

Método de Decomposição de Benders Especializado Aplicado ao Problema de Localização em dois Níveis de Facilidades não Capacitadas

Paganini Barcellos de Oliveira

Estudante de Mestrado da Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 - Pampulha - Belo Horizonte - MG - CEP 31270-901
paganini@decea.ufop.br

Alexandre Xavier Martins

Universidade Federal de Ouro Preto (UFOP)
xmartins@decea.ufop.br

Gilberto de Miranda Junior

Universidade Federal de Minas Gerais (UFMG)
miranda@dep.ufmg.br

Ricardo Saraiva de Camargo

Universidade Federal de Minas Gerais (UFMG)
rcamargo@dep.ufmg.br

RESUMO

Este trabalho consiste na aplicação do método de Decomposição de Benders Especializado para a resolução do Problema de Localização em dois Níveis de Facilidades não Capacitadas. Trata-se de um problema da área de Otimização de Sistemas de Grande Porte que tem grande aplicabilidade nos mais variados modelos de sistemas logísticos existentes. O grande desafio de problemas dessa natureza é estabelecer quais facilidades serão instaladas de forma a garantir uma configuração viável de custo mínimo que implique na eficiência máxima no atendimento dos clientes em uma rede. O problema pode ser modelado de inúmeras formas, sendo que em todas elas a ideia é estabelecer o quantitativo de fluxo que sai de uma facilidade de primeiro nível, passa por outra de segundo nível e atende um cliente final. Para demonstrar a eficiência do método proposto serão feitas análises em relação ao tempo computacional de resolução e qualidade da solução.

PALAVRAS CHAVE. Problema de Localização em dois Níveis de Facilidades não Capacitadas, Método de Decomposição de Benders, Otimização de Sistemas de Grande Porte.
Área Principal: Otimização Combinatória; Programação Matemática.

ABSTRACT

This paper consists of applying the Specialized Benders Decomposition method for solving the Two Level Uncapacitated Facility Location Problem. This is a problem of the field of Optimization of Large Scale Systems that has wide applicability in several models of existing logistics systems. The great challenge of such problems is to establish what facilities will be installed to ensure a feasible minimum cost configuration that implies in the maximum efficiency to meet the customer's network. The problem itself can be modeled in many forms, and in all of them the idea is to establish the amount of flow that leaving a facility first level, pass by another second level and meets one end customer. To demonstrate the efficiency of the proposed method, analyzes in relation to the computational time of resolution and quality of the solution will be made.

KEYWORDS. Two Level Uncapacitated Facility Location Problem; Benders decomposition; Optimization of Large Scale Systems.
Main Area: Combinatorial Optimization; Mathematical Programing.

1. Introdução

A estrutura hierárquica de muitos sistemas de grande porte tem como característica a existência de dois diferentes níveis de facilidades¹ para o atendimento de um conjunto de clientes. As facilidades de primeiro nível podem ser vistas como geradoras de recursos, enquanto as de segundo nível funcionam como agentes intermediários de distribuição física desses recursos.

Para [Klose e Drexel \(2005\)](#) a escolha do local de instalação de facilidades em um sistema de distribuição tem caráter estratégico para muitos tipos de empresas. Nos casos em que a região geográfica que compreende a rede de clientes e facilidades é muito extensa (considerar todo um estado, um país, ou até mesmo um continente, por exemplo), o posicionamento das facilidades se torna um fator ainda mais determinante para o sucesso das empresas.

Alguns exemplos reais de sistemas hierárquicos em dois níveis de facilidades podem ser vistos em: serviços de atendimento a saúde, na localização estratégica dos centros de especialidade médicas (primeiro nível), hospitais regionais (segundo nível) e postos de saúde (clientes) de um estado; na produção e distribuição de produtos como veículos, alimentos e vestuário, onde é necessário escolher onde construir as fábricas ou usinas, que abastecerão os centros de distribuição e que posteriormente irão repassar os produtos aos comerciantes varejistas espalhados ao redor do continente; e serviços de transmissão de sinais de TV e internet, na instalação de centrais de geração do sinal, que enviam dados para as torres de recepção/transmissão que por sua vez os repassam para as antenas locais dos clientes de uma grande cidade.

Para todos os exemplos citados, o problema consiste em estabelecer quais facilidades estarão ativas dentre todos os locais candidatos (primeiro e segundo nível), de forma a atender toda a demanda dos clientes com o menor custo total possível (custo de instalação e fluxo de recursos). A literatura sobre problemas de localização de facilidades é muito extensa e apresenta diversas variações para o problema que vão desde o acréscimo de outros níveis de facilidades, até problemas com múltiplos produtos e com restrição de capacidade, bem como outras características presentes em situações reais, como mostrado no trabalho de [Klose e Drexel \(2005\)](#). Neste sentido, percebe-se que o desenvolvimento de pesquisas que envolvam problemas neste contexto se justifica.

O presente trabalho objetivará na aplicação de um método de Decomposição de Benders Especializado para a resolução do Problema de Localização em dois Níveis de Facilidades não Capacitadas. Primeiramente será feita uma breve revisão sobre o assunto, bem como serão apresentados os conceitos, as técnicas e a formulação utilizados ao longo deste trabalho. Posteriormente serão apresentados os experimentos computacionais e seus respectivos resultados. Por fim, serão apresentadas as conclusões sobre a metodologia proposta.

2. Problema de Localização em dois Níveis de Facilidades não Capacitadas

Em meados dos anos sessenta, os trabalhos desenvolvidos pelos pesquisadores [Balinski \(1964\)](#) e [Manne \(1964\)](#) citados por [Kaufman et al. \(1977\)](#), resultaram em um modelo básico para o problema de localização de facilidades dos quais não era considerada a restrição de capacidade de cada uma das facilidades, o chamado Problema de Localização e dois Níveis de Facilidades não Capacitadas (*Two Level Uncapacitated Facility Location Problem* - TLUFLP). As principais características do TLUFLP são determinar pontos de abertura de facilidades de primeiro e segundo nível que resultem na mínima distância/custo de atendimento dos clientes.

Segundo [Aardal et al. \(1996\)](#) pouco se sabe sobre as propriedades estruturais dos TLUFLP, exceto o fato de tratar de um problema do tipo NP-difícil e que existiam poucos trabalhos sobre o TLUFLP na literatura até o início da década de noventa. A Figura (1) ilustra um exemplo de configuração de um TLUFLP na qual existem três candidatos a facilidade de primeiro nível (círculos), seis candidatos de segundo nível (losangos) e oito clientes (quadrados) a serem atendidos. Cada cliente é atendido por um par de facilidades ativas (brancas), e os traços representam fluxo de

¹Uma definição para palavra facilidade, no contexto deste trabalho, seria um local físico que dispõe de uma quantidade suficiente de recursos (produtos ou serviços, por exemplo), para atendimento de todos os clientes.

recursos que sai da facilidade de primeiro nível, passa pela de segundo nível e chega ao cliente. Por se tratar de um problema não capacitado, não faz sentido que um cliente seja atendido por mais de uma facilidade de primeiro e segundo nível.

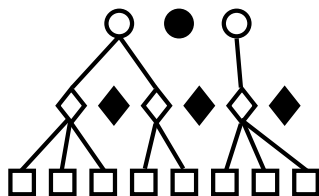


Figura 1: Representação esquemática de uma solução para o TLUFPL

Os principais trabalhos utilizando métodos exatos sobre o TLUFPL podem ser descritos a começar por Kaufman et al. (1977) que desenvolveu um algoritmo de *Branch-and-Bound* baseado em um algoritmo para o problema de um único nível (*Uncapacitated Facility Location Problem*) proposto por Efraymson e Ray (1966). Seguido pela elaboração de um método *Branch-and-Bound*, utilizando um procedimento dual com direção de subida (proposto por Erlenkotter (1978)) e decrescimento do primal proposta por Ro e Tcha (1984). Outro trabalho interessante foi apresentado por Barros e Labbé (1994), no qual eles sugeriram um método de relaxação lagrangeana combinado com uma heurística primal para gerar bons limites superior e inferior em um algoritmo de *Branch-and-Bound*. Em trabalhos mais recentes abordando o TLUFPL, destaque para os trabalhos de Gendron et al. (2013) e Baïou e Barahona (2014). O primeiro propôs a resolução do TLUFPL com uma restrição de atribuição simples por meio de um algoritmo Lagrangeano associado a um *Branch-and-Bound* para resolução de instâncias industriais reais e outras artificiais. Já o segundo identificou uma classe de problemas do TLUFPL que possui resolução em tempo polinomial, bem como apresentou algumas inequações e facetadas associadas ao politopo do problema.

Diferentes formulações foram propostas para o problema ao longo dos anos, dentre as quais destacam-se os modelos de Balinski (1964) e Manne (1964) e o de Narula e Ogbu (1979). A principal diferença entre os modelos citados são que o de Balinski (1964) e Manne (1964) utiliza apenas uma variável de fluxo de três índices enquanto o modelo de Narula e Ogbu (1979) é construído com duas variáveis de fluxo de dois índices. Para este trabalho optou-se pela utilização da formulação de Balinski (1964) e Manne (1964), que será apresentada em sequência.

Para descrever o modelo matemático do TLUFPL as seguintes definições foram adotadas: Consideremos K e J o conjunto de nós candidatos para se tornarem facilidades de nível 1 e nível 2, respectivamente, e I o conjunto de clientes. Consideremos também c_{ijk} serem os custos de atendimento de um cliente $i \in I$ por uma facilidade $k \in K$ e uma facilidade $j \in J$. Além disso, caso uma facilidade entre o conjunto de nós candidatos no nível 2 seja instalada, então haverá um custo de instalação f_k associado, o mesmo acontece para as facilidades de nível 1, com o parâmetro a_j . As variáveis de decisão do problema $z_k \in \{0, 1\}$ para todo $k \in K$ e $y_j \in \{0, 1\}$ para todo $j \in J$, indicam se entre todos os nós candidatos, as facilidades i e j estão instaladas, sendo também definidas ao longo das atribuições o valor das variáveis $x_{ijk} \geq 0$, quando as facilidades $k \in K$ e $j \in J$ são alocadas ao cliente $i \in I$. Neste contexto o problema pode ser modelado como segue:

$$\min \sum_{k \in K} f_k z_k + \sum_{j \in J} a_j y_j + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk} \quad (1)$$

$$\text{s.a.: } \sum_{j \in J} \sum_{k \in K} x_{ijk} = 1 \quad \forall i \in I \quad (2)$$

$$y_j - \sum_{k \in K} x_{ijk} \geq 0 \quad \forall i \in I, j \in J \quad (3)$$

$$z_k - \sum_{j \in J} x_{ijk} \geq 0 \quad \forall i \in I, k \in K \quad (4)$$

$$x_{ijk} \geq 0 \quad \forall i \in I, j \in J, k \in K \quad (5)$$

$$z_k \in \{0, 1\} \quad \forall k \in K \quad (6)$$

$$y_j \in \{0, 1\} \quad \forall j \in J \quad (7)$$

As restrições (2) garantem o atendimento de todos os clientes, enquanto que as restrições (3) e (4) eliminam a possibilidade de haver atribuição de fluxo oriundas de facilidades que não se encontram instaladas. Neste contexto a função objetivo (1) resulta na configuração de custo mínimo para as variáveis z_k , y_j e x_{ijk} que respeitam as restrições apresentadas.

3. Metodologia de resolução do TLUFLP

3.1. Método de Decomposição de Benders

O chamado método de Decomposição de Benders proposto por Benders (1962) foi criado para solucionar problemas de natureza inteira mista e não-linear. O algoritmo consiste na proposta de decomposição do problema original em dois outros problemas, um deles o problema mestre e o outro o subproblema. A escolha do método se deu principalmente pelas características básicas das variáveis e restrições presentes na formulação de Balinski (1964) e Manne (1964), que são: possuir duas variáveis complicantes z_k e y_j , para todo $k \in K$ e $j \in J$ respectivamente, que quando fixadas resultam em um subproblema que consiste em atribuir a cada $i \in I$ a configuração de menor custo entre as facilidades $j \in J$ e $k \in K$ ativas.

A aplicação do método de Decomposição de Benders no TLUFLP pode ser alcançada através da relaxação do Problema Mestre responsável pela fixação dos valores das variáveis z_k e y_j , referentes a decisão de instalar ou não uma facilidade, associado aos incrementos de cortes gerados por meio do Subproblema Dual, para cada um dos clientes $i \in I$. Primeiramente vamos mostrar o subproblema primal gerado a partir da fixação das variáveis y_j e z_k do problema original (parâmetros \bar{y}_j e \bar{z}_k do subproblema primal), como segue:

$$\min \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk} \quad (8)$$

$$\text{s.a.: } \sum_{j \in J} \sum_{k \in K} x_{ijk} = 1 \quad \forall i \in I \quad (9)$$

$$\sum_{k \in K} x_{ijk} \leq \bar{y}_j \quad \forall i \in I, j \in J \quad (10)$$

$$\sum_{j \in J} x_{ijk} \leq \bar{z}_k \quad \forall i \in I, k \in K \quad (11)$$

$$x_{ijk} \geq 0 \quad \forall i \in I, j \in J, k \in K \quad (12)$$

Dualizando o subproblema primal, temos:

$$\max \sum_{i \in I} v_i - \sum_{i \in I} \sum_{j \in J} \bar{y}_j w_{ij} - \sum_{i \in I} \sum_{k \in K} \bar{z}_k u_{ik} \quad (13)$$

$$\text{s.a.: } v_i - w_{ij} - u_{ik} \leq c_{ijk} \quad \forall i \in I, j \in J, k \in K \quad (14)$$

$$w_{ij} \geq 0 \quad \forall i \in I, j \in J \quad (15)$$

$$u_{ik} \geq 0 \quad \forall i \in I, k \in K \quad (16)$$

$$v_i \in \mathbb{R} \quad \forall i \in I \quad (17)$$

Assim, é possível montar o Problema Mestre (PM) de Benders, baseado no Subproblema Dual (SD) e na criação de uma variável η que a cada iteração do algoritmo irá ajudar a compor um novo corte para o problema.

$$\min \sum_{k \in K} f_k z_k + \sum_{j \in J} a_j y_j + \eta \quad (18)$$

$$\text{s.a.: } \eta \geq \sum_{i \in I} \bar{v}_i - \sum_{i \in I} \sum_{j \in J} y_j \bar{w}_{ij} - \sum_{i \in I} \sum_{k \in K} z_k \bar{u}_{ik} \quad \forall i \in I, j \in J, k \in K \quad (19)$$

$$\sum_{j \in J} y_j \geq 1 \quad (20)$$

$$\sum_{k \in K} z_k \geq 1 \quad (21)$$

$$\eta \geq 0 \quad (22)$$

$$z_k \in \{0, 1\} \quad \forall k \in K \quad (23)$$

$$y_j \in \{0, 1\} \quad \forall j \in J \quad (24)$$

As restrições (20) e (21) foram incorporadas ao PM para garantir que quando aplicado o algoritmo de Benders, haja sempre pelo menos uma facilidade instalada em cada nível. Essas duas restrições somente são necessárias no início da execução do algoritmo, quando o PM ainda não possui cortes de Benders (19).

A aplicação do método de Benders tradicional no TLUFLP, não se mostra uma técnica muito eficiente quando aplicada na íntegra, porém à medida que algumas técnicas de aceleração do método são acrescentadas, espera-se que tenha bons resultados em relação ao tempo computacional de resolução. Neste sentido, foram escolhidas algumas técnicas de aceleração do método de Benders para construção de um algoritmo especializado de Benders, que são: desenvolvimento de uma versão de múltiplos cortes de Benders; Inserção de cortes Pareto-Ótimo (Papadakos (2008)); resolução do SD sem a utilização de programação linear; utilização de fase de pré-aquecimento do método de Benders; e utilização de uma única árvore de *Branch-and-Bound*.

3.2. Múltiplos cortes de Benders para o TLUFLP

Segundo You e Grossmann (2011) pelo fato do algoritmo de Decomposição Benders tradicional retornar apenas um corte por iteração para o Problema Mestre, em problemas de grande porte, a sua convergência pode ser lenta, sendo que o algoritmo pode levar muitas iterações para chegar ao limite de tolerância predefinido. Para acelerar o algoritmo de Benders os trabalhos de Birge e Louveaux (1988) e You e Grossmann (2011) propõem a decomposição da variável η em cenários de forma a retornar múltiplos cortes (*Multicuts*) em cada iteração. Contudo, a adição de múltiplos cortes somente é possível em casos específicos que o modelo utilizado pode ser desagregado em relação a algum índice. No caso do TLUFLP o SD pode ser decomposto em i subproblemas (para cada cliente $i \in I$), conforme mostrado nas equações a seguir:

$$\begin{aligned} \max v_i - \sum_{j \in J} \bar{y}_j w_{ij} - \sum_{k \in K} \bar{z}_k u_{ik} \\ \text{s.a.: } v_i - w_{ij} - u_{ik} \leq c_{ijk} & \quad \forall j \in J, k \in K \\ w_{ij} \geq 0 & \quad \forall j \in J \\ u_{ik} \geq 0 & \quad \forall k \in K \\ v_i \in \mathbb{R} \end{aligned}$$

Assim é possível montar um PM utilizando o SD desagregado para cada i como segue:

$$\min \sum_{k \in K} f_k z_k + \sum_{j \in J} a_j y_j + \sum_{i \in I} \eta_i \quad (25)$$

$$\text{s.a.: } \eta_i \geq \bar{v}_i - \sum_{j \in J} y_j \bar{w}_{ij} - \sum_{k \in K} z_k \bar{u}_{ik} \quad \forall j \in J, k \in K \quad (26)$$

$$\sum_{j \in J} y_j \geq 1 \quad (27)$$

$$\sum_{k \in K} z_k \geq 1 \quad (28)$$

$$\eta_i \geq 0 \quad (29)$$

$$z_k \in \{0, 1\} \quad \forall k \in K \quad (30)$$

$$y_j \in \{0, 1\} \quad \forall j \in J \quad (31)$$

A variável η passa a ser desagregada para cada $i \in I$, e a função objetivo do PM (25) passa a incrementar o $\sum_{i \in I} \eta_i$.

3.3. Cortes Pareto-Ótimo

Em um trabalho desenvolvido por [Magnanti e Wong \(1981\)](#), eles perceberam que o sub-problema dual de Benders possui múltiplas soluções ótimas, o que possibilita gerar vários cortes diferentes. Eles então desenvolveram uma técnica que garante a geração de apenas um único corte que domina todos os outros, o chamado corte Pareto-Ótimo. [Papadakos \(2008\)](#) por sua vez, baseado nas idéias de [Magnanti e Wong \(1981\)](#), propôs um método mais simples de geração de cortes Pareto-Ótimo, o qual obteve sucesso.

A construção dos cortes Pareto-Ótimo de [Papadakos \(2008\)](#) para o TLUFPL, parte do princípio da montagem de um Subproblema Dual de Papadakos (SDPk), que tem a mesma estrutura do SD, com a diferença da substituição das variáveis $z_k (k \in K)$ e $y_j (j \in J)$ do PM pelas variáveis $z_k^0 (k \in K)$ e $y_j^0 (j \in J)$ respectivamente, calculadas por meio do uso dos *core points*². Assim podemos escrever o SDPk desagregado para cada cliente $i \in I$ da seguinte forma:

$$\max v_i - \sum_{j \in J} \bar{y}_j^0 w_{ij} - \sum_{k \in K} \bar{z}_k^0 u_{ik} \quad (32)$$

$$\text{s.a.: } v_i - w_{ij} - u_{ik} \leq c_{ijk} \quad \forall j \in J, k \in K \quad (33)$$

$$w_{ij} \geq 0 \quad \forall j \in J \quad (34)$$

$$u_{ik} \geq 0 \quad \forall k \in K \quad (35)$$

$$v_i \in \mathbb{R} \quad (36)$$

²Um ponto y_j^0 , para todo cliente i é um *core point* se ele pertence ao interior relativo da casca convexa, ou seja, se $y_j^0 \in \text{ir}(Y^c)$, onde $\text{ir}(Y^c)$ é o interior relativo da casca convexa Y^c de Y (conjunto de soluções viáveis).

Para atualização dos *core points* é necessária a criação de um parâmetro λ tal que $0 \leq \lambda \leq 1$, e além disso considerar $\lambda = \frac{1}{2}$ conforme sugerido por Papadakos (2008). A atualização de $z_k^0 (k \in K)$ e $y_j^0 (j \in J)$ em cada iteração de Benders t será feita por meio das equações que seguem:

$$z_k^{0(t)} = \lambda z_k^{0(t-1)} + (1 - \lambda) z_k^{(t)} \quad \forall k \in K \quad (37)$$

$$y_j^{0(t)} = \lambda y_j^{0(t-1)} + (1 - \lambda) y_j^{(t)} \quad \forall j \in J \quad (38)$$

Neste contexto podemos montar i cortes para cada um dos i SDPk como exposto em sequência.

$$\eta_i \geq \bar{v}_i - \sum_{j \in J} y_j^0 \bar{w}_{ij} - \sum_{k \in K} z_k^0 \bar{u}_{ik} \quad \forall j \in J, k \in K \quad (39)$$

3.4. Algoritmo para resolução do Subproblema Dual - ASD

Em toda iteração t do método de Decomposição de Benders padrão obtém-se dois vetores (y^t, z^t) referentes à solução do PM, que são parâmetros a serem usados no SD e guardam a informação sobre quais facilidades estão ativas e quais não.

A resolução do subproblema do TLUFLP desta forma pode ser vista como um problema de atribuição, no qual a solução ótima consiste em associar o par de facilidades $(j \in J, k \in K)$ que atendem cada cliente $i \in I$ com o menor custo. Assim resolver esse problema por inspeção pode resultar em economia de tempo computacional. Para apresentar o método resolução do ASD, algumas notações foram definidas:

$$O_y = \{j \in J : y_j^t = 1\} \text{ e } O_z = \{k \in K : z_k^t = 1\}$$

$$C_y = \{j \in J : y_j^t = 0\} \text{ e } C_z = \{k \in K : z_k^t = 0\}$$

As facilidades $k \in K$ e $j \in J$ abertas (*open*) em cada iteração t do algoritmo de Benders serão pertencerão aos conjuntos O_z e O_y , respectivamente. O mesmo valor para as facilidades $k \in K$ e $j \in J$ fechadas (*closed*), com C_z e C_y . Ao analisarmos a função objetivo do SD (13), nota-se que a função atingirá seu valor ótimo quando o somatório dos v_i for máximo e o restante da equação for zero. Assim, utilizando os conceitos de complementaridade de folga que são:

$$u_{ik}^t \left(z_k^t - \sum_{j \in J} x_{ijk}^t \right) = 0 \quad \forall i \in I, k \in K \quad (40)$$

$$w_{ij}^t \left(y_j^t - \sum_{k \in K} x_{ijk}^t \right) = 0 \quad \forall i \in I, j \in J \quad (41)$$

$$x_{ijk}^t (c_{ijk} - v_i^t + u_{ik}^t + w_{ij}^t) = 0 \quad \forall i \in I, j \in J, k \in K \quad (42)$$

E explorando de maneira mais ampla a restrição (14), podemos chegar as seguintes conclusões:

$$v_i^t = \min \{c_{ijk} : j \in O_y, k \in O_k\} \quad \forall i \in I \quad (43)$$

$$u_{ik}^t = 0 \quad \forall k \in O_z \quad (44)$$

$$w_{ij}^t = 0 \quad \forall j \in O_y \quad (45)$$

$$u_{ik}^t \geq \max_{j \in O_y} \{ \max \{0, v_i^t - c_{ijk}\} \} \quad \forall k \in C_z \quad (46)$$

$$w_{ij}^t \geq \max_{k \in O_z} \{ \max \{0, v_i^t - c_{ijk}\} \} \quad \forall j \in C_y \quad (47)$$

Através das equações de (43) à (47) é possível encontrar uma solução ótima para o SD de Benders sem a utilização do método SIMPLEX. As duas últimas, (46) e (47), são aquelas que demandaram um maior cuidado durante sua implementação. Neste contexto o Algoritmo (1) foi desenvolvido para resolução do SD.

Algoritmo 1: Resolução do ASD

```

início
  para todo ( $i \in I$ ) faça
     $v_i^t \leftarrow \min \{c_{ijk}\} \quad \forall j \in O_y, k \in O_k$ 
     $u_{ik}^t \leftarrow 0 \quad \forall k \in O_z$ 
     $w_{ij}^t \leftarrow 0 \quad \forall j \in O_y$ 
     $u_{ik}^t \leftarrow \max \{0, v_i^t - c_{ijk}\} \quad \forall j \in O_y, k \in C_z$ 
     $w_{ij}^t \leftarrow \max \{0, v_i^t - c_{ijk}\} \quad \forall k \in O_z, j \in C_y$ 
    para todo ( $k \in C_z \wedge j \in C_y$ ) faça
      se ( $(u_{ik}^t + w_{ij}^t) < (v_i^t - c_{ijk})$ ) então
         $df \leftarrow (v_i^t - c_{ijk}) - (u_{ik}^t + w_{ij}^t)$ 
        selecione  $\{1 \vee 2 \vee 3 \vee 4 \vee 5\}$  (// Válido para todo i) faça
          1.  $u_{ik}^t \leftarrow u_{ik}^t + df/2$ 
              $w_{ij}^t \leftarrow w_{ij}^t + df/2$ 
          2.  $u_{ik}^t \leftarrow u_{ik}^t + df$ 
          3.  $w_{ij}^t \leftarrow w_{ij}^t + df$ 
          4. se ( $w_{ij}^t > u_{ik}^t$ ) então  $u_{ik}^t \leftarrow u_{ik}^t + df$ 
             senão  $w_{ij}^t \leftarrow w_{ij}^t + df$ 
          5. se ( $w_{ij}^t < u_{ik}^t$ ) então  $u_{ik}^t \leftarrow u_{ik}^t + df$ 
             senão  $w_{ij}^t \leftarrow w_{ij}^t + df$ 
        fim
      fim
    fim
  fim

```

Pelo Algoritmo (1) é possível notar que o cálculo das variáveis v_i^t é feito de forma automática. As variáveis u_{ik}^t e w_{ij}^t são calculadas em função da configuração atual das facilidades $k \in K$ e $j \in J$, respectivamente, que podem estar abertas ou fechadas, como mostrado nas equações de (44) à (47). Nos casos em que temos um par de facilidades (primeiro e segundo nível) do tipo C_z e C_y , pode ser preciso fazer a integralização das variáveis w_{ij}^t e u_{ik}^t . São propostas cinco maneiras simples de integralização das variáveis, de forma a manter o conceito de complementaridade de folga. Todas elas utilizam a ideia de que se $(v_i^t - c_{ijk}) - (u_{ik}^t + w_{ij}^t) = df > 0$, devemos fazer com que essa diferença df seja atribuída a uma das duas variáveis, ou distribuída entre elas.

As cinco formas testadas podem ser resumidas em: Distribuir $\frac{df}{2}$ para w_{ij}^t e u_{ik}^t ; Somar df em w_{ij}^t ; Somar df em u_{ik}^t ; Somar df em w_{ij}^t caso ($w_{ij}^t > u_{ik}^t$), caso contrario somar em u_{ik}^t ; Somar df em u_{ik}^t caso ($w_{ij}^t > u_{ik}^t$), caso contrario somar em w_{ij}^t .

3.5. Pré-aquecimento e uso de uma única árvore de *Branch-and-Bound*

Outras duas técnicas de aceleração do método de Benders aplicadas no TLUFLP foram o Pré-aquecimento e uso de uma única árvore de *Branch-and-Bound*. O Pré-aquecimento (*Hot Start*) visa a obtenção de uma solução inicial de forma rápida e eficiente para acelerar a convergência dos métodos de resolução de problemas lineares. O uso de uma única árvore de *Branch-and-Bound* é feita por meio de funções *Callback* provenientes dos resolvidores comerciais, como o CPLEX³,

³Resolvidor comercial da empresa IBM (*International Business Machines*).

por exemplo, que permite cada nó da árvore de *Branch-and-Bound* seja explorado uma única vez.

A técnica de Pré-aquecimento do método de Benders aplicada ao TLUFLP foi feita através da relaxação linear das variáveis y_j e z_k que passam a ser \mathbb{R}^+ para todo $j \in J$ e $k \in K$, respectivamente, nas quatro primeiras iterações do método de Benders. O número de iterações foi definido de tal forma que a quantidade de cortes adicionados ao PM não prejudicasse o desempenho do método aplicado, um vez que os resolvidores comerciais são sensíveis à quantidade de cortes iniciais adicionados, isto é, o acréscimo de cortes (iniciais) em demasia melhora o limite inferior do PM mas também piora o tempo computacional ao longo da busca na árvore de *Branch-and-Bound*. Para o problema estudado, quatro se mostrou um bom número.

Neste trabalho optou-se pela utilização do CPLEX como resolvidor, e as funções *Callback* adotadas foram: a *CutCallback* e a *LazyConstraintCallback*. A ideia de executar a função *CutCallback* parte do princípio que seja feita a expansão sobre todos os nós da árvore de *Branch-and-Bound* no qual o valor das variáveis do PM são viáveis na região de viabilidade do TLUFLP, porém não inteira viáveis. Já a *LazyConstraintCallback* terá o mesmo papel, mas somente para as variáveis inteiras, e que desta forma são candidatos a serem uma solução incumbente.

4. Resultados Computacionais

Os experimentos computacionais realizados tinham por objetivo a avaliação da performance do algoritmo especializado de Benders propostos neste trabalho. Para a realização dos testes, um conjunto de instâncias foram geradas em formato semelhante ao proposto por Ro e Tcha (1984). Na construção das instâncias considerou-se o custo de transporte de uma facilidade de primeiro nível para outra de segundo nível igual a 0.0125 unidades monetária por milha, ao passo que o custo de transporte de uma facilidade de segundo nível para um cliente foi de 0.0250 unidades monetária por milha percorrida.

Em relação às distâncias entre todos os locais (facilidades e clientes), as mesmas foram geradas de maneira aleatória em um intervalo uniformemente distribuído que variava entre 100 e 5000 milhas. Os custos fixos das facilidades de segundo nível foram gerados aleatoriamente no intervalo uniformemente distribuído de 15000 às 20000 unidades monetárias, enquanto que os custos fixos das facilidades de primeiro nível variou de 50000 a 60000 unidades monetárias. A demanda de cada um dos clientes foi gerada também de forma randômica que variava de 50 a 2000 unidades, sendo esta incorporada a todos os parâmetros de custo c_{ijk} para um mesmo cliente $i \in I$. As dimensões das instâncias geradas foram semelhantes às 10 primeiras utilizadas no trabalho de Ro e Tcha (1984). Além disso outras 8 instâncias maiores foram geradas e testadas.

Todos os experimentos foram realizados em um notebook Asus, com um processador Intel Core I7-3610QM 2.3GHz, e 8G de RAM, em um sistema operacional Linux de 64bits. Os algoritmos foram implementados em Concert, utilizando o CPLEX 12.4 como resolvidor, tendo como critérios de parada um tempo limites de 10800 segundo, equivalente a três horas de processamento na máquina.

Em cada um dos testes foram analisados o limites inferior e superior, o *gap* de dualidade, a solução para o problema e o tempo computacional de resolução. Além disso, também foram implementados o Problema Original (PO) descrito pelas equações de (1) à (7) e o método de Benders tradicional (BT) utilizando apenas as equações de (18) à (24) do PM e as equações de (13) à (17) do SD, ambos para avaliação da performance do método de Benders Especializado (BE) desenvolvido. O BE reúne todas as técnicas de aceleração descritas ao longo deste trabalho, que são: realização quatro iterações de pré-aquecimento, inserção de múltiplos cortes tipo Pareto-Ótimo de Papadakis (39) no PM, utilização de uma única árvore de *Branch-and-Bound*, e resolução do subproblema dual através do Algoritmo (1).

As Tabelas (1) e (2) ilustram os resultados obtidos em todos os testes realizados. A Tabela (1) mostra o comportamento do BE frente a mudanças na forma de realizar o ajuste das variáveis w_{ij}^t e u_{ik}^t do Algoritmo (1). Já na Tabela (2) é possível observar os tempos de resolução dos seguintes algoritmos: PO, BT e BE.

Tabela 1: Tempos de resolução do TLULFP para as variações do cálculo de w_{ij}^t e u_{ik}^t pelo BE.

Instâncias	Mét. 1	Mét. 2	Mét. 3	Mét. 4	Mét. 5
	t(s)	t(s)	t(s)	t(s)	t(s)
5-5-50	0.01	0.01	0.01	0.01	0.00
5-15-50	0.02	0.02	0.02	0.02	0.02
5-25-50	0.04	0.03	0.04	0.03	0.03
7-10-50	0.02	0.02	0.02	0.02	0.02
7-15-50	0.05	0.05	0.05	0.05	0.05
7-20-50	0.04	0.04	0.03	0.03	0.04
7-25-50	0.16	0.16	0.16	0.16	0.15
10-10-50	0.02	0.03	0.02	0.02	0.03
10-15-50	0.04	0.04	0.04	0.04	0.04
10-20-50	0.11	0.11	0.11	0.11	0.11
10-50-100	1.64	1.60	1.64	1.55	1.68
15-25-100	0.65	0.66	0.58	0.67	0.59
15-30-150	1.11	1.06	1.01	1.11	1.01
30-50-200	15.71	10.09	12.39	13.93	12.17
30-100-200	1190.66	547.91	664.20	797.42	779.37
30-100-300	936.46	446.00	347.57	224.73	622.22
40-100-300	8061.81	7347.27	*	4242.35	*
50-100-500	8772.49	*	3268.36	7062.12	5479.82

* Não resolvidos em três horas de processamento.

Tabela 2: Tempos de resolução das instâncias para os três algoritmos.

Instâncias	PO		BT		BE	
	t(s)	gap(%)	t(s)	gap(%)	t(s)	gap(%)
5-5-50	0.00	0.00	0.01	0.00	0.01	0.00
5-15-50	0.01	0.00	0.07	0.00	0.02	0.00
5-25-50	0.03	0.00	0.37	0.00	0.03	0.00
7-10-50	0.01	0.00	0.05	0.00	0.02	0.00
7-15-50	0.06	0.00	0.11	0.00	0.05	0.00
7-20-50	0.02	0.00	0.31	0.00	0.03	0.00
7-25-50	0.12	0.00	7.74	0.00	0.16	0.00
10-10-50	0.02	0.00	0.08	0.00	0.02	0.00
10-15-50	0.02	0.00	0.48	0.00	0.04	0.00
10-20-50	0.02	0.00	2.01	0.00	0.11	0.00
10-50-100	1.07	0.00	*	2.35	1.55	0.00
15-25-100	0.43	0.00	44.75	0.00	0.67	0.00
15-30-150	0.77	0.00	486.02	0.00	1.11	0.00
30-50-200	28.19	0.00	*	10.14	13.93	0.00
30-100-200	600.46	0.00	*	86.53	797.42	0.00
30-100-300	953.10	0.00	*	92.26	224.73	0.00
40-100-300	5667.11	0.00	*	95.59	4242.35	0.00
50-100-500	*	0.94	*	97.30	7062.12	0.00

* Não resolvidos em três horas de processamento.

Para montagem Tabela (2) foram utilizados os dados referentes a resolução das instâncias do TLUFLP pelo método 4 do Algoritmo (1). O *gap* de dualidade de cada um dos teste realizados foi obtido através da expressão: $gap(\%) = 100(\text{limite superior} - \text{limite inferior})/\text{limite superior}$.

5. Conclusão

A partir dos resultados obtidos pode-se concluir que o resolvidor do CPLEX se mostrou uma boa alternativa para a resolução do TLUFLP, porém apenas para instâncias que são relativamente pequenas. A Tabela (2) deixou evidente, que para problemas mais simples o uso do resolvidor comercial é sempre mais eficiente do que a implementação do BT e BE. Esse resultado mostra a qualidade dos resolvidores disponíveis no mercado que apesar de não serem códigos específicos para um determinado problema, conseguem resolver muitos problemas com certa facilidade.

Em relação a implementação do BT nota-se a fragilidade do método quando a dimensão dos problemas aumenta. Ao observar o comportamento do algoritmo frente às oito últimas instâncias testadas (Tabela (2)) percebe-se que o método não apresenta um desempenho satisfatório, com tempos computacionais muito altos ou até mesmo não resolvíveis dentro do limite de tempo.

O método BE mostrou-se competitivo, quando comparado com a implementação do Problema Original, e além disso, para instâncias maiores mostrou-se mais eficiente. Por se tratar de uma implementação específica para o TLUFLP, o método tende a ser melhor a medida que as dimensões do problema aumentam. Como exemplo a última instância mostrada na Tabela (1) somente foi resolvida pelo BE, enquanto as demais ultrapassaram o tempo de três horas de processamento.

Quando analisado a Tabela (1), nota-se que houve pouca variação entre métodos de integralização das variáveis w_{ij}^t e u_{ik}^t do SD para as treze primeiras instâncias testadas. Por outro lado, as cinco últimas indicaram que o método quatro tende a apresentar melhores resultados.

Por fim, pode-se dizer que, apesar de poucos testes, o BE se mostrou um método competitivo de resolução do TLUFLP. Como sugestão para trabalhos futuros fica o aperfeiçoamento do método BE por meio de outras técnicas de aceleração do método de Benders, como geração de outros tipos de cortes, a realização de testes em instâncias realísticas maiores ou criadas de diferentes formas, bem como um estudo mais detalhado sobre a forma de resolução do subproblema dual por inspeção. Assim espera-se que seja possível melhorar ainda mais a taxa de convergência do método BE.

Agradecimentos

Os autores agradecem às agências FAPEMIG, CNPq e CAPES, a Universidade Federal de Minas Gerais (UFMG), e a Universidade Federal de Ouro Preto (UFOP), pelo apoio financeiro dado ao desenvolvimento deste trabalho.

Referências

- Aardal, K., Labbé, M., Leung, J., e Queyranne, M. (1996). On the two-level uncapacitated facility location problem. *INFORMS J. COMPUT*, 8:289–301.
- Baïou, M. e Barahona, F. (2014). A polyhedral study of a two level facility location model. *RAIRO - Operations Research*, 48(2):153–165.
- Balinski, M. (1964). On finding integer solutions to linear programs. pages 225–248.
- Barros, A. e Labbé, M. (1994). A general model for uncapacitated facility and depot location problem. *Location Science*, 2(3):173–191.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerisch Mathematik*, 4:238–252.
- Birge, J. R. e Louveaux, F. V. (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384–392.



- Efroymsen, M. e Ray, T. (1966). A branch-and-bound algorithm for plant location. *Operational Research*, 14:361 – 368.
- Erlenkotter, D. (1978). A dual-based procedure for uncapacitated facility location. *Operational Research*, 26:992–1009.
- Gendron, B., Khuong, P.-V., e Semet, F. (2013). A Lagrangian-based branch-and-bound algorithm for the two-level uncapacitated facility location problem with single-assignment constraints. Technical report, CIRRELT.
- Kaufman, L., Eede, M. V., e Hansen, P. (1977). A plant and warehouse location problem. *Operational Research Quarterly*, 28(3):547 – 554.
- Klose, A. e Drexl, A. (2005). Facility location models for distribution system design. *European Journal of Operational Research*, 162(1):4 – 29.
- Magnanti, T. L. e Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464 – 483.
- Manne, A. (1964). Plant location under economics of scale. 11:213–235.
- Narula, S. C. e Ogbu, U. (1979). An hierarchal location–allocation problem. *Omega*, 7(2):137–143.
- Papadakos, N. (2008). Practical enhancements to the magnanti–wong method. *Operations Research Letters*, 36(4):444 – 449.
- Ro, H. B. e Tcha, D. W. (1984). A branch and bound algorithm for the two-level uncapacitated facility location problem with some side constraints. *European Journal of Operational Research*, 18(3):349 – 358.
- You, F. e Grossmann, I. E. (2011). Multicut benders decomposition algorithm for process supply chain planning under uncertainty.