

O PROBLEMA DE LOCALIZAÇÃO DE CONCENTRADORES EM ÁRVORES: UM PROCEDIMENTO PARA ACELERAR UM ALGORITMO BASEADO EM DECOMPOSIÇÃO DE BENDERS

João Carlos Abreu Júnior¹ Thiago F. Noronha¹ Andréa Cynthia Santos²

¹Universidade Federal de Minas Gerais

Avenida Antônio Carlos 6627, CEP 31270-901, Belo Horizonte, MG, Brazil

joao.junior@dcc.ufmg.br, tfn@dcc.ufmg.br

²ICD-LOSI, UMR CNRS 6281, Université de Technologie de Troyes

12 rue Marie Curie, CS 42060, 10004 Troyes Cedex, France

{andrea.duhamel}@utt.fr

RESUMO

Seja $G = (N, E)$ um grafo completo direcionado, onde N é o conjunto de nós e E é o conjunto de arcos. Além disto, $W_{ij} \in \mathbb{R}$ é a demanda de fluxo do nó $i \in N$ para o nó $j \in N$, c_{ij} é o custo de trafegar cada unidade de fluxo em um arco $(i, j) \in E$ e p um número inteiro positivo. O problema de localização de concentradores em árvore (THLP, do inglês *Tree of Hubs Location Problem*) consiste em selecionar um subconjunto $P \subset N$ com p nós, denominados concentradores (do inglês *Hubs*), e conectá-los na forma de uma árvore. Em seguida, cada um dos nós em $N \setminus P$ é alocado a um único nó em P de modo que exista um único caminho entre cada par de nós i, j em G e cujo custo total de rotear as demandas W_{ij} é minimizado. O custo de trafegar uma unidade de fluxo entre nós concentradores sofre um fator de desconto α que é dado por $c_{ij} \times \alpha$. O estado da arte de algoritmos para este problema é capaz de resolver instâncias de até 100 nós usando um algoritmo baseado em decomposição de Benders, em um elevado tempo computacional. Isto se deve especialmente ao fato de que $O(|N|^2)$ subproblemas devem ser resolvidos a cada iteração do algoritmo, utilizando um algoritmo de programação linear. Neste trabalho, propomos um algoritmo *ad-hoc* que resolve cada subproblema em $O(|N|^2)$. Desta forma, acelera-se o tempo de execução do algoritmo de decomposição de Benders para o THLP. Resultados preliminares indicam que o tempo de resolução dos subproblemas pode ser acelerado em até 28%.

PALAVRAS CHAVE. Localização de Concentradores em Árvore, Localização de Concentradores, Decomposição de Benders

Área Principal: Otimização Combinatória, Logística e Transporte.

ABSTRACT

Given a directed complete graph $G = (N, E)$, where N is the set of nodes and E is the set of arcs. Let $W_{ij} \in \mathbb{R}$ be the flow demand of node $i \in N$ to node $j \in N$, c_{ij} be the cost of routing every flow unit in an arc $(i, j) \in E$ and p be a positive integer. The Tree of Hubs Location Problem consists in selecting a set of $P \subset N$ with p nodes, called of Hubs, and connect them by means of a tree. In addition, each node $N \setminus P$ is set to a single node belonging to P in such a way that there is a single path among any pair of nodes $i, j \in G$ and the total cost to route all the demands W_{ij} is minimized. The routing cost for every flow unit passing through Hubs receives a discount α given by $c_{ij} \times \alpha$. The state-of-the-art of algorithms for this problem is able to solve instances with up to 100 nodes in a high computational time, using a Benders decomposition. This is due to the fact that $O(|N|^2)$ subproblems have to be solved at each iteration. In this study, an

ad-hoc algorithm to solve $O(|N|^2)$ subproblems is proposed. Thus, the computational time of the Benders decomposition algorithm for THLP is speed up. Preliminary results indicate that about 28% of running time is saved.

KEYWORDS. Tree of Hubs Location, Hub Location, Benders decomposition

Main Area: Combinatorial Optimization, Logistics and Transport.

1. Introdução

A classe de problemas de localização de concentradores é estudado desde 1986, Alumur e Kara (2008), Campbell e O’Kelly (2012), Hekmatfar e Pishvaei (2009), e consiste em minimizar o custo de rotear demandas entre diversas localidades, comumente chamadas de concentradores. Esses nós se distinguem dos nós clientes por terem uma infraestrutura para tratar as demandas. Diferentes problemas de localização de concentradores são encontrados na literatura e o nível de dificuldade de resolução depende, entre outros, da maneira como esses concentradores estão conectados, se existem custos associados aos nós, se o roteamento das demandas deve circular por algum concentrador ou não e se um cliente se conecta a um ou mais concentradores (resp. alocação simples ou múltipla). Muitos desses problemas pertencem à classe de problemas *NP-difíceis* e encontram aplicações em redes de telecomunicações, logística, transporte e instalação de infraestruturas tais como trens de alta velocidade e aeroportos.

Neste trabalho, o problema de localização de concentradores em uma estrutura global de árvore geradora é investigado. Considera-se um grafo $G = (N, E)$, onde N é o conjunto de nós e E o conjunto de arestas. Uma árvore $\mathcal{T} = (N', E')$ de G é um grafo conexo e sem ciclos, onde $N' \subseteq N$ e $E' \subseteq E$, Cormen *et al.* (2001). Se $N' = N$, a árvore é chamada de árvore geradora. No problema de localização de concentradores (THLP, do inglês *Tree of Hub Location Problem*), demandas $W_{ij} \in \mathbb{R}$ são associadas a cada par de nós $i, j \in N$ e custos $c_{ij} \in \mathbb{R}_+$ são associados a cada aresta $(i, j) \in E$. Além disto, seja $3 \leq p \leq |N| - 1$ a quantidade de concentradores a serem instalados. Um desconto $0 \leq \alpha \leq 1$ é dado aos custos associados às arestas que conectam concentradores. O problema de localização de concentradores em árvores consiste em selecionar p nós concentradores conectados através de uma árvore. O custo das arestas que conectam os nós concentradores recebem um desconto α , i.e. $c_{ij} \times \alpha$ para todo par i e j selecionado como concentradores. No segundo nível de decisão do problema, deseja-se conectar os outros $|N| - p$ nós a um único concentrador de modo que o custo total de rotear todas as demandas W_{ij} seja minimizado. A solução final corresponde a uma árvore geradora de G , onde os nós concentradores também formam uma árvore.

O THLP foi introduzido em Contreras *et al.* (2010). Nesse trabalho, os autores provam que o problema é NP-difícil e apresentam uma formulação linear inteira mista (MILP, do inglês *Mixed Integer Linear Programming formulation*) contendo $O(|N|^2)$ variáveis binárias, $O(|N|^3)$ variáveis contínuas e $O(|N|^3)$ restrições. A formulação resolve instâncias de até 25 nós. Em Contreras *et al.* (2009), os autores apresentam um MILP baseado em caminhos entre cada par de nós, contendo $O(|N|^2)$ variáveis binárias, $O(|N|^4)$ variáveis contínuas e $O(|N|^4)$ restrições. Um algoritmo de relaxação Lagrangiana e uma heurística para o THLP também são propostos. Os resultados computacionais mostram que os gaps obtidos pela relaxação Lagrangiana são de no máximo 10% para instâncias com até 100 nós. Em Sá *et al.* (2013), os autores propõem um excelente algoritmo de decomposição de Benders que resolve instâncias com até 100 nós baseado na formulação proposta por Contreras *et al.* (2009). Uma heurística baseada em algoritmos genéticos com chaves aleatórias é apresentado em Pessoa *et al.* (2012).

Neste trabalho, introduzimos um procedimento para melhorar a resolução do subproblema associado à decomposição de Benders proposto por Sá *et al.* (2013). O restante do trabalho está organizado da seguinte forma: na Seção 2, a formulação matemática proposta por Contreras *et al.* (2009) para o THLP é introduzida. Em seguida, na Seção 3, a decomposição de Benders proposta

em Sá *et al.* (2013) é descrita e o procedimento proposto neste trabalho é apresentado na Seção 4. Finalmente, experimentos computacionais e conclusões são fornecidos respectivamente nas Seções 5 e 6.

2. Formulação matemática para o THLP

Uma formulação linear inteira mista para o THLP foi proposta em Contreras *et al.* (2009). As variáveis z_{kk} indicam os nós selecionados para serem concentradores. Neste caso, $z_{kk} = 1$ e $z_{kk} = 0$ em caso contrário. Quando $i \neq k$, as variáveis z_{ik} determinam a conexão dos nós clientes aos nós concentradores. Neste caso, $z_{ik} = 1$ e $z_{ik} = 0$ em caso contrário. As variáveis y_{km} , ($k < m$), representam as arestas que conectam dois nós concentradores. Portanto, quando $y_{km} = 1$ existe uma aresta conectando os nós concentradores k e m , 0 caso contrário. As variáveis x_{ij}^{km} modelam a quantidade de fluxo partindo do nó i com destino ao nó j que atravessa a aresta (k, m) . As constantes O_i , D_i e W_{ij} representam respectivamente, o fluxo total que parte do nó i , a demanda total de fluxo que chega ao nó i e a demanda de fluxo do nó j partindo do nó i . O fator de desconto $0 \leq \alpha \leq 1$ é dado aos custos associados às arestas que conectam dois nós concentradores e p é o número de nós concentradores que deverá ser selecionado, onde $3 \leq p \leq |N| - 1$.

A formulação é apresentada de (1)-(14), cuja função objetivo (1) minimiza o custo total do transporte do fluxo entre cada par de nós, satisfazendo todas as demandas. A primeira parte dessa função representa o custo de transporte do fluxo dos nós clientes até os nós concentradores e eventualmente de um nó concentrador até o destino. A segunda parte representa o custo de transporte através dos nós concentradores. Na segunda parte, o custo de cada arco possui um desconto α . As restrições (2) garantem que um cliente i poderá ser alocado apenas a nós concentradores k . As restrições (3) estabelecem que cada cliente i será alocado a apenas um nó k . As restrições (2) e (3) juntas determinam então que cada nó cliente i será alocado a apenas um nó concentrador k . As restrições (4) e (5) estabelecem que a aresta (k, m) poderá ser utilizada somente se k e m são nós concentradores. A restrição (6) assegura que p nós concentradores são selecionados. A restrição (7) garante que os nós concentradores formam uma árvore. As restrições (8) representam a conservação de fluxo entre cada par de nós i, j . As restrições (9) garantem que o fluxo de i para j são transportados através das arestas (k, m) . As restrições (7)-(9) estabelecem que os nós concentradores são conectados através de uma árvore. As restrições (10) e (11) estabelecem que o fluxo entre cada par de nós i e j circula apenas em nós concentradores. O domínio das variáveis x, y e z é dado nas restrições de (12) - (14).

$$\begin{aligned} \min \quad & \sum_{i \in N} \sum_{\substack{k \in N \\ k \neq i}} (c_{ik} O_i + c_{ki} D_i) z_{ik} + \\ & \sum_{i \in N} \sum_{\substack{j \in N \\ j > i}} \sum_{k \in N} \sum_{\substack{m \in N \\ m \neq k}} (c_{km} W_{ij} + c_{mk} W_{ji}) \times \alpha \times x_{ij}^{km} \end{aligned} \quad (1)$$

sujeito a:

$$z_{ik} \leq z_{kk} \quad \forall i \neq k \quad i, k \in N \quad (2)$$

$$\sum_{k \in N} z_{ik} = 1 \quad \forall i \in N \quad (3)$$

$$y_{km} \leq z_{kk} \quad \forall k < m \quad k, m \in N \quad (4)$$

$$y_{km} \leq z_{mm} \quad \forall k < m \quad k, m \in N \quad (5)$$

$$\sum_{k \in N} z_{kk} = p \quad (6)$$

$$\sum_{k \in N} \sum_{\substack{m \in N \\ m > k}} y_{km} = \sum_{k \in N} z_{kk} - 1 \quad (7)$$

$$\sum_{\substack{k \in N \\ k \neq m}} x_{ij}^{km} + z_{im} = \sum_{\substack{r \in N \\ r \neq m}} x_{ij}^{mr} + z_{jm} \quad \forall i < j, m \quad i, j, m \in N \quad (8)$$

$$x_{ij}^{km} + x_{ij}^{mk} \leq y_{km} \quad \forall i < j, k < m \quad i, j, k, m \in N \quad (9)$$

$$\sum_{\substack{m \in N \\ m \neq k}} x_{ij}^{km} \leq z_{kk} \quad \forall i < j, k \quad i, j, k \in N \quad (10)$$

$$\sum_{\substack{m \in N \\ m \neq k}} x_{ij}^{mk} \leq z_{kk} \quad \forall i < j, k \quad i, j, k \in N \quad (11)$$

$$x_{ij}^{km} \geq 0 \quad \forall i < j, k \neq m \quad i, j, k, m \in N \quad (12)$$

$$y_{km} \in \{0, 1\} \quad \forall k < m \quad k, m \in N \quad (13)$$

$$z_{ik} \in \{0, 1\} \quad \forall i, k \in N \quad (14)$$

3. Algoritmo de decomposição de Benders para o THLP

A formulação apresentada na Seção 2 é utilizada na decomposição de Benders proposta em Sá *et al.* (2013). A decomposição de Benders consiste em manter as variáveis z e y no problema mestre e as variáveis x no subproblema. Assim o subproblema é formado pela função objetivo (15) e as restrições (8)-(12).

$$\sum_{i \in N} \sum_{\substack{j \in N \\ j > i}} \sum_{k \in N} \sum_{\substack{m \in N \\ m \neq k}} (c_{km} W_{ij} + c_{mk} W_{ji}) \times \alpha \times x_{ij}^{km} \quad (15)$$

Após associar as variáveis u_{ijm} , e_{ijkm} , s_{ijk} e t_{ijm} , respectivamente, as restrições (8)-(11), o dual do subproblema para cada par de nós $i, j \in N$ com $i < j$, será escrito como a função objetivo (16) e as restrições (17)-(22).

$$\max \quad \sum_{m \in N} (\bar{z}_{jm} - \bar{z}_{im}) u_{ijm} - \sum_{k \in N} \sum_{\substack{m \in N \\ k < m}} \bar{y}_{km} e_{ijkm} - \sum_{k \in N} \bar{z}_{kk} (s_{ijk} + t_{ijk}) \quad (16)$$

subject to:

$$u_{ijm} - u_{ijk} - e_{ijkm} - s_{ijk} - t_{ijm} \leq \alpha (c_{km} W_{ij} + c_{mk} W_{ji}) \quad \forall k < m \quad k, m \in N \quad (17)$$

$$u_{ijm} - u_{ijk} - e_{ijkm} - s_{ijk} - t_{ijm} \leq \alpha (c_{km} W_{ij} + c_{mk} W_{ji}) \quad \forall k > m \quad k, m \in N \quad (18)$$

$$u_{ijm} \in \mathbb{R} \quad \forall m \in N \quad (19)$$

$$e_{ijkm} \geq 0 \quad \forall k < m; k, m \in N \quad (20)$$

$$t_{ijm} \geq 0 \quad \forall m \in N \quad (21)$$

$$s_{ijm} \geq 0 \quad \forall m \in N \quad (22)$$

O problema mestre é então formado pela função objetivo (23) e as restrições (2)-(7), (13), (14) e (24)-(26). As restrições (24) e (25) são, respectivamente, os cortes de otimalidade e viabilidade adicionados a partir de uma solução do dual do subproblema. Nestas restrições, os conjuntos H e G são, respectivamente, o conjunto de cortes de otimalidade e de viabilidade. Um corte de otimalidade ocorre quando o subproblema possui solução ótima e o corte de viabilidade ocorre quando o subproblema é ilimitado.

$$\min \sum_{i \in N} \sum_{k \in N, k \neq i} (c_{ik}O_i + c_{ki}D_i)z_{ik} + \sum_{i \in N} \sum_{j \in N, j > i} \eta_{ij} \quad (23)$$

$$\eta_{ij} \geq \sum_{m \in N} (z_{jm} - z_{im})u_{ijm}^h - \sum_{k \in N} \sum_{m \in N, k < m} y_{km}e_{ijkm}^h - \sum_{k \in N} z_{kk}(s_{ijk}^h + t_{ijk}^h) \quad \forall i < j \quad i, j \in N \quad h \in H \quad (24)$$

$$\sum_{m \in N} (z_{jm} - z_{im})u_{ijm}^h - \sum_{k \in N} \sum_{m \in N, k < m} y_{km}e_{ijkm}^h - \sum_{k \in N} z_{kk}(s_{ijk}^h + t_{ijk}^h) \leq 0 \quad \forall i < j \quad i, j \in N \quad h \in G \quad (25)$$

$$\eta_{ij} \geq 0 \quad \forall i < j \quad i, j \in N \quad (26)$$

O algoritmo baseado na decomposição de Benders é dividido em duas fases: uma fase chamada de aquecimento (do inglês *warm-start*) é apresentada no algoritmo 1 e uma fase denominada inteira (do inglês *integer*) é apresentada no algoritmo 2. No algoritmo 1, fase aquecimento, a variável LB é o limite inferior do problema e a variável h é a quantidade de iterações. Na linha 2, as variáveis são inicializadas. Em seguida, uma solução inicial (z^0, y^0) é gerada na linha 3. O laço das linhas 4 - 21 é executado cinco vezes. Nas linhas 5, 6 e 7, os seguintes passos são realizados, respectivamente: o subproblema com a solução (z^0, y^0) é resolvido, os cortes de otimalidade do problema mestre são adicionados e o problema mestre é então resolvido. Em seguida, nas linhas 8 e 9 são atualizadas, respectivamente, a solução do problema mestre nas variáveis z e y , e o valor do limite inferior fornecido pelo problema mestre. Posteriormente, na linha 10, o subproblema com a solução atual do problema mestre é resolvido. As linhas 12 e 13 são executadas se o subproblema for ilimitado, caso contrário as linhas 16 e 17 serão executadas. A linha 12 adiciona os cortes de viabilidade no problema mestre e a linha 13 encontra o melhor valor de λ para atualizar o ponto (z^0, y^0) . Os cortes de otimalidade são adicionados no problema mestre (linha 16). Em seguida, o valor do parâmetro λ é fixado na linha 17. O ponto inicial (z^0, y^0) e o número de iterações são atualizados, respectivamente, nas linhas 19 e 20.

No algoritmo 2, fase inteira, as variáveis LB e UB representam, respectivamente, os valores dos limites inferiores e superiores do problema, a variável h contabiliza o número de iterações necessárias para resolver o problema na otimalidade e a variável λ é o multiplicador para atualizar o ponto inicial (z^0, y^0) . Na linha 2 deste algoritmo as variáveis são inicializadas. Na linha 3 uma solução inicial para as variáveis y e z é retornada pela fase aquecimento. O laço das linhas 4 - 15 é executado até que o limite superior seja igual ao limite inferior, caracterizando assim a solução ótima para o THLP. O subproblema considerando a solução z^0 e y^0 do problema mestre é obtido na linha 5. Os cortes de otimalidade são adicionados na linha 6. Em seguida, o problema mestre é resolvido e a solução encontrada é retornada, respectivamente, nas linhas 7 e 8. O LB é atualizado na linha 9. A linha 10 resolve o subproblema utilizando a solução atual do problema mestre. Na linha 11, os cortes de otimalidade no problema mestre são adicionados e o ponto inicial (z^0, y^0) é atualizado na linha 12. Finalmente, se necessário, o limite superior da solução é atualizado e o número de iterações do algoritmo é incrementado, respectivamente, nas linhas 13 e 14.

4. Algoritmo Eficiente para resolução do subproblema

O algoritmo proposto para resolver o subproblema da decomposição de Benders da Seção 3 é apresentado nesta seção, para o caso em que as variáveis z e y são inteiras. A idéia geral do algoritmo consiste em resolver o subproblema primal para encontrar seu valor ótimo e então resolver o dual do subproblema através de um sistema de restrições.

No THLP, se a solução do problema mestre encontra variáveis z e y inteiras, a solução forma uma árvore que conecta todos os nós do conjunto N . Portanto, existe um único caminho

Algoritmo 1: Fase Aquecimento do algoritmo de decomposição de Benders para o THLP

```
1 inicio
2    $LB \leftarrow 0, h \leftarrow 0$  ;
3   Encontrar um ponto Magnanti e Wong inicial  $(z^0, y^0)$  ;
4   enquanto  $h < 5$  faça
5     Resolver subproblemas com  $(z^0, y^0)$  ;
6     Adicionar os cortes (24) para o problema mestre ;
7     Resolver o problema mestre ;
8      $(z^h, y^h) \leftarrow$  Solução do problema mestre ;
9      $LB \leftarrow$  Valor da solução do problema mestre ;
10    Resolver os subproblemas com  $(z^h, y^h)$  ;
11    se Subproblema é ilimitado então
12      Adicionar os cortes (25) no problema mestre ;
13      Encontrar o melhor valor para  $\lambda$  ;
14    fim
15    senão
16      Adicionar os cortes (24) no problema mestre ;
17       $\lambda \leftarrow 0.5$  ;
18    fim
19     $(z^0, y^0) \leftarrow (1 - \lambda)(z^0, y^0) + \lambda(z^h, y^h)$  ;
20     $h \leftarrow h + 1$  ;
21  fim
22 fin
```

Algoritmo 2: Fase Inteira do algoritmo de decomposição de Benders para o THLP

```
1 inicio
2    $UB \leftarrow \infty, LB \leftarrow 0, h \leftarrow 0, \lambda \leftarrow 0.5$ 
3    $(z^0, y^0) \leftarrow$  solução retornada pela fase aquecimento
4   enquanto  $UB \neq LB$  faça
5     Resolver subproblemas com  $(z^0, y^0)$ 
6     Adicionar os cortes (24) no problema mestre
7     Resolver o problema mestre
8      $(z^h, y^h) \leftarrow$  Solução do problema mestre
9      $LB \leftarrow$  valor da solução do problema mestre
10    Resolver subproblemas com  $(z^h, y^h)$ 
11    Adicionar o corte (24) no problema mestre
12     $(z^0, y^0) \leftarrow (1 - \lambda)(z^0, y^0) + \lambda(z^h, y^h)$ 
13    Atualizar UB, se necessário
14     $h \leftarrow h + 1$ 
15  fim
16 fin
```

para cada par de nós $i, j \in N$. Quando esses caminhos são conhecidos, os valores das variáveis x podem ser obtidos por inspeção. Os caminhos entre cada par de nós $i, j \in N$ podem ser encontrados executando um algoritmo de busca em profundidade em grafo partindo de cada nó $i \in N$ e assim é possível obter o valor da função objetivo do subproblema primal.

Em Cormen *et al.* (2001) é mostrado que um sistema de restrições de diferenças pode ser resolvido por um algoritmo de caminho mais curto de fonte única, como por exemplo Dijkstra e Bellman-Ford. Um sistema de restrições de diferenças é formado por várias restrições, onde cada restrição é composta por exatamente duas variáveis: uma com o sinal positivo e a outra com sinal negativo. Se no dual do subproblema aparecer apenas as variáveis u , esse problema pode ser classificado como um sistema de restrição de diferenças e poderá ser resolvido por um algoritmo polinomial.

Se os valores das variáveis z e y são inteiros, as restrições (10) e (11) do primal do subproblema são redundantes e podem ser retiradas do modelo. Assim, as variáveis duais associadas s e t também podem ser retiradas do problema dual. O dual do subproblema passa a ser composto das variáveis u e e . Na função objetivo do dual do subproblema, só aparecem duas variáveis u , uma com sinal positivo e outra com sinal negativo. Então a função objetivo será modelada no sistema de restrições, onde o lado direito da restrição é o valor da função objetivo obtida pela solução por inspeção do primal do subproblema. As demais restrições do sistema de restrição são as restrições do problema dual, associadas às variáveis primais y quando essas variáveis são iguais a 1, sem considerar as variáveis e . Deste modo, após resolver o sistema de restrições e obter o valor das variáveis u , as variáveis e podem ser obtidas da seguinte forma: as variáveis e que aparecem na função objetivo do dual do subproblema serão iguais a 0, pois elas aparecem com sinal negativo e o problema é de maximização. As demais variáveis e são iguais a 0 ou a diferença entre as variáveis u e o lado direito das restrições do dual.

O algoritmo proposto neste trabalho consiste em resolver o subproblema primal com o algoritmo de busca em largura, montar o sistema de restrição e resolvê-lo com o algoritmo de Bellman-Ford para encontrar os valores das variáveis duais u . Em seguida, os valores das variáveis duais e são atribuídos.

5. Experimentos Computacionais

Os experimentos computacionais foram executados em uma máquina Intel Xeon E5405 com 2.00 GHz de clock e 16 GB de memória RAM, com o sistema operacional Linux. O algoritmo de Benders proposto em Sá *et al.* (2013), chamado aqui de Benders (V1), e o algoritmo proposto na Seção 4, identificado aqui por Benders (V2), foram codificados utilizando a linguagem de programação C++. Estes algoritmos interagem com o CPLEX na versão 12.5 para resolver os problemas de programação linear inteira mista. Um conjunto de instâncias conhecido como AP (do inglês *Australian Post*) é utilizado nos experimentos computacionais e é descrita abaixo.

As instâncias AP são uma abstração do serviço de correio da Austrália em algumas cidades e foram utilizadas primeiro em Ernst e Krishnamoorthy (1996). Essas instâncias são utilizadas na literatura para problemas similares ao problema de localização de concentradores e são definidas em um grafo completo e direcionado, onde existe um custo $c_{ij} \in \mathbb{R}_+$ para cada par de nós $i, j \in N$ e uma demanda W_{ij} do nó de origem i para o nó de destino j . Neste trabalho, utiliza-se as instâncias AP de 50 nós, variando a quantidade p de nós concentradores em 3, 5, 8 e o valor α de desconto dos custos nas arestas que conectam nós concentradores pertencem ao conjunto $\{0.2, 0.5\}$.

Avaliou-se no experimento, o impacto do algoritmo proposto para resolver os subproblemas da decomposição de Benders quando as variáveis z e y são inteiras. Os algoritmos Benders (V1) e Benders (V2) foram executados em instâncias AP de 50 nós, com um limite de tempo de execução igual a 86400 segundos (24 horas). Os resultados deste experimento são mostrados na Tabela 1. Apenas as informações da segunda fase de cada algoritmo são apresentadas nesta tabela pois a fase aquecimento é idêntica para os dois algoritmos. A quantidade p de concentradores e o fator de desconto α são dados, respectivamente, nas colunas 1 e 2. As colunas 3, 4, 5, 6 e 7 são

referentes ao algoritmo Benders (V1). A coluna 3 é o gap da solução retornada pelo algoritmo, calculado por $(UB - LB) / UB$, onde UB e LB são respectivamente o limite superior e o limite inferior da solução. A coluna 4 é o número de iterações que o algoritmo executou. As colunas 5 e 6 são os tempos totais gastos em segundos para resolver, respectivamente, o problema mestre e o subproblema. A coluna 7 apresenta a razão entre o tempo total para resolver o subproblema e o número de iterações. As colunas 8, 9, 10, 11 e 12 representam estas mesmas informações para o algoritmo Benders (V2). A coluna 13 indica a razão entre: (i) a diferença entre o tempo de resolução dos subproblemas do Benders (V1) e o tempo de resolução dos subproblemas do Benders (V2) e (ii) o tempo de resolução dos subproblemas em Benders (V1).

p	α	Benders (V1)					Benders (V2)					%
		gap	Iter.	MP(s)	SP(s)	SP/Iter.	gap	Iter.	MP(s)	SP(s)	SP/Iter.	
3	0.2	0.00	1	17.18	579.38	579.38	0.00	1	17.17	411.97	411.97	28.89
3	0.5	0.00	2	1511.00	1173.55	586.78	0.00	2	1625.34	850.64	425.32	27.52
3	0.8	0.00	2	6777.70	1173.45	586.72	0.00	2	7180.10	846.16	423.08	27.89
5	0.2	0.00	2	124.46	1173.33	586.66	0.00	2	118.73	847.93	423.96	27.73
5	0.5	0.00	2	18136.76	1187.93	593.97	0.00	2	18705.20	897.95	448.98	24.41
5	0.8	0.02	2	92870.56	1175.96	587.98	0.02	2	171774.20	844.74	422.37	28.16

Tabela 1: Comparação entre Benders (V1) e Benders (V2).

Os resultados computacionais indicam que para todas as instâncias de testes, o algoritmo Benders (V2) permitiu reduzir significativamente os tempos computacionais para resolver os subproblemas, como mostrado na última coluna da Tabela 1. Esses resultados preliminares são promissores e permitirão realizar pesquisas futuras para melhorar ainda mais a qualidade das soluções produzidas.

6. Conclusão

Neste trabalho, o problema de localização de concentradores em árvores (THLP) é investigado. O THLP consiste em encontrar p nós concentradores, alocar os $|N| - p$ nós clientes, satisfazendo todas as demandas de fluxo, de forma que o custo total de rotear os fluxos sejam minimizados. Foi proposto um algoritmo para acelerar a resolução dos subproblemas na decomposição de Benders. Os resultados computacionais mostraram que o algoritmo proposto reduziu o tempo de execução de resolução dos subproblemas em até 28%.

Os resultados preliminares obtidos são interessantes e abrem perspectivas para pesquisas futuras afim de melhorar a qualidade das soluções produzidas, tais como inserção de cortes, inclusão de outros procedimentos de aceleração, etc.

Referências

- Alumur, S. e Kara, B. Y.** (2008), Network hub location problems: The state of the art. *European Journal of Operational Research*, v. 190, n. 1, p. 1 – 21.
- Campbell, J. F. e O’Kelly, M. E.** (2012), Twenty-five years of hub location research. *Transportation Science*, v. 46, n. 2, p. 153–169.
- Contreras, I., Fernández, E. e Marín, A.** (2009), Tight bounds from a path based formulation for the tree of hub location problem. *Computers & Operations Research*, v. 36, n. 12, p. 3117 – 3127.
- Contreras, I., Fernández, E. e Marín, A.** (2010), The tree of hubs location problem. *European Journal of Operational Research*, v. 202, n. 2, p. 390 – 400.
- Cormen, T. H., Stein, C., Rivest, R. L. e Leiserson, C. E.** *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edição, 2001.

Ernst, A. T. e Krishnamoorthy, M. (1996), Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, v. 4, n. 3, p. 139 – 154.

Hekmatfar, M. e Pishvaei, M. Hub location problem. Zanjirani Farahani, R. e Hekmatfar, M. (Eds.), *Facility Location*, Contributions to Management Science, p. 243–270, 2009.

Pessoa, L., Santos, A. C. e Resende, M. A biased random-key genetic algorithm for the tree of hubs location problem. *Proceedings of the 13ème congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF)*, Angers, France, p. 458–459, 2012.

Sá, E. M. D., de Camargo, R. S. e Miranda, G. (2013), An improved benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research*, v. 226, n. 2, p. 185 – 202.