ALGORITMO EXATO APLICADO AO PROBLEMA DE LOCALIZAÇÃO DE CONCENTRADORES EM ÁRVORE SOB CONGESTIONAMENTO

Elisangela Martins de Sá
elisangeladep@ufmg.br

Ricardo Saraiva de Camargo
rcamargo@dep.ufmg.br

Gilberto Miranda Junior
miranda@dep.ufmg.br

UFMG - Programa de Pós Graduação em Engenharia de Produção
Av. Antônio Carlos, 6627 - Pampulha
CEP: 31270-901 - Belo Horizonte - Minas Gerais - Brasil

Resumo

Este trabalho aborda o projeto de uma rede eixo-raio com topologia de árvore que leva em consideração os efeitos do congestionamento no projeto da rede. Este tipo de topologia de rede tem um grande apelo em sistemas de transporte e redes de telecomunicações onde o custo para instalar cada conexão é muito alto, sendo conveniente projetar uma rede com o menor número de conexões, ou seja, uma árvore. Uma formulação de programação não-linear inteira mista é apresentada para modelar o problema. Para resolver o problema, são propostos um algoritmo baseado em aproximação externa e um algoritmo híbrido que integra o método de decomposição de Benders e um algoritmo de aproximação externa. Os resultados dos teste computacionais mostram a eficiência do algoritmo proposto comparado ao aplicativo comercial CPLEX.

Palavras chave: Localização de concentradores em árvore; Congestionamento; Aproximação externa; Método de decomposição de Benders.

Abstract

This paper addresses the design of a hub-and-spoke network with a tree topology taking into account the congestion effect during the network design. This kind of network topology have a great appeal in transportation systems and telecommunication networks where the cost to install each connection is very large, being convenient to design a network with fewest connections, that is, a tree. We present a non-linear programming formulation to model the problem. In order to solve it, a algorithm based

on outer approximation and Benders decomposition method is proposed. The computational results shows the efficiency of the proposed algorithms compared with the general purpose solver CPLEX.

KEYWORDS: Tree of hub location problem; Congestion; Outer approximation; Benders decomposition method.

1 Introdução

Um sistema eixo-raio é uma topologia de rede usada para rotear fluxo entre vários pontos de origem e de destino. Neste tipo de rede, ao invés de se estabelecer uma conexão direta entre cada par de origem e de destino, os pontos de demandas são conectados a instalações intermediárias, conhecidas como concentradores. Nos sistemas eixo-raio, fluxos de diferentes origens são agrupados nos concentradores, antes de serem encaminhados, possivelmente através de concentradores intermediários, ao seu destino final. Desta forma, os concentradores funcionam então como pontos de triagem, agregação, roteamento e distribuição. A combinação da agregação dos fluxos de demanda nos concentradores e seu roteamento na rede de concentradores, permitem o uso de meios de transportes maiores e mais eficientes reduzindo o custo por unidade transportada. Em outras palavras, economia de escala pode ser obtida (O'Kelly, 1986, 1987).

Este trabalho aborda o projeto de uma rede eixo-raio com topologia de árvore. O problema de localização de concentradores em árvore foi proposto por Contreras et al. (2009) e consiste em projetar uma rede eixo-raio onde os concentradores são interconectados por meio de uma árvore e cada ponto de demanda é alocado a um único concentrador (Figura 1). Este tipo de topologia é adequado para modelar sistemas de transporte ou redes de telecomunicações onde o custo para instalar uma conexão entre quaisquer pares de concentradores é muito alto, sendo conveniente projetar uma rede conexa, com o menor número de conexões possíveis, isto é, uma 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).

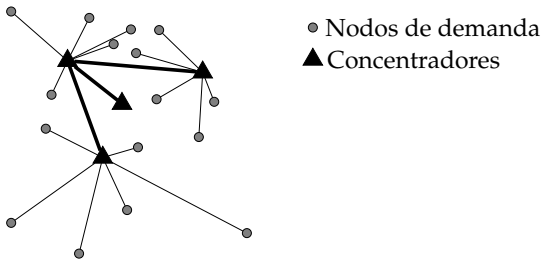

Figura 1: Ilustração de uma rede eixo-raio com topologia de árvore.

Apesar da potencial aplicação prática, esta área de pesquisa ainda é bem recente, existindo, portanto, poucos trabalhos abordando esta topologia de rede. Gelareh (2008) aborda o projeto de uma rede eixo-raio com topologia de árvore no contexto de projeto de rede de transporte público apresentando uma formulação matemática para modelar o problema cujo objetivo consiste em minimizar o custo total de transporte e o custo total de infraestrutura. Desconsiderando o custo de instalação de infraestrutura, Contreras et al. (2009, 2010) apresenta uma variante p-mediana do PLCA, onde o número de concentradores a serem instalados já é conhecido a priori. Com o intuito de propor métodos exatos para resolver o problema de larga escala em um tempo razoável, Sá (2011) propõe vários algoritmos baseados no método de decomposição de Benders (Benders, 1962). Onde, instâncias

com até 100 pontos de demandas são resolvidas de forma exata.

Este trabalho propõe o problema de localização de concentradores sob congestionamento que pretende explorar o efeito do congestionamento ao projetar a rede. Este tipo de abordagem é extremamente relevante uma vez que ao se projetar uma rede eixo-raio sem levar em conta os efeitos do congestionamento pode se subestimar o real custo do sistema. Além disso, devido aos grandes recursos financeiros envolvidos, bem como o impacto direto nos custos operacionais e a eficiência dos serviços oferecidos, é importante modelar o sistema a ser projetado de forma mais realística possível (Gelareh, 2008). Assim, é fundamental abordar os efeitos do congestionamento durante o projeto do sistema.

Neste trabalho, o cálculo do fluxo que contribui para o congestionamento é baseado na abordagem de Ernst et al. (2009); Elhedhli & Hu (2005) que consideram apenas os fluxos oriundos da rede de local. Em um contexto de transporte público, como por exemplo em sistemas de metrôs ou trem de alta velocidade, este congestionamento é representado pela fila para compra do bilhete de embarque.

Este trabalho está organizado da seguinte forma: a próxima seção apresenta uma formulação para modelar o problema; A seção 3 apresenta uma algoritmo baseado em aproximação externa e um algoritmo híbrido baseado em aproximação externa e decomposição de Benders para resolver o problema. Os resultados dos testes computacionais comparando os algoritmos proposto com o aplicativo CPLEX são apresentados na seção 4. Finalmente, a seção 5 apresenta uma conclusão deste trabalho.

2 Definições e formulação

Seja N e K os conjuntos de pontos de demanda e de pontos candidatos a tornar-se concentradores, respectivamente. Neste trabalho considera-se que $K \equiv N$, isto é, todos os pontos de demandas são candidatos em potenciais a concentradores. Para simplificar a notação, no restante deste trabalho, os índice i e j $(i, j \in N)$ serão usados para denotar os pontos de origem e de destino, e os índices k e m $(k, m \in K)$, concentradores.

Seja w_{ij} a quantidade de fluxo que deve ser roteada da origem i até o destino j. Sendo assim, uma vez que a rede de concentradores consiste em uma árvore não-dirigida e cada ponto não-concentrador é alocado diretamente a um único concentrador, então cada par de origem-destino $i - j$ está conectado por um único caminho. Em outras palavras, as demandas w_{ij} e w_{ji} são roteadas através do mesmo caminho, sendo necessário, portanto, considerar apenas o caminho onde $i < j$. Para simplificar a notação defini-se, também, $O_i = \sum_j w_{ij}$ e $D_i = \sum_j w_{ji}$, o fluxo total que tem a sua origem e destino, respectivamente, em i.

Com relação aos parâmetros do problema relacionados a custos, associa-se à cada conexão entre dois pontos i e j $(i, j \in N)$, um custo unitário de transporte $c_{ij} > 0$. Caso i e j sejam concentradores então um fator de desconto $\alpha (0 \leq \alpha \leq 1)$, representando a economia de escala, é aplicado, resultando no seguinte custo unitário de transporte: αc_{ij}. Além do custo de transporte, defini-se por f_k o custo de localizar um concentrador no ponto k.

Com a finalidade de levar em consideração os efeitos do congestionamento, um custo não-linear proveniente do congestionamento é adicionado à função objetivo . Para modelar o crescimento explosivo dos custo de congestionamento será utilizada a função *power*

law, introduzida por (Gillen & Levinson, 1999), muito utilizada para modelar congestionamento em redes eixo-raio (Elhedhli & Hu, 2005; R. S. Camargo et al., 2009). Seja $g_k \geq 0$ uma variável de decisão que calcula o fluxo no concentrador oriundo da rede local, então a função *power law* é dada por:

$$\tau(g_k) = a g_k^b,$$

onde a e b são constante positivas ($b \geq 1$).

Além disso, defini-se as variáveis de fluxo $x_{ijkm} \geq 0$ para representar a quantidade de fluxo com o origem em i e destino j que é roteada através da conexão $k-m$; e variáveis inteiras $z_{ik} \in \{0,1\}$ e $y_{km} \in \{0,1\}$ que indicam se ponto i é alocado ao concentrador k ($z_{ik} = 1$) ou não ($z_{ik} = 0$) e se a conexão entre concentradores $k-m (k < m)$ é utilizado para ligar os concentradores k e m ($y_{km} = 1$) ou não ($y_{km} = 0$), respectivamente. Considera-se que se um concentrador é instalado no ponto k, então $z_{kk} = 1$, caso contrário, $z_{kk} = 0$. Sendo assim, a formulação para o problema de localização de concentradores em árvore sob congestionamento, baseada na formulação proposta por Contreras et al. (2009), é dada por:

$$\min \sum_k [f_k z_{kk} + \tau(g_k)] + \sum_i \sum_{k:k\neq i} (O_i + D_i) c_{ik} z_{ik} + \sum_i \sum_{j:i<j} \sum_k \sum_{m:m\neq k} (\alpha c_{km} w_{ij} + \alpha c_{mk} w_{ji})\, x_{ijkm} \quad (1)$$

Sujeito a:

$$\sum_k z_{ik} = 1 \qquad \forall i \quad (2)$$

$$z_{mk} + y_{km} \leq z_{kk} \qquad \forall k < m \quad (3)$$

$$z_{mk} + y_{mk} \leq z_{kk} \qquad \forall k < m \quad (4)$$

$$\sum_k \sum_{m:m>k} y_{km} = \sum_k z_{kk} - 1 \quad (5)$$

$$g_k = \sum_i O_i z_{ik} \qquad \forall k \quad (6)$$

$$\sum_{k:k\neq m} x_{ijkm} + z_{im} = \sum_{r:r\neq m} x_{ijmr} + z_{jm} \qquad \forall i < j,\ m \quad (7)$$

$$\sum_{k:k\neq m} x_{ijmk} \leq z_{mm} \qquad \forall i < j,\ m \quad (8)$$

$$\sum_{k:k\neq m} x_{ijkm} \leq z_{mm} \qquad \forall i < j,\ m \quad (9)$$

$$x_{ijkm} + x_{ijmk} \leq y_{km} \qquad \forall i, j, k < m \quad (10)$$

$$x_{ijkm} \geq 0 \qquad \forall\ i, j, k \neq m \quad (11)$$

$$g_k \geq 0 \qquad \forall\ k \quad (12)$$

$$y_{km}, z_{ik} \in \{0,1\} \qquad \forall\ i, k < m. \quad (13)$$

Onde a função (1) minimiza o custo de instalação dos concentradores, custos de congestionamento e o custo total de transporte. As Restrições (2) asseguram que cada ponto não-concentrador seja alocado a um único concentrador. Restrições (3) e (4) só admitem conexões envolvendo o concentrador k, conexões entre um ponto não-concentrador e um concentrador ou conexões entre dois concentradores, se ele estiver instalado. Restrição (5) é necessária para garantir a formação de uma árvore. Restrições (6) contabiliza o fluxo local que entra no concentrador k. Restrições (7) asseguram o balanceamento do fluxo.

Restrições (8) e (9) garantem que o fluxo entre cada par de pontos i, j pode usar apenas concentradores instalados. Restrições (10) garantem que o fluxo entre cada par de pontos i, j pode usar uma conexão entre concentradores somente se esta conexão estiver instalada. Por fim, a restrições (11)-(13) definem o domínio das variáveis do problema.

3 Método de resolução

Devido as características da formulação proposta como não-linearidade e decomponibilidade do problema caso as variáveis binárias sejam fixadas um algoritmo híbrido baseado no método de decomposição de Benders (Benders, 1962) e aproximação externa (Duran & Grossmann, 1986; Fletcher & Leyffer, 1996) é proposto para resolver o problema. A ideia básica deste algoritmo, proposto por R. S. de Camargo et al. (2011), consiste na resolução do problema não-linear utilizando a técnica de aproximação externa (OA), do inglês *outer aproximation*, o que resulta em uma formulação de PLIM com um grande número de restrições e variáveis. Esta formulação pode então ser resolvida através da projeção das variáveis fracionárias x_{ijkm} por meio de um procedimento de planos cortante, como o método de decomposição de Benders (BD). A ideia central de ambos os métodos é decompor o problema original em três problemas menores, problema mestre (PM) e dois subproblema, SP OA e SP BD, resolvendo-os iterativamente até que o limite inferior originário do PM convirja para o limite superior proveniente dos subproblemas.

3.1 Aproximação externa (OA)

Ao fixar o vetor de variáveis $(z, y, x) = (z^h, y^h, x^h)$, em uma dada iteração h, este problema pode ser reduzido a um problema de programação não-linear puro que é dado por:

$$\min \sum_k \tau(g_k) \tag{14}$$

$$\text{s.t.: } g_k = \sum_i O_i z_{ik}^h \qquad \forall k \tag{15}$$

$$g_k \geq 0 \qquad \forall\, k. \tag{16}$$

O subproblema acima é trivial de ser resolvido uma vez que o custo de congestionamento está unicamente definido a partir das variáveis z_{ik}. A partir de uma solução g^h do (sub)problema acima pode-se inferir o gradiente da função $\tau(g)$ em (z^h, y^h, x^h, g^h). Uma vez que a não-linearidade da formulação proposta está restrita apenas a função objetivo, então o problema (1)-(13) pode ser reformulada através de aproximação externa, dando origem ao seguinte PM OA:

$$\min \sum_k [f_k z_{kk} + \xi_k] + \sum_i \sum_{k:k\neq i} (O_i + D_i) c_{ik} z_{ik} + \sum_i \sum_{j:i<j} \sum_k \sum_{m:m\neq k} (\alpha c_{km} w_{ij} + \alpha c_{mk} w_{ji})\, x_{ijkm} \tag{17}$$

$$\text{s.t.: } (2)-(13) \tag{18}$$

$$\xi_k \geq \tau(g_k^h) + \tau'(g_k^h)(g_k^h - g_k) \qquad \forall k \in N, h \in \mathbf{H} \tag{19}$$

$$\xi_k, \geq 0 \qquad \forall k. \tag{20}$$

Onde, **H** é o número de iterações e as restrições (19) são os cortes OA.

Definindo, LS e LI como sendo os limites inferiores e superiores, respectivamente, então um método de aproximação externa para resolver o problema proposto é apresentado no Algoritmo 1.

Algoritmo 1: Algoritmo OA

1 Faça $LS = +\infty$, $LI = -\infty$
2 Se $LS - LI < \epsilon$, então pare. Fim da execução, a solução ótima foi obtida.
3 Resolva o PM OA, obtendo o valor ótimo das variáveis z_{ik}.
4 Faça $LI = S^*_{PM}$ e atualize z_{ik} no subproblema (14)-(16).
5 Adicione os corte OA (19) ao PM OA.
6 Faça: $LS = \min\{LS, SP^* + \sum_k f_k z_{kk} + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik} + \sum_i \sum_{\substack{j \\ i<j}} \sum_k \sum_{\substack{m \\ m \neq k}} (\alpha c_{km} w_{ij} + \alpha c_{mk} w_{ji})\ x_{ijkm}\}$.

Uma vez que a formulação para o PM OA é uma formulação de PLIM com estrutura decomponível, ou seja, ao fixar as variáveis complicantes z_{ik} e y_{km}, o problema resultante pode ser decomposto em $(n^2 - n)/2$ subproblemas de roteamento; logo o método de decomposição de Benders parece ser adequado para resolver o problema.

3.2 Método de decomposição de Benders (BD)

O método de decomposição de Benders (Benders, 1962) foi proposto para resolver problemas de programação matemática que possui um conjunto de variáveis ditas complicantes, isto é, dado que este conjunto de variáveis esteja fixado problema resultante é mais maleável. A ideia básica do método é decompor o problema original em dois problemas mais simples: um problema composto pelas variáveis inteiras e uma variável fracionária adicional, conhecido como problema mestre, e um problema linear, conhecido como subproblema de Benders, que é problema original com as variáveis complicantes fixadas. O algoritmo resolve os dois problemas iterativamente. Gerando a cada iteração uma restrição, conhecida como corte de Benders, que é adicionado ao PM. O algoritmo para quando os limites inferiores (LI), solução ótima do problema mestre, e superiores (LS) obtido ao resolver o SP convirjam para a solução ótima do problema original, ou seja, LI=LS.

Sendo assim, ao fixar as variáveis inteiras z_{ik}, y_{km}, tem-se o seguinte o seguinte subproblema de Benders (SP BD):

$$\max \sum_{j:i<j} [\sum_{m} [(z^h_{jm} - z^h_{im}) u_{ijm} - z^h_{mm}(s_{ijm} + t_{ijm})] - \sum_{k} \sum_{m:m>k} y^h_{km} e_{ijkm}] \tag{21}$$

$$\text{s. a: } u_{ijm} - u_{ijk} - e_{ijkm} - s_{ijk} - t_{ijm} + (w_{ij} + w_{ji}) v_{ikm} \leq \alpha(c_{km} w_{ij} + c_{mk} w_{ji}) \quad \forall\, k < m \tag{22}$$

$$u_{ijm} - u_{ijk} - e_{ijmk} - s_{ijk} - t_{ijm} + (w_{ij} + w_{ji}) v_{ikm} \leq \alpha(c_{km} w_{ij} + c_{mk} w_{ji}) \quad \forall\, k < m \tag{23}$$

$$u_{ijm} \in \mathcal{R} \quad \forall m \tag{24}$$

$$s_{ijm} \geq 0 \quad \forall m \tag{25}$$

$$t_{ijm} \geq 0 \quad \forall m \tag{26}$$

$$e_{ijkm} \geq 0 \quad \forall\, k < m \tag{27}$$

Onde este subproblema é definido como o dual do problema original com as variáveis binárias temporariamente fixadas. Sendo assim, este subproblema irá fornecer sempre um limite superior para a solução ótima.

Usando a função objetivo do SP BD para montar os cortes de Benders, tem-se o seguinte problema mestre de Benders (PM):

$$\min \sum_{k} [f_k z_{kk} + \xi_k] + \sum_{i} \sum_{k:k\neq i} (O_i + D_i) c_{ik} z_{ik} + \sum_{i} \eta \tag{28}$$

$$\text{s.t.: } (2) - (6), (12) - (13) \text{ e } (19) - (20) \tag{29}$$

$$\eta \geq \sum_{m} [(z_{jm} - z_{im}) u^h_{ijm} - z_{mm}(s^h_{ijm} + t^h_{ijm})] - \sum_{k} \sum_{m:m>k} y_{km} e^h_{ijkm} \quad \forall h \tag{30}$$

$$0 \geq \sum_{m} [(z_{jm} - z_{im}) u^g_{ijm} - z_{mm}(s^g_{ijm} + t^g_{ijm})] - \sum_{k} \sum_{m:m>k} y_{km} e^g_{ijkm} \quad \forall g \tag{31}$$

$$\eta \geq 0. \tag{32}$$

Onde as restrições (30) são conhecidas como cortes de otimalidade, e são geradas sempre que a solução ótima do SP BD é limitada; enquanto as restrições (31) são os cortes de viabilidade, e são geradas quando a solução do SP BD é ilimitada.

Conforme apresentado em Sá (2011) a versão clássica do método de decomposição não é muito eficiente para resolver o problema de localização de concentradores em árvore. Portanto, neste trabalho uma versão aprimorada de Benders baseada no algoritmo λ-ótimo proposto por Sá (2011) é implementada. A ideia básica deste algoritmo é gerar a cada iteração cortes adicionais o mais forte possível, conhecidos como cortes Pareto-ótimo (Magnanti & Wong, 1981), que não são dominados por nenhum outro corte. Para a geração destes novos cortes, o seguinte subproblema, proposto por Papadakos (2008), é utilizado.

$$\max \sum_{m} ((z^0_{jm} - z^0_{im}) u_{ijm} - z^0_{mm}(s_{ijm} + t_{ijm})) - \sum_{k} \sum_{\substack{m \\ m>k}} y^0_{km} e_{ijkm} \tag{33}$$

$$\text{s. a: } (22) - (27) \quad \forall\, k, m : k < m. \tag{34}$$

Onde, (z^0, y^0) é um ponto, conhecido como *core point*, que pertence ao interior relativo da casca convexa do poliedro formado pelas restrições do PM. Levando em conta que a cada iteração este poliedro muda, uma vez que, novas restrições são adicionadas ao problema. Então, a seguinte estratégia, também proposto por Papadakos (2008), é utilizada para atualizar o *core point* $(z^0, y^0)^h$ na iteração h.

$$(z^0, y^0)^{h+1} = \lambda(z^0, y^0)^h + (1-\lambda)(z, y)^h \tag{35}$$

Caso $(z, y)^h$, a solução corrente do problema, seja uma solução primal viável, então qualquer valor para λ $(0 < \lambda < 1)$ pode ser utilizado. Caso contrário o valor de λ é calculado usando a estratégia λ-ótimo proposta por Sá (2011) que consiste em resolver um subproblema adicional para encontrar um valor para λ $(0 < \lambda < 1)$ tal que a atualização do *core point* usando a equação (35) resulta em um novo *core point* viável.

Sendo assim, um algoritmo híbrido que combina as técnicas de aproximação externa e método de decomposição de Benders com adição de cortes Pareto-ótimo e atualização de *core point* via λ-ótimo, é apresentado no Algoritmo 2.

Algoritmo 2: Algoritmo híbrido para resolução do PLCAC

1 Faça $LS = +\infty$, $LI = -\infty$
2 Se $LS - LI < \epsilon$, então pare. Fim da execução, a solução ótima foi obtida.
3 Resolva o subproblema (33)-(34) para cada par $i - j$:
adicione ao PM um corte Pareto-ótimo (30)
4 Resolva o PM, obtendo o valor ótimo das variáveis z_{ik} e y_{km}
5 Atualize z_{ik} e f_{ikm} no SP OA (14)-(16)
6 Adicione os corte OA (19) ao PM OA
7 Atualize z_{ik} e y_{km} no SP (21)-(27)
8 Resolva o SP (21)-(27) para cada par $i - j$:
9 Se o SP for ilimitado, adicione ao PM um corte de viabilidade de Benders usando (31)
Caso contrário, adicione ao PM um corte de otimalidade usando (30)
10 Se os SP for limitado, então faça: $LS = \min\{LS\ , SP^* + \sum_k f_k z_{kk} + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik}\}$. Caso contrário, resolva o problema λ-ótimo
11 Atualize os *core points* usando (35) e volte ao passo 2.

4 Testes computacionais

Os testes foram feitos usando um conjunto de instâncias padrão da literatura: conjunto AP do serviço postal australiano introduzido por Ernst & Krishnamoorthy (1996). Cada instância AP é denotada por AP$n.\alpha$ onde n é o número de pontos de demanda e α 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. Foram feitos dois tipos de testes: um primeiro conjunto de teste que tem como finalidade analisar a topologia da rede considerando diferentes níveis de congestionamento e um segundo conjunto que tem objetiva a avaliação da eficiência dos algoritmos propostos.

Considerando a dependência do congestionamento no sistema com relação a capacidade do mesmo, deve-se levar em conta que os efeitos do congestionamento começa deteriorizar o nível de serviço quando o fluxo atinge um limiar de $m\%$ da capacidade do sistema. Neste trabalho é considerado o limiar de 70% da capacidade. Sendo assim, baseado na função *power law* o custo de congestionamento em um dado concentrador k é definido por: $\tau(g_k) = \max\{0, a(g_k - 0.7\Gamma_k)^b\}$.

Em todos os testes o valor de b é fixado em 2, logo a *power law* é uma função quadrática, enquanto o valor de a é variado para abordar diferentes níveis de custo de congestionamento. Onde, três níveis são abordados: sem congestionamento ($a = 0.0$), e dois níveis de congestionamento ($a = 0.01$ e $a = 0.1$).

A Figura 2 apresenta a topologia da rede ao variar o nível dos custos de congestionamento. De acordo com a figura, pode-se comprovar a importância de se abordar os efeitos do congestionamento durante o projeto da rede, uma vez que, estes custos influenciam fortemente o desenho da rede. De acordo com as configurações apresentadas na figura, existe uma tendência de se aumentar o número de concentradores instalados conforme o custo do congestionamento aumenta reduzindo a sobrecarga dos mesmos.

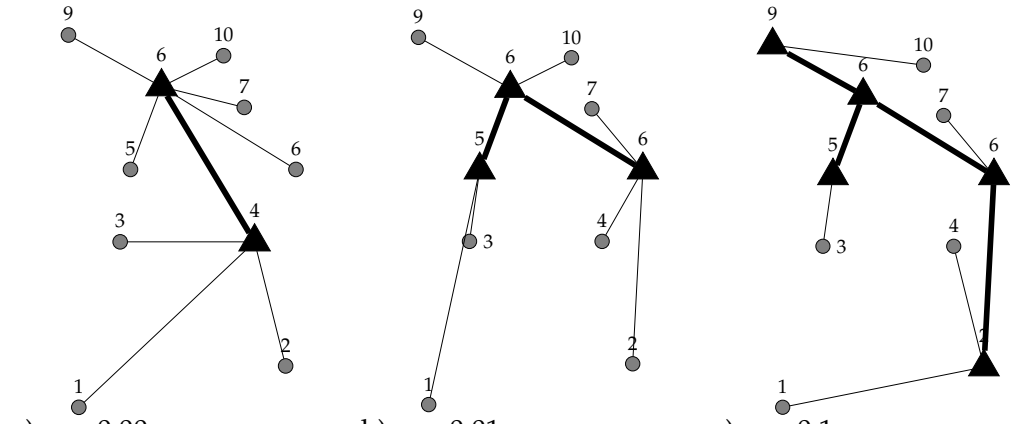

Figura 2: Configurações da rede para diferentes níveis de congestionamento

Com o objetivo de avaliar o desempenho dos algoritmos propostos, Algoritmo 1 (OA) e Algoritmo 2 (OA+BD), foram feitos um conjunto testes comparando ambos os algoritmos com o aplicativo comercial CPLEX. Todos os testes foram executados em um computador 1260 Xeon Westmere 2,66 GHz e 24 GB de memoria usando o sistema operacional Linux. Todos os problemas apresentados foram implementado em C++ usando a biblioteca Concert Technology CPLEX 12.5. Os testes foram feitos adotando como critério de parada o tempo limite de 14400 segundos, ou seja, 4 horas de processamento.

A Tabela 1 apresenta os resultados dos testes computacionais para instâncias AP10 e AP20. Onde as colunas: n apresenta o número de pontos de demanda; α, a economia de escala; a, o fator de congestionamento; p, o número de concentradores que foram instalados; $Cong.custo$, a participação do custo de congestionamento no custo total; e $\#Iter$, o número de iterações dos algoritmos propostos. De acordo com a tabela, pode-se observar que conforme o nível do custo de congestionamento aumenta o número de concentradores

instalados também aumentam. Além disso, pode-se perceber que a participação do custo de congestionamento no custo total (*Cong. custo*) é influenciado tanto pelo nível do custo de congestionamento (parâmetro a) quanto pelo fator de economia de escala aplicado (α). Com relação ao desempenho computacional dos algoritmos, de acordo com a tabela, os algoritmos propostos são capazes de resolver um maior número de instâncias propostos que o aplicativo CPLEX, isto é, enquanto o algoritmo híbrido OA+BD é capaz de resolver todas as instâncias propostas, o algoritmo OA é capaz de resolver vinte instâncias dentre o total de 24 instâncias propostas dentro do tempo limite apresentando um gap de otimalidade médio de 45% para instâncias não resolvidas na otimalidade. Já, o aplicativo CPLEX resolve apenas dezesseis instâncias não sendo capaz de fornecer nenhum gap. Com relação ao tempo computacional, com exceção de apenas uma instâncias de 10 nodos, o algoritmo OA+BD resolve todas as outras instâncias em menor tempo computacional.

Tabela 1: Resultados computacionais usando instâncias AP10 e AP20.

n	α	a	p	*Cong. custo*	CPLEX	OA			OA+BD	
					Tempo[s]	Tempo[s]	OA GAP	#Iter	Tempo[s]	#Iter
10	2	0	2	0.00%	**0.08**	0.11	—	1	0.09	2
10	4	0	1	0.00%	0.09	0.08	—	1	**0.03**	1
10	6	0	1	0.00%	0.08	0.07	—	1	**0.03**	1
10	8	0	1	0.00%	0.08	0.09	—	1	**0.03**	1
10	2	0.01	3	11.87%	99.95	11.67	—	8	**2.03**	8
10	4	0.01	3	11.27%	84.85	31.9	—	13	**4.12**	10
10	6	0.01	3	10.74%	179.66	19.62	—	11	**5.23**	12
10	8	0.01	2	21.51%	109.71	17.4	—	11	**6.36**	13
10	2	0.1	5	19.81%	386.71	24.94	—	11	**10.2**	14
10	4	0.1	5	17.82%	419.01	107.41	—	18	**33.52**	20
10	6	0.1	5	16.64%	635.43	152.4	—	20	**81.88**	26
10	8	0.1	5	15.59%	433.59	115.27	—	17	**61.19**	23
20	2	0	2	0.00%	3.23	2.49	—	1	**1.24**	3
20	4	0	2	0.00%	9	9.78	—	1	**2.38**	5
20	6	0	1	0.00%	13.42	7.71	—	1	**1.46**	3
20	8	0	1	0.00%	6.82	6.4	—	1	**0.79**	2
20	2	0.01	3	16.58%	14400	2516.74	—	12	**281.8**	14
20	4	0.01	3	15.77%	14400	3151.87	—	12	**416.89**	14
20	6	0.01	3	14.86%	14400	7369.74	—	15	**586.35**	16
20	8	0.01	2	23.39%	14400	7523.7	—	17	**743.08**	15
20	2	0.1	4	8.27%	14400	14400	46.43%	11	**3029.55**	25
20	4	0.1	4	10.71%	14400	14400	9.93%	16	**2613.55**	23
20	6	0.1	4	10.11%	14400	14400	70.64%	12	**4010.68**	24
20	8	0.1	4	9.57%	14400	14400	52.46%	17	**8868.91**	25

[a] Tempo limite de excedido.

Para avaliar melhor os algoritmos propostos, a Figura 3 apresenta uma gráfico em escala relativa apresenta a razão entre o tempo computacional gastos pelo aplicativo CPLEX e pelo algoritmo OA para resolver o problema, sob o tempo gasto pelo algoritmo OA+BD. Logo ele apresenta quão rápido o algoritmo OA+BD é se comparado com os outros algoritmos. De acordo com a figura o desempenho do algoritmo OA+BD chega a ser até 50 vezes mais rápido do que o CPLEX e até 10 vezes mais rápido que o algoritmo OA. Além disso, de acordo com o gráfico pode-se concluir que o desempenho do algoritmo OA é superior ao do *solver* CPLEX para a maioria das instâncias testadas.

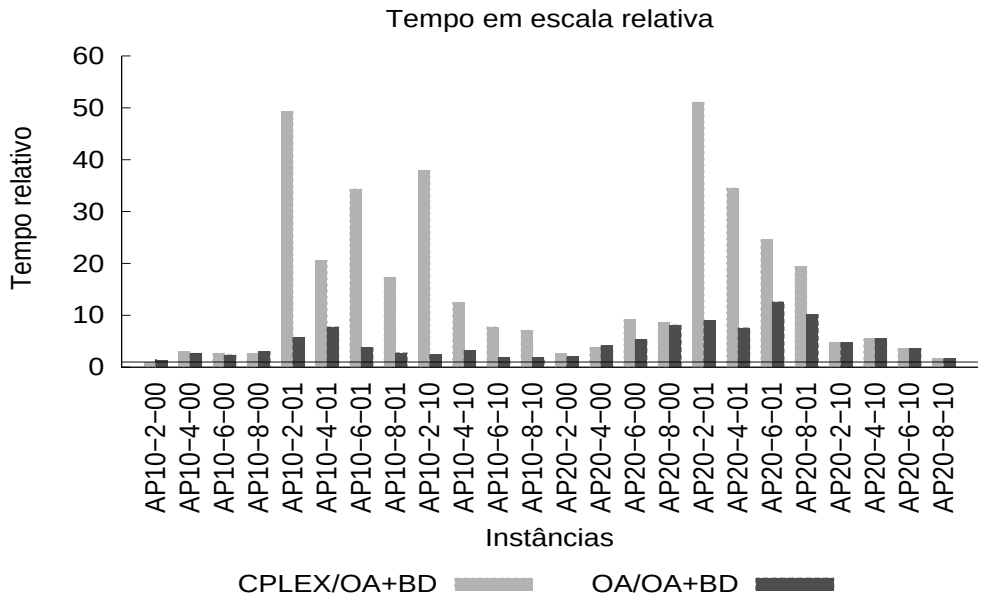

Figura 3: Gráfico de comparação do desempenho computacional do algoritmo OA+BD frente ao aplicativo CPLEX e algoritmo OA em escala relativa.

5 Conclusão

Com o intuito de melhor representar os custos no projeto de rede, este trabalho propõe o problema de localização de concentradores em árvore sob congestionamento. Para modelar o problema uma formulação de programação não-linear inteira mista é proposta. Para resolver o problema não-linear são propostos dois algoritmos baseados em métodos de decomposição, aproximação externa (OA) e aproximação externa com decomposição de Benders (OA+BD). Os algoritmos propostos são comparados com o aplicativo comercial CPLEX, comprovando a superioridade dos dois algoritmos para resolver as instâncias testadas. Ao comparar os algoritmos OA e OA+BD, o resultado dos testes computacionais mostram que o segundo apresenta melhor desempenho resolvendo todas as instâncias testadas em menor tempo.

Referências

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerisch Mathematik*, *4*, 238–252.

Camargo, R. S., Miranda Jr, G., Ferreira, R., & Luna, H. P. (2009). Multiple allocation hub-

and-spoke network design under hub congestion. *Computers and Operations Research*, *36*, 3097–3106.

Camargo, R. S. de, Miranda Jr., G. de, & Ferreira, R. P. (2011). A hybrid outer-approximation/benders decomposition algorithm for the single allocation hub location problem under congestion. *Operations Research Letters*, *39*(5), 329 - 337.

Contreras, I., Fernández, E., & Marín, A. (2009). Tight bounds from a path based formulation for the tree of hub location problem. *Comput. Oper. Res.*, *36*(12), 3117–3127.

Contreras, I., Fernández, E., & Marín, A. (2010). The tree of hubs location problem. *European Journal of Operational Research*, *202*, 390–400.

Duran, M., & Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed integer nonlinear programms. *Mathematical Programming*, *36*, 307–339.

Elhedhli, S., & Hu, F. X. (2005). Hub-and-spoke network design with congestion. *Computers & Operations Research*. (To appear)

Ernst, A. T., Hamacher, H., Jiangc, H., Krishnamoorthy, M., & Woegingerd, G. (2009). Uncapacitated single and multiple allocation p-hub center problems. *Computers and Operations Research*, *36*(7), 2230–2241.

Ernst, A. T., & Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, *4*, 139–154.

Fletcher, R., & Leyffer, S. (1996). Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, *66*, 327–349.

Gelareh, S. (2008). *Hub location models in public transport planning*. Unpublished doctoral dissertation, University of Saarlandes, Germany.

Gillen, D., & Levinson, D. M. (1999). Full cost of air travel in the california corridor. *Presented in the 78th Annual meeting of Transportation Research Board*, 10–14.

Magnanti, T. L., & Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, *29*(3), 464–483.

O'Kelly, M. E. (1986). The location of interacting hub facilities. *Transportation Science*, *20*, 92–106.

O'Kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, *32*, 393–404.

Papadakos, N. (2008). Practical enhancements to the magnanti–wong method. *Operations Research Letters*, *36*, 444–449.

Sá, E. M. d. (2011). *Localização de concentradores aplicada ao transporte público*. Unpublished master's thesis, Universidade Federal de Minas Gerais.