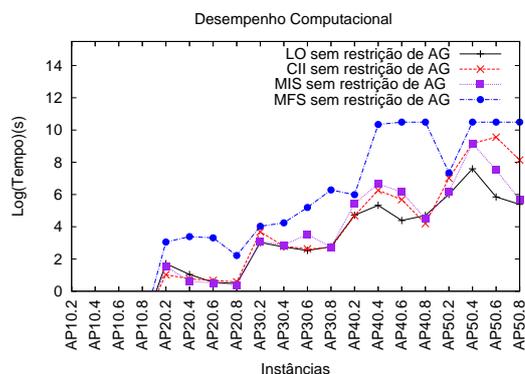


Figura 5.8: Esforço computacional usando instâncias AP sem restrições de AG.

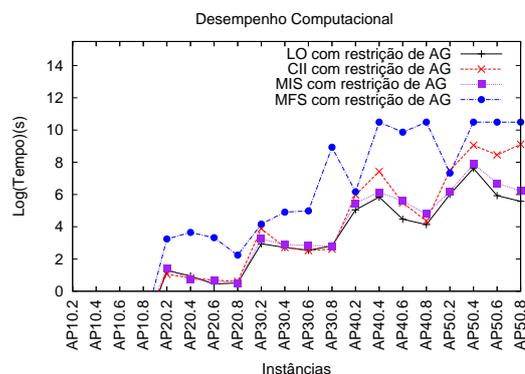


Figura 5.9: Esforço computacional usando instâncias AP com restrições de AG.

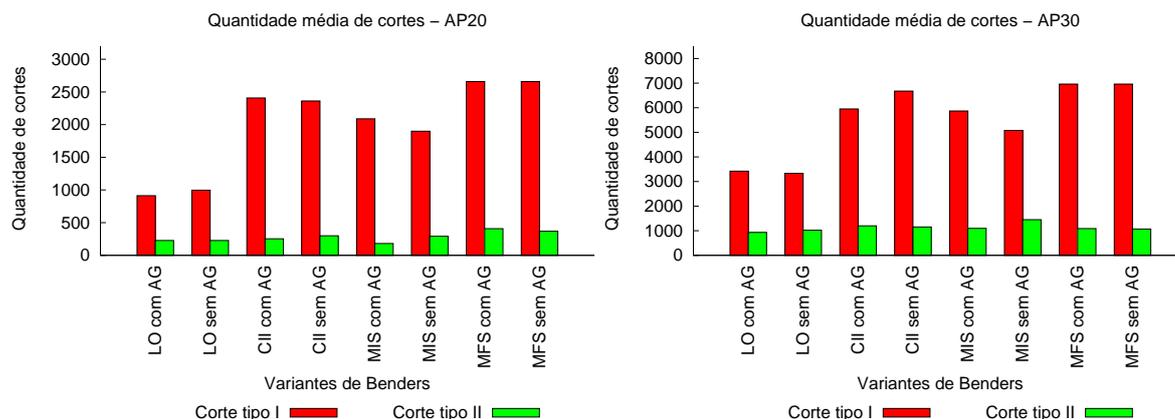


Figura 5.10: Quantidade média de cortes para instâncias AP20.

Figura 5.11: Quantidade média de cortes para instâncias AP30.

Uma outra forma de comparar as quatro variantes de Benders é verificar a quantidade de cortes gerados por cada uma. As Figuras 5.10 e 5.11 apresentam a quantidade média de cortes tipo I e de cortes tipo II para as instâncias AP com 20 e 30 pontos de demanda, respectivamente. A variante Benders MIS gera a menor quantidade média de cortes tipo II para instâncias com 20 pontos de demandas, enquanto a variante Benders LO gera a menor quantidade de cortes tipo II para instâncias com 30 pontos de demandas e gera a menor quantidade de cortes tipo I. Com relação a adição de restrição de AG, pode-se perceber que a variante Benders MIS é mais sensível a esta mudança, uma vez que com a adição destas restrições o número de cortes tipo II diminuem, aumentando a quantidade de cortes tipo I.

Para analisar a convergência de cada variante de Benders foi feito um novo conjunto teste usando as instâncias AP20.2 e AP30.3. Estes testes foram feitos sem o uso das restrições de AG e sem o uso das técnicas: pré-processamento e *Benders-and-cut*. A convergência do LI e do LS das quatro variantes é apresentada nas figuras 5.12 e 5.13. De acordo com estas figuras, a variante Benders LO apresenta a melhor convergência, convergindo para o ótimo antes que as demais variantes; enquanto a variante Benders MFS, apesar de apresentar uma boa convergência ao utilizar a instância AP20.2, apresenta um péssimo desempenho quando testada usando a instância AP30.2, com seu LI demorando um tempo considerável para convergir.

Uma justificativa para o mau desempenho desta variante pode ser encontrado na Figura 5.14, que mostra a parcela de tempo total gasto por cada algoritmo para resolver o PM e os subproblemas. Observando a figura, esta variante gasta uma parcela de tempo grande resolvendo os subproblemas, lembrando que dentre os subproblemas

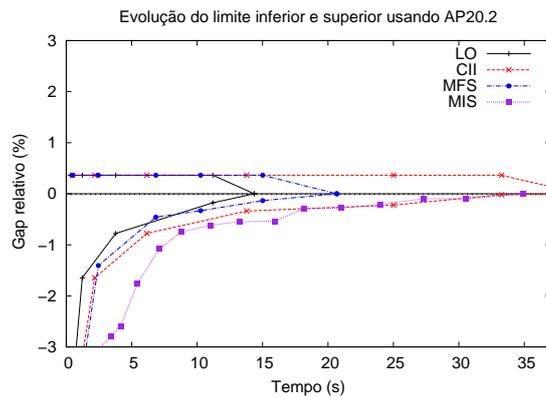


Figura 5.12: Convergência dos limites usando a instância AP20.2.

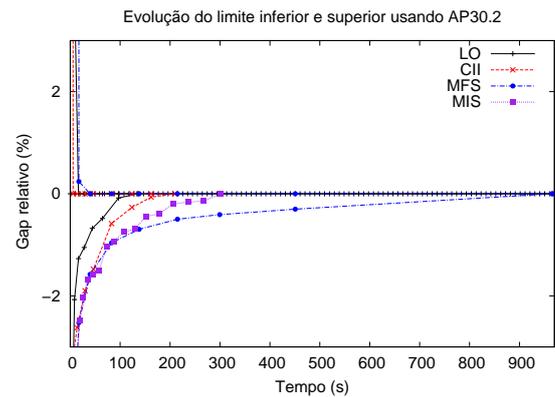


Figura 5.13: Convergência dos limites usando a instância AP30.2.

estão os subproblemas para encontrar o MFS, que são problemas inteiros.

Uma outra questão interessante para se observar na Figura 5.14 é o tempo gasto pelas variantes Benders CII e Benders LO para resolver esses problemas. Devido ao fato de resolver um problema adicional, para o cálculo do  $\lambda$ -ótimo, a variante Benders LO gasta mais tempo resolvendo subproblemas que a variante Benders CII. Entretanto, a variante Benders LO gasta menos tempo resolvendo o PM, confirmando a efetividade dos cortes pareto-ótimo gerados usando o  $\lambda$ -ótimo. De acordo com as análises anteriores pode-se concluir que para pequenas instâncias, o Benders CII apresenta o melhor desempenho, não compensando a resolução de um problema adicional; no entanto

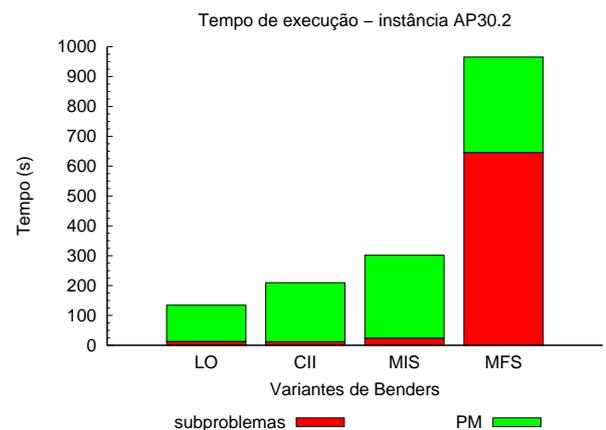
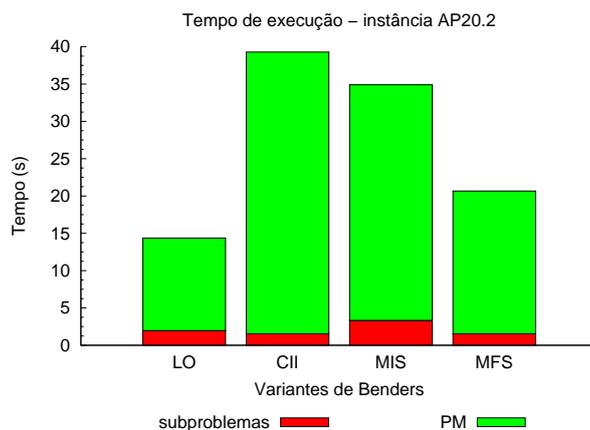


Figura 5.14: Tempo total de execução dos PMs e dos subproblemas usando as instâncias AP20.2 e AP30.2.

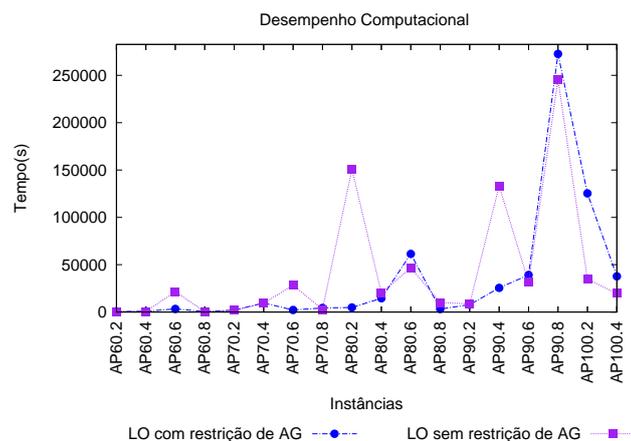


Figura 5.15: Esforço computacional - variante Benders LO com e sem restrição de AG.

conforme o tamanho do problema aumenta a resolução de um problema adicional para gerar cortes mais eficiente é compensada pela redução do tempo de resolução do PM.

Com o objetivo de verificar a eficiência da variante Benders LO para resolver o PLCATP, foram feitos testes usando instâncias AP com 60, 70, 80, 90 e 100 pontos de demanda. Para cada teste foi estabelecido um tempo limite de 144 h. A Tabela 5.5 apresenta os resultados dos testes para as versões com e sem restrições de AG. De acordo com a tabela, a variante Benders LO se mostra bastante eficiente ao resolver instâncias com até 100 pontos de demandas. O desempenho computacional da variante Benders LO pode ser observado na Figura 5.15, enquanto o número de cortes de Benders gerados se encontram na Figura 5.16. De acordo com as figuras, a versão do Benders LO com restrição de AG, apresenta um melhor desempenho computacional para a maior parte das instâncias e gera uma menor quantidade de cortes, tanto tipo I quanto tipo II, que a versão sem restrição de AG. Apesar desta última versão se destacar ao resolver as instâncias de testes AP90.6, AP90.8, AP100.2 e AP100.4.

Tabela 5.5: Resultados computacionais usando instâncias AP - LO.

Instância	ff	Nº de Conc.	LO com AG							LO sem AG						
			Nº de ciclos		Nº de corte		Tempo		GAP (%)	Nº de ciclos		Nº de corte		Tempo		GAP (%)
			LP	Int	I	II	LP	Int		LP	Int	I	II	LP	Int	
AP60.2	2	5	5	1	18928	2312	493.32	587.99	-	5	1	18590	2650	468.14	542.10	-
AP60.4	2	4	5	1	16106	5134	707.13	842.70	-	5	1	17361	3879	623.31	660.75	-
AP60.6	2	3	5	2	19379	5401	633.59	3341.47	-	5	2	18667	6113	667.31	21472.30	-
AP60.8	2	2	5	1	18505	2735	497.61	566.03	-	5	1	18426	2814	448.42	504.41	-
AP70.2	2	6	5	1	24204	4776	1487.60	1976.91	-	5	2	29047	4763	1277.16	1903.16	-
AP70.4	2	6	5	3	32563	6077	1598.88	9557.76	-	5	3	33509	5131	1242.69	9223.97	-
AP70.6	2	4	5	1	20864	8116	1613.23	2195.93	-	5	2	26513	7297	1580.44	28398.73	-
AP70.8	2	3	5	2	26071	7739	2054.29	4316.14	-	5	2	26150	7660	2121.07	2753.74	-
AP80.2	2	6	5	1	30836	7084	3145.44	4857.15	-	5	2	36712	7528	2529.10	151117.14	-
AP80.4	2	5	5	3	42790	7770	2986.15	14611.02	-	5	1	29684	8236	2915.68	20180.10	-
AP80.6	2	4	5	2	33376	10864	4551.91	61428.20	-	5	3	39650	10910	4232.72	46637.13	-
AP80.8	2	2	5	1	27642	10278	2883.22	3179.87	-	5	2	31045	13195	3694.75	10100.73	-
AP90.2	2	8	5	1	40081	7979	3838.62	7744.44	-	5	1	39220	8840	4588.92	8615.71	-
AP90.4	2	7	5	3	51739	12341	5139.20	25524.28	-	5	4	58230	13860	5093.58	133289.20	-
AP90.6	2	5	5	2	39944	16126	5833.20	39136.53	-	5	2	39901	16169	5976.59	31599.92	-
AP90.8	2	3	5	1	30773	17287	5642.56	272659.53	-	5	2	38759	17311	6103.50	245679.52	-
AP100.2	2	7	5	2	56939	12361	8157.63	125300.54	-	5	3	66040	13160	6797.66	34939.66	-
AP100.4	2	7	5	3	64208	14992	8407.11	37744.49	-	5	2	53531	15769	8196.17	19717.63	-
AP100.6	2	a	a	a	a	a	a	a	0.8273	a	a	a	a	a	a	b
AP100.8	2	a	a	a	a	a	a	a	b	a	a	a	a	a	a	b

<sup>a</sup> Tempo limite excedido.

<sup>b</sup> Não foi obtido GAP.

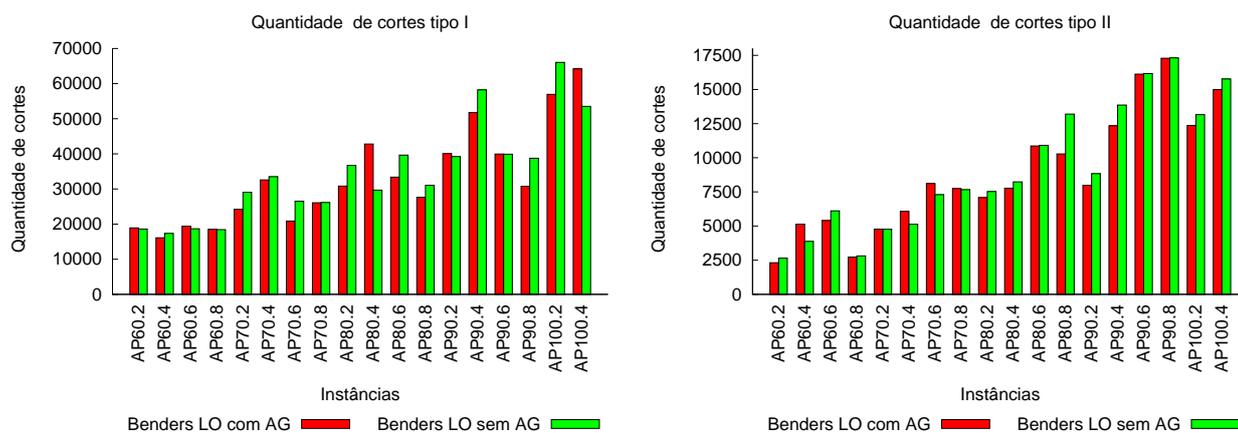


Figura 5.16: Quantidade de cortes gerados pela variante Benders LO com e sem restrição de AG.

Esses resultados são bastante satisfatórios, uma vez que mostra que o método de decomposição de Benders, usando a metodologia desenvolvida neste trabalho, é capaz de resolver instâncias de grande porte do problema alvo deste trabalho de forma exata.

# Capítulo 6

## Conclusão

*“Problemas não são obstáculos, mas oportunidades ímpares de superação e evolução”*

Nelson Rodrigues

Neste trabalho um modelo baseado na formulação proposta por [Contreras et al. \(2009\)](#) foi utilizado para modelar o sistema de transporte público. Em função da complexidade do problema e dos altos valores financeiros envolvidos, o método de decomposição de Benders, que é um método exato, foi utilizado para resolução do problema. No entanto resultados computacionais mostram que a versão clássica de Benders só consegue resolver problemas de pequeno porte.

Para acelerar a convergência do método e resolver instâncias maiores, uma versão do método de decomposição de Benders usando cortes pareto-ótimo foi apresentada. Entretanto, como ao utilizar a formulação para o PLCATP pode-se gerar os dois tipos de cortes, corte tipo I e corte tipo II, duas variantes do método de decomposição de Benders voltadas para esta classe de problema são apresentadas. Além disso, uma nova técnica para melhorar a geração de cortes pareto-ótimo, nomeada  $\lambda$ -ótimo, é proposta. Resultados Computacionais comprovam que estas técnicas melhoram a convergência do método e que a nova técnica, utilizando o  $\lambda$ -ótimo, apresenta um desempenho superior as outras três metodologias.

Uma conclusão deste trabalho é que o método de decomposição é adequado para resolver o problema em questão. Além disso, pode-se concluir que adição de cortes pareto-ótimo e a técnica que gera cortes baseado no MIS são realmente eficazes para melhorar o desempenho do método. Sendo que o melhor desempenho é apresentado pela técnica proposta neste trabalho,  $\lambda$ -ótimo.

Como possíveis trabalhos futuros, pode-se citar:

- Desenvolvimento de outros modelos para o transporte público. Exemplo: formulações com alocação múltipla; modelo que relaxa a exigência de rede de concen-

---

tradores em forma de árvore ou que simplesmente permita conexões diretas.

- Ao invés de se utilizar um fator de desconto fixo, seria conveniente utilizar um fator de desconto dependente da quantidade de fluxo como proposto por [O'Kelly e Bryan \(1998\)](#).
- Considerar o efeito do congestionamento no projeto da rede.

# Referências Bibliográficas

- Alumur, S. e Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190:1–21.
- Alumur, S. A.; Kara, B. Y. e Karasan, O. E. (2009). The design of single allocation incomplete hub networks. *Transportation Research Part B: Methodological*, 43(10):936–951.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerisch Mathematik*, 4:238–252.
- Bryan, D. L. e O’Kelly, M. E. (1999). Hub-and-spoke networks in air transportation: an analytical review. *Journal of Regional Science*, 2(30):275–295.
- Calik, H.; Alumur, S. A.; Kara, B. Y. e Karasan, O. E. (2009). A tabu-search based heuristic for the hub covering problem over incomplete hub networks. *Computers & Operations Research*, 36(12):3088–3096.
- Camargo, R. S.; Miranda Jr, G. e Luna, H. P. (2008). Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers and Operations Research*, 35:1047–1064.
- Camargo, R. S.; Miranda Jr, G. e Luna, H. P. (2009). Benders decomposition for hub location problems with economies of scale. *Transportation Science*, 43:86–97.
- Campbell, J. F. (1992). Location and allocation for distribution systems with transshipments and transportation economies of scale. *Annals of Operations Research*, 40:77–99.
- Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72:387–405.
- Campbell, J. F.; Ernst, A. T. e Krishnamoorthy, M. (2002). Hub location problems. In Drezner, Z. e Hamacher, H. W., editores, *Facility Location: Applications and Theory*, chapter 12, pp. 373–407. Springer, 1ª edição.

- Campbell, J. F.; Ernst, A. T. e Krishnamoorthy, M. (2005a). Hub arc location problems: Part i - introduction on results. *Management Science*, 51(10):1540–1555.
- Campbell, J. F.; Ernst, A. T. e Krishnamoorthy, M. (2005b). Hub arc location problems: Part ii - formulations and optimal algorithms. *Management Science*, 51(10):1556–1571.
- Cetiner, S.; Sepil, C. e Sural, H. (2006). Hubbing and routing in postal delivery systems. Technical report, Middle East Technical University, Industrial Engineering Department, 06532 Ankara, Turkey.
- Contreras, I.; Fernández, E. e Marín, A. (2009). Tight bounds from a path based formulation for the tree of hub location problem. *Computer and Operations Research*, 36:3117–3127.
- Contreras, I.; Fernández, E. e Marín, A. (2010). The tree of hubs location problem. *European Journal of Operational Research*, 202:390–400.
- Dutta, A. e Kubat, P. (1999). Design of partially survivable networks for cellular telecommunication systems. *European Journal of Operational Research*, 118:52–64.
- Ebery, J.; Krishnamoorthy, M.; Ernst, A. e Boland, N. (2000). The capacitated multiple allocation hub location problema: Formulations and algorithms. *European Journal of Operational Research*, 120:614–631.
- Elhedhli, S. e Hu, F. X. (2005). Hub-and-spoke network design with congestion. *Computers & Operations Research*. To appear.
- Ernst, A. T. e Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4:139–154.
- Fischetti, M.; Salvagnin, D. e Zanette, A. (2008). Minimal infeasible subsystems and benders cuts. Technical report, DEI University of Padova, [http://www.dei.unipd.it/fisch/papers/MIS\\_for\\_Benders.pdf](http://www.dei.unipd.it/fisch/papers/MIS_for_Benders.pdf).
- Gelareh, S. (2008). *Hub Location Models in Public Transport Planning*. PhD thesis, University of Saarlandes, Germany.
- Geoffrion, A. M. e Graves, G. W. (1974). Multicommodity distribution system design by Benders decomposition. *Management Science*, 20:822–844.
- Jaillet, P.; Song, G. e Yu, G. (1996). Airline network design and hub location problems. *Location Science*, 4:195–211.

- Kara, B. Y. e Tansel, B. C. (2003). The latest arrival hub location problem. *Management Science*, 47:1408–1420.
- Klincewicz, J. G. (1998). Hub location in backbone/tributary network design: a review. *Location Science*, 6:307–335.
- Maculan, N.; Plateau, G. e Lissier, G. (2003). Integer linear models with a polynomial number of variables and constraints for some classical combinatorial optimization problems. *Pesquisa Operacional*, 23(1):161–168.
- Magnanti, T. L. e Wong, R. T. (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.
- Mercier, A.; Cordeau, J. F. e Soumis, F. (2005). Acomputational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers and Operations Research*, 32(6):1451–1476.
- Nickel, S.; Schobel, A. e Sonneborn, T. (2001). Hub location problems in urban traffic networks. In Niittymähi e Pursula, editores, *Mathematical Methods and Optimisation in Transportation Systems*, pp. 1–12. Kluwer Academic Publisher.
- O’Kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32:393–404.
- O’Kelly, M. E. (1992). Hub facility location with fixed costs. *Papers in Regional Science*, 71(3):293–306.
- O’Kelly, M. E. (1998). A geographer’s analysis of hub-and-spoke networks. *Journal of Transport Geography*, 3(6):171–186.
- O’Kelly, M. E. e Bryan, D. L. (1998). Hub location with flow economies of scale. *Transportation Research Part B*, 32(8):605–616.
- Papadakos, N. (2008). Practical enhancements to the magnanti–wong method. *Operations Research Letters*, 36:444–449.
- Rei, W.; Gendreau, M.; Cordeau, J.-F. e Soriano, P. (2006). Accelerating benders decomposition by local branching. *INFORMS Journal on Computing*. In press.
- Saharidis, G. K. e Ierapetritou, M. G. (2010). Improving benders decomposition using maximum feasible subsystem (mfs) cut generation strategy. *Computers and Chemical Engineering*, 34:1237–1245.

- Sasaki, M.; Suzuki, A. e Drezner, Z. (1999). On the selection of hub airports for an airline hub-and-spoke. *Computer & Operations Research*, 26:1411–1422.
- Sung, C. e Jin, H. (2001). Dual-based approach for a hub network design problem under non-restrictive policy. *European Journal of Operational Research*, 132:88–105.
- Yamana, H. e Carello, G. (2005). Solving the hub location problem with modular link capacities. *Computers & Operations Research*, 32:3227 – 3245.
- Yoon, M. e Current, J. (2008). The hub location and network design problem with fixed and variable arc costs: formulation and dual-based solution heuristic. *Journal of the Operational Research Society*, 59(1):80–89.