

UMA NOVA FORMULAÇÃO E UM CONJUNTO DE DESIGUALDADES VÁLIDAS PARA O PROBLEMA DA COBERTURA MÁXIMA p -HUB COM ALOCAÇÃO SIMPLES E COBERTURA PARCIAL

Patrick Doglio

Pós-Graduação em Engenharia de Produção - UFF
R. Passo da Pátria, nº 156, Bloco D, sala 309, São Domingos, Niterói-RJ
patrickdoglio@id.uff.br

Marcos Roboredo, Artur Pessoa

Departamento de Engenharia de Produção - UFF
R. Passo da Pátria, nº 156, Bloco D, sala 306, São Domingos, Niterói-RJ
mroboredo@id.uff.br, artur@producao.uff.br

RESUMO

O problema da cobertura máxima p -hub com alocação simples e cobertura parcial considera uma rede hub-and-spoke com locais onde hubs podem ser instalados e diversos fluxos a serem transportados. O problema decide onde localizar p hubs e como alocar cada ponto a um único hub localizado, gerando um grau de cobertura para cada fluxo, em função do seu custo de atendimento. O problema visa maximizar a soma ponderada dos fluxos pelo seu grau de cobertura. Propomos neste trabalho uma nova formulação e um conjunto de desigualdades válidas para o problema. Propomos ainda que tais desigualdades sejam incluídas sob demanda durante o algoritmo de *branch-and-bound*. Nós apresentamos diversos experimentos que mostram que a formulação proposta supera a melhor formulação conhecida para todas as instâncias testadas, mesmo para o caso em que as desigualdades não são incluídas. Além disso, apresentamos a solução ótima de diversas instâncias em aberto.

PALAVRAS CHAVE. Programação Inteira, Localização de Hubs, Cobertura Máxima.

Tópicos: Otimização Combinatória, Logística e Transportes

ABSTRACT

The single allocation p -hub maximal covering problem considers a hub-and-spoke network with locations where hubs can be located and several flows to be transported. The problem decides where to locate p hubs and how to allocate each point to a single located hub, generating a coverage degree for each flow. The problem aims to maximize the sum of each flow weighted by its coverage degree. We propose a new formulation and a set of valid inequalities for the problem. We also propose to include those constraints on demand during the branch-and-bound algorithm. We present several experiments that show that our formulation outperforms the best one proposed in the literature for all tested instances even when do not include any proposed inequality. Besides we present the optimal solution for several open instances.

KEYWORDS. Integer Programming, Hubs Location, Maximal Covering.

Topics: Combinatorial Optimization, Logistics and Transport

1. Introdução

Hubs são facilidades especiais designadas para atuarem como pontos de troca em sistemas de transporte de produtos ou dados, onde é necessário transferir fluxos entre diversos pares origem e destino (O/D). Em sistemas de distribuição *many-to-many*, ao invés de pares O/D serem ligados diretamente, os fluxos são consolidados através da utilização de hubs, o que implica na redução do total de rotas necessárias para o transporte. Além disso, a consolidação de fluxos através de hubs resulta na redução do custo de transporte devido à exploração de economias de escala. Esse tipo de modelo de sistema de distribuição é amplamente aplicado na aviação [Button, 2002], sistemas logísticos [Ishfaq e Sox, 2011] e redes de telecomunicações [Kim e O'Kelly, 2009].

O problema de localização de hubs (PLH) foi proposto primeiramente por O'Kelly [1986]. Em um PLH, em geral, duas decisões são tomadas: a localização de um número finito de hubs e a alocação de pontos não hub a hubs localizados. Estas decisões geram um custo de atendimento para o fluxo de cada par O/D da rede. Campbell [1994] mostrou que os objetivos em um PLH podem ser diversos, como por exemplo, minimizar o custo total de atendimento, minimizar o maior custo de atendimento e objetivos ligados a cobertura fluxos na rede.

De acordo com o tipo de alocação considerada, cada PLH pode ser classificado em alocação simples (AS), alocação múltipla (AM) ou alocação r . Na AS, cada ponto não hub deve ser atribuído a um único hub localizado enquanto na AM, cada ponto não hub pode ser alocado a quantos hubs localizados forem necessários. Já no caso da alocação r , cada ponto não hub deve ser atribuído a no máximo r hubs localizados, representando uma generalização de alocação simples ($r = 1$) e alocação múltipla ($r = +\infty$). Para mais detalhes a respeito de categorização e definições relativas a PLHs, ver Contreras e O'Kelly [2019] e Farahani et al. [2013].

Dentre os PLHs apresentados por Campbell [1994], um deles é o problema da cobertura máxima p -hub (PCMpH). Neste problema, exatamente p hubs são localizados e um dos tipos de alocação definidas anteriormente (AS, AM ou r) é feita. Dadas as decisões de localização e alocação, o fluxo de cada par O/D recebe o grau de cobertura 1 (totalmente coberto) ou 0 (não coberto). O objetivo então do problema é maximizar a soma de fluxos que são totalmente cobertos.

Peker e Kara [2015] classificaram o tipo de cobertura do PCMpH como binária uma vez que só admite dois tipos de grau de cobertura (1 ou 0) e propuseram uma generalização do problema que admite cobertura parcial. Esta nova versão considera um conjunto discreto de graus de cobertura com valores entre 0 e 1, inclusive, e uma função não crescente que associa o fluxo de cada par O/D a um destes graus, de acordo com o seu custo de atendimento. As decisões de localização e alocação são então tomadas de modo a maximizar a soma ponderada dos fluxos pelos seus respectivos graus de cobertura.

Este trabalho foca no PCMpH com alocação simples e cobertura parcial (AS-CP-PCMpH). Para resolver o problema de maneira exata, propomos uma nova formulação de Programação Linear Inteira Mista (PLIM) e um conjunto de desigualdades válidas. Como o número de desigualdades cresce exponencialmente com o tamanho da instância, propomos gerá-las sob demanda durante o clássico algoritmo de *branch-and-bound*. Mostramos ainda que as desigualdades podem ser separadas utilizando o clássico algoritmo do corte mínimo $s - t$. Nós geramos diversas instâncias de diversos tamanhos, com base nos conjuntos de dados CAB, TR e AP, frequentemente usados para PLHs. Em seguida, comparamos a formulação proposta com e sem a inclusão das desigualdades com a melhor formulação encontrada para o problema, proposta por Peker e Kara [2015]. Resultados mostram que a formulação proposta apresenta uma performance computacional superior, mesmo quando as desigualdades válidas não são utilizadas. O bom desempenho computacional da metodologia proposta possibilita ainda a obtenção da solução ótima de diversas instâncias que até

então estavam em aberto na literatura.

2. Revisão da Literatura

Nesta seção, focamos a revisão de literatura sobre o PCMpH e suas variantes. Tais variantes se diferenciam entre si basicamente pelo tipo de alocação, que pode ser simples (AS) múltipla (AM) ou r e pelo tipo de cobertura, que pode ser binária (CB) ou parcial (CP).

Nos primeiros trabalhos relativos ao PCMpH, o tipo de cobertura considerado era apenas a cobertura binária. Para o AM-CB-PCMpH, Weng et al. [2006] propuseram duas metaheurísticas e uma formulação PLI com $O(n^2)$ variáveis e restrições. Posteriormente, Qu e Weng [2009] propuseram uma metaheurística baseada em reconexão de caminhos. Hwang e Lee [2012] focaram sua atenção no AS-CB-PCMpH. Os autores propuseram uma formulação PLI com $O(n^4)$ variáveis e restrições.

Peker e Kara [2015] introduziram o conceito de cobertura parcial ao PCMpH. Os autores apresentaram duas formulações PLIM com $O(n^2)$ variáveis: uma para o AS-PCMpH e outra para o AM-PCMpH. As formulações propostas pelos autores podem ser aplicadas nos dois tipos de cobertura. Os autores mostraram ainda que as formulações propostas possuem uma performance computacional superior as formulações existentes até aquele momento. Janković et al. [2017] apresentaram heurísticas e novas formulações PLIM para as mesmas variantes abordadas em Peker e Kara [2015]. Apesar de mais recente, para o AS-CP-PCMpH (foco deste trabalho), a formulação proposta pelos autores não superou a formulação proposta por Peker e Kara [2015].

Alguns trabalhos, apesar de recentes, consideram apenas PCMpH com cobertura binária. Silva e Cunha [2017] propuseram uma metaheurística baseada em busca tabu para o AS-CB-PCMpH. Recentemente, Doglio et al. [2020] propuseram um conjunto de desigualdades válidas para a formulação PLIM proposta por Peker e Kara [2015] para a variante AS-CB-PCMpH. Os autores mostraram que o uso das desigualdades propostas possibilitou a obtenção de soluções ótimas para diversas instâncias em aberto. Para a alocação r , encontramos um único trabalho que considera a cobertura binária [Janković e Stanimirović, 2017].

Este trabalho foca no problema AS-CP-PCMpH. As principais contribuições aqui apresentadas são: uma nova formulação PLI e um conjunto de desigualdades válidas.

3. Definição formal do problema e melhor formulação da literatura

Esta seção apresenta uma definição formal para o AS-CP-PCMpH, bem como apresenta a principal formulação PLIM para o problema, proposta por Peker e Kara [2015]. De agora em diante, este problema é referido apenas como PCMpH.

Seja $G = (N, A)$ um grafo completo, onde N e A denotam respectivamente o conjunto de nós e arcos. Seja ainda um conjunto H ($H \subseteq N$) representando os potenciais hubs. Cada arco $(i, j) \in A$ representa um par O/D com um fluxo associado denotado por w_{ij} . O problema possui duas decisões a serem tomadas. A primeira consiste em escolher p hubs em H para serem localizados. A segunda decisão é alocar cada ponto de N a um único hub localizado (alocação simples).

As decisões de localização e alocação geram um custo de atendimento para cada par O/D $(i, j) \in A$. Suponha por exemplo que k e m são dois hubs localizados e o ponto i seja alocado a k e o ponto j seja alocado a m . Neste caso, o par O/D (i, j) é associado ao par de hubs (k, m) , o que gera o custo c_{ij}^{km} , dado por $c_{ij}^{km} = d_{ik} + \alpha d_{km} + d_{mj}$, onde d representa a matriz de distâncias ou tempos entre os nós do grafo e $\alpha \in [0, 1]$ é o fator de desconto para o uso do transporte entre hubs. A Equação que calcula c_{ij}^{km} também é válida quando $k = m$. Neste caso, tem-se $d_{km} = 0$ e o par O/D (i, j) está sendo servido por um único hub ao invés de um par.

Para o cálculo do grau de cobertura do fluxo de cada par O/D, o problema considera um conjunto discreto L com valores entre 0 e 1, de modo que $\{0,1\} \subseteq L$, representando os possíveis graus de cobertura. O problema considera ainda uma função não crescente $f : \mathbb{R} \rightarrow L$ que associa o custo de atendimento de um dado par O/D ao grau de cobertura do seu fluxo.

O PCMpH visa maximizar a soma ponderada dos fluxos pelos seus graus de cobertura. A formulação com melhor performance computacional para o PCMpH foi proposta por Peker e Kara [2015] e é apresentada a seguir. A formulação usa as constantes λ_{ij} , definida para cada par O/D (i, j) como $\lambda_{ij} = \max_{k,m \in H} f(c_{ij}^{km})$. As variáveis da formulação são:

$z_{ij} \rightarrow$ grau de cobertura do par O/D (i, j) .

$x_{ik} \rightarrow$ Variável binária indicando que o ponto i foi alocado a hub localizado k ($x_{ik} = 1$) ou não ($x_{ik} = 0$). Quando $i = k$, a variável indica também se um hub é localizado em k ($x_{kk} = 1$) ou não ($x_{kk} = 0$).

A formulação completa é a que segue:

$$\max \sum_{i \in N} \sum_{j \in N} w_{ij} z_{ij} \quad (1)$$

$$s.a. \sum_{k \in H} x_{ik} = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{k \in H} x_{kk} = p \quad (3)$$

$$z_{ij} \leq \sum_{k \in H} f(c_{ijkm}) x_{ik} + \lambda_{ij} (1 - x_{jm}) \quad \forall i, j \in N, m \in H \quad (4)$$

$$z_{ij} \leq \sum_{k \in H} (f(c_{ijkm}) - 1) x_{ik} + 1 \quad \forall i, j \in N, m \in H \quad (5)$$

$$x_{ik} \leq x_{kk}, \quad \forall i \in N, k \in H \quad (6)$$

$$0 \leq z_{ij} \leq \lambda_{ij}, \quad \forall i, j \in N \quad (7)$$

$$x_{ik} \in \{0, 1\}, \quad \forall i \in N, k \in H \quad (8)$$

O propósito da função objetivo (1) é maximizar a soma ponderada do fluxo de cada par O/D pelo seu grau de cobertura. As restrições (2) garantem que cada ponto não hub vai ser alocado a um único hub localizado (alocação simples). A restrição (3) garante a localização de exatamente p hubs. As restrições (4) e (5) calculam o grau de cobertura do fluxo do par O/D (i, j) (z_{ij}). Estas restrições garantem que z_{ij} vai assumir valor $f(c_{ijkm})$ para os hubs k e m tais que $x_{ik} = x_{jm} = 1$. Apenas as restrições (4) são suficientes para determinar o valor correto de z_{ij} . No entanto, Peker e Kara [2015] mostraram que uso das restrições (5) fortalece a formulação. As restrições (6) garantem que um ponto não hub i só pode ser atribuído a um hub k se este estiver localizado. As restrições (7) e (8) garantem o domínio das variáveis.

4. Contribuições deste trabalho

Este trabalho apresenta duas contribuições PCMpH: uma nova formulação PLI e um conjunto de desigualdades válidas.

4.1. Formulação Proposta

A formulação proposta é similar a formulação proposta por Peker e Kara [2015] e considera as constantes a_{ijkm}^l definidas da seguinte forma:

$$a_{ijkm}^l = \begin{cases} 1 & \text{se o par de hubs } (k, m) \text{ cobre o par O/D } (i, j) \text{ com grau } l(f(c_{ijkm}) = l) \\ 0 & \text{c.c} \end{cases} \quad (9)$$

A formulação proposta usa variáveis binárias $z_{ij}^l, \forall i, j \in N, \forall l \in L$ que indicam se o par O/D (i, j) é coberto com grau de cobertura l ($z_{ij}^l = 1$) ou não ($z_{ij}^l = 0$). A formulação usa ainda as variáveis binárias x já definidas para a formulação (1) - (8). A formulação proposta é a que segue:

$$\max \sum_{i \in N} \sum_{j \in N} \sum_{l \in L} w_{ij} \times l \times z_{ij}^l \quad (10)$$

$$s.a. (2), (3), (6), (8) \quad (11)$$

$$\sum_{l \in L} z_{ij}^l = 1 \quad \forall i, j \in N \quad (12)$$

$$z_{ij}^l \leq \sum_{k \in H} a_{ijkm}^l x_{ik} + (1 - x_{jm}) \quad \forall i, j \in N, m \in H, l \in L, \text{ com } l > 0 \quad (13)$$

$$z_{ij}^l \leq \sum_{k \in H} (a_{ijkm}^l - 1) x_{ik} + 1 \quad \forall i, j \in N, m \in H, l \in L, \text{ com } l > 0 \quad (14)$$

$$z_{ij}^l \in \{0, 1\}, \quad \forall i, j \in N, l \in L \quad (15)$$

A função objetivo (10) é equivalente a (1) e visa maximizar a soma ponderada do fluxo de cada par O/D pelo seu grau de cobertura. As restrições (12) garantem que exatamente um grau de cobertura $l \in L$ é atribuído a um dado par O/D (i, j) . Já as restrições (13) são similares as restrições (4) e garantem o cálculo correto das variáveis z_{ij}^l . Esta variável pode assumir valor 1 toda vez que existem $k, m \in H$ tais que i está alocado a k ($x_{ik} = 1$), j está alocado a m ($x_{jm} = 1$) e o par de hubs (k, m) cobre o par O/D (i, j) com grau de cobertura l ($a_{ijkm}^l = 1$). As restrições (14) são similares as restrições (5) e não são essenciais para o correto cálculo de cada variável z_{ij}^l . Porém, o uso destas servem para fortalecer a formulação.

A formulação (10) - (15) aqui proposta é mais forte ou igual a formulação (1) - (8) da literatura uma vez que para todo par O/D (i, j) , temos $z_{ij} = \sum_{l \in L} l \times z_{ij}^l$ e na formulação proposta cada variável $z_{ij}^l, l \in L$, é calculada em uma restrição diferente.

4.2. Desigualdades válidas

Este trabalho também propõe um conjunto de desigualdades válidas para a formulação proposta (10) - (15). Por definição, cada variável z_{ij}^l só pode assumir valor 1 se existem $k, m \in H$ tais que $x_{ik} = x_{jm} = a_{ijkm}^l = 1$. Podemos ilustrar esta relação entre as variáveis x e z através da construção de um grafo que possui um nó origem em i e um nó destino em j e em que cada caminho nesse grafo aciona exatamente duas variáveis x que são suficientes para que z_{ij}^l possa assumir valor 1. Para ilustrarmos este conceito, consideremos um caso onde um dado par O/D (i, j) é coberto com grau l pelos pares de hub $(k_1, m_3), (k_2, m_1), (k_2, m_2)$ e (k_3, m_3) . A construção do grafo é feita de maneira bem simples e intuitiva. Além do vértice origem i e do vértice destino j , existe uma primeira camada de vértices composta pelos vértices que atuam como primeiro hub em algum par de hubs que cobre (i, j) com grau l (k_1, k_2 e k_3 , no exemplo), além de uma camada de vértices composta pelos vértices que atuam como segundo hub em algum par de hubs que cobre (i, j) com grau l (m_1, m_2 e m_3 no exemplo). O vértice i é conectado com cada vértice da primeira camada,

associando a variável x correspondente. De maneira similar, cada vértice da segunda camada é conectado com o vértice j , associando a variável x correspondente. Arcos conectando a primeira com a segunda camada só são incluídos se o par de hubs correspondente cobre (i, j) com grau l (No exemplo, (k_1, m_3) , (k_2, m_1) , (k_2, m_2) e (k_3, m_3)) associadas a uma constante 1, cada. A Figura 1 ilustra o grafo para o exemplo com as características comentadas anteriormente.

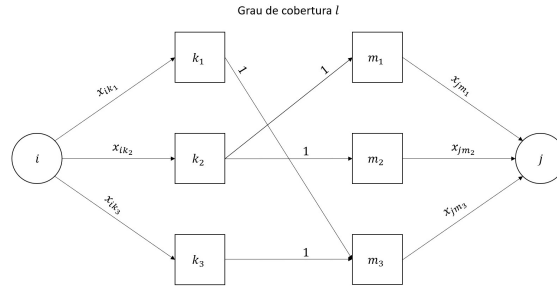


Figura 1: Ilustração da representação de z_{ij}^l como caminhos em um grafo.

Notemos que neste grafo toda combinação de duas variáveis x que fazem com que z_{ij}^l possa assumir valor 1 estão representadas através de um único caminho no grafo. Por exemplo, o caminho $i \rightarrow k_1 \rightarrow m_3 \rightarrow j$ acionam as variáveis x_{ik_1} e x_{jm_3} , que quando ambas assumem valor 1 fazem com que z_{ij}^l também assumam valor 1. A ideia aqui proposta é mostrarmos que uma desigualdade válida pode ser extraída a partir de grafos construídos dessa maneira, sempre que tomamos um subconjunto de vértices S tal que $i \in S$ e $j \notin S$. Para ilustrar essa ideia, considere S como sendo o subconjunto de vértices formado pelos vértices i, k_2, m_1 e m_2 . A Figura 2 destaca tal conjunto.

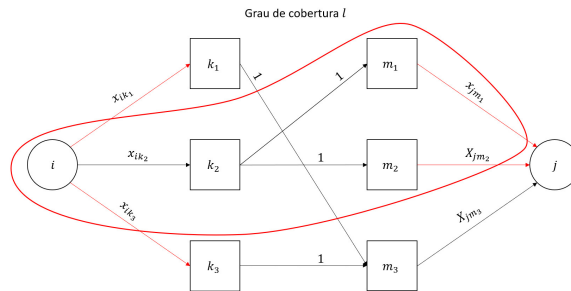


Figura 2: Conjunto S usado para a geração da desigualdade.

Notemos que os arcos que conectam vértices do subconjunto a vértices de fora do subconjunto e vice-versa (destacadas em vermelho) formam um corte em arestas. Em outras palavras, qualquer caminho de i para j necessariamente tem que passar por uma destas arestas destacadas. Unindo a informação prévia com o fato de que z_{ij}^l só assume valor 1 se existir um caminho no grafo, então a seguinte desigualdade é válida para este exemplo:

$$z_{ij}^l \leq x_{ik_1} + x_{ik_3} + x_{jm_1} + x_{jm_2}. \quad (16)$$

Outras desigualdades poderiam ser obtidas de maneira similar, escolhendo outros subconjuntos, desde que i esteja no subconjunto e j não. Uma observação importante é que se o par de hubs (k, m) cobre (i, j) com grau l , então não é recomendável que k esteja no subconjunto e m não ou

vice-versa pois, se isso acontece, a desigualdade continua válida mas teríamos a constante 1 do lado direito da desigualdade e, conseqüentemente, a desigualdade não seria capaz de cortar nenhuma solução fracionária. Vamos agora apresentar as desigualdades propostas de maneira genérica.

Sejam $i, j \in N$ e $l \in L$. Seja ainda o grafo orientado $G_{ij}^l(V_{ij}^l, A_{ij}^l)$ cujo conjunto de vértices V_{ij}^l é formado por i, j e todo ponto $k \in H$ tal que existe $m \in H$ com $a_{ijkm}^l = 1$ ou $a_{ijmk}^l = 1$. Já o conjunto de Arcos A_{ij}^l é formado por todos arcos do tipo (k, m) , onde $a_{ijkm}^l = 1$ associados a uma constante 1, arcos do tipo (i, k) associados a uma variável x_{ik} , onde k é o primeiro hub de um par de hubs que cobre (i, j) com grau l e arcos do tipo (m, j) associados a uma variável x_{jm} , onde m é o segundo hub de um par de hubs que cobre (i, j) com grau l . Seja S_{ij}^l o conjunto de subconjuntos dos vértices de G_{ij}^l , onde cada um destes subconjuntos $S \in S_{ij}^l$ satisfaz as seguintes características: $i \in S, j \notin S$ e se $a_{ijkm}^l = 1$ então $k, m \in S$ ou $k, m \notin S$. Os arcos de A_{ij}^l que conectam S ao seu complementar e vice-versa formam um corte em arcos de G_{ij}^l e portanto só existe caminho de i para j se ao menos uma destes arcos são utilizados. Temos então a seguinte desigualdade válida:

$$z_{ij}^l \leq \sum_{h \notin S \wedge (i,h) \in A_{ij}^l} x_{ih} + \sum_{h \in S \wedge (h,j) \in A_{ij}^l} x_{jh}, \forall i, j \in N, \forall l \in L, \forall S \in S_{ij}^l. \quad (17)$$

Para ilustrar a capacidade de melhoria do uso das desigualdades (17), suponhamos que durante a resolução de uma relaxação da formulação, a solução fracionária encontrada para o exemplo utilizado nesta seção seja $x_{ik_1}=0,1, x_{ik_2}=0,4, x_{ik_3}=0,1, x_{jm_1}=0,1, x_{jm_2}=0,1$ e $x_{jm_3}=0,2$. De acordo com o uso apenas das restrições (13) e (14), teríamos que z_{ij}^l assumiria 0,6. No entanto, este valor se reduziria para 0,4 caso acrescentássemos a desigualdade (16).

4.2.1. Inclusão de desigualdades válidas na formulação

O número de desigualdades em (17) é exponencial, o que torna inviável a geração das mesmas antes da resolução da formulação. Assim, este trabalho propõe resolver a formulação (10) - (15) através de algum resolvedor que permita o acionamento de uma função do tipo *callback* em cada nó da árvore de *branch-and-bound* para inclusão de novas restrições. Dentro do *callback*, o valor corrente da solução relaxada é fornecido e devemos decidir quantas e quais desigualdades do tipo (17) serão adicionadas a formulação. A estratégia de inclusão de desigualdades que adotamos neste trabalho é incluir, para cada $i, j \in N$ e $l \in L$, a desigualdade associada ao conjunto $S \in S_{ij}^l$ que mais viola (17). Tal problema é chamado de Problema da Separação.

Formalmente, o Problema da Separação é definido da seguinte forma. Seja $(\bar{z}, \bar{x})$ uma solução que satisfaz as restrições da formulação (1) - (8) com as restrições de integralidade relaxadas e satisfaz algumas restrições de (17). O Problema da Separação consiste em encontrar, para cada $i, j \in N$ e $l \in L$, o conjunto $S \in S_{ij}^l$ que minimize o lado direito de (17). Dado que os valores de $\bar{x}$ podem ser vistos como capacidades dos arcos de G_{ij}^l , o Problema da Separação é equivalente ao problema Problema do corte mínimo $s - t$ [Dantzig e Fulkerson, 1964].

4.3. Algoritmo proposto

O Algoritmo 1 mostra os passos do nosso método, que é chamado de B&C por se tratar de um algoritmo do tipo *branch-and-cut*. Este algoritmo se inicia na linha 13, onde a formulação (10) - (15) é resolvida com a inclusão de um *callback* de cortes chamado CALLBACK(). Este *callback* é executado após a resolução de cada relaxação na árvore de *branch-and-bound*, recebendo os valores das variáveis x e z , simbolizados por $\bar{x}$ e $\bar{z}$, respectivamente. Quando o *callback* é executado, é

verificado se a diferença percentual relativa entre o valor relaxado da árvore e a melhor solução inteira (*gap*) é maior do que ϵ (linha 2), onde $\epsilon \in [0, 1]$ é um valor pré-definido. Se sim, para todo par todo par O/D (i, j) e para todo grau de cobertura $l > 0$ que não seja muito próximo de 0 ($z_{ij}^l > 0.001$), se os valores das variáveis $\bar{x}$ são fracionários (linha 4), a função CORTEMINIMO() é executada (linha 5), cujo objetivo é resolver o clássico problema do corte mínimo $s - t$ sobre o grafo G_{ij}^l encontrando o conjunto S que minimiza o lado direito de (17). Se o conjunto S encontrado viola (17) (linha 6), então a desigualdade associada é inserida (linha 7). O valor ϵ utilizado é calibrado de modo a balancear o *tradeoff* entre o custo computacional de se incluir novas desigualdades e o ganho obtido na relaxação.

Algoritmo 1 B&C

```

1: Procedimento CALLBACK( $\bar{x}, \bar{z}$ )
2:   Se  $gap > \epsilon$  então
3:     Para todo par O/D  $(i, j)$  e todo  $l \in L \setminus \{0\}$  com  $z_{ij}^l > 0.001$  faça
4:       Se  $\bar{x}$  é fracionário então
5:          $S \leftarrow \text{CORTEMINIMO}(G_{ij}^l)$ 
6:         Se  $\sum_{h \notin S \wedge (i,h) \in A_{ij}^l} \bar{x}_{ih} + \sum_{h \in S \wedge (h,j) \in A_{ij}^l} \bar{x}_{jh} < z_{ij}^l$  então
7:           Insere a desigualdade  $z_{ij}^l \leq \sum_{h \notin S \wedge (i,h) \in A_{ij}^l} x_{ih} + \sum_{h \in S \wedge (h,j) \in A_{ij}^l} x_{jh}$ 
8:         Fim Se
9:       Fim Se
10:     Fim Para
11:   Fim Se
12: Fim Procedimento
13: Resolve a Formulação (10) - (15) tomando CALLBACK() como um callback de inclusão de restrições.
14: Retorna Solução ótima encontrada pelo resolvidor.

```

5. Experimentos Computacionais

Todos os experimentos, incluindo os da literatura, foram realizados em um mesmo computador com processador Intel Core i5-7200U 2.70GHz CPU e 8 GB de memória RAM, tendo o Windows 10 como sistema operacional. A linguagem de programação utilizada foi Julia. A resolução das formulações e todo o gerenciamento da árvore de *branch-and-bound* são feitos pelo resolvidor CPLEX 12.8. As resoluções dos problemas de corte mínimo (linha 5 do Algoritmo 1 são feitas através da biblioteca *SparseMaxFlowMinCut* [Pessoa, 2019].

5.1. Geração das instâncias

Nesta seção são descritas as instâncias usadas neste trabalho. Tais instâncias foram geradas com base em três diferentes bases de dados muito comuns em problemas envolvendo localização de hubs: CAB [O'Kelly, 1986], TR [Tan e Kara, 2007] e AP [Ernst e Krishnamoorthy, 1996].

O conjunto CAB é baseado no transporte aéreo de passageiros entre cidades americanas. Ao todo, o conjunto possui 25 pontos. O conjunto TR possui 81 pontos e é baseado no transporte de carga na Turquia. Já o conjunto AP (Australian Post) é baseado no sistema de entregas postais da cidade de Sydney, na Austrália. O conjunto AP original possui 200 pontos, mas [Ernst e Krishnamoorthy, 1996] propuseram um esquema de agregação capaz de gerar conjuntos menores. Neste trabalho, usamos o esquema de agregação prévio e geramos conjuntos AP com 40 e 50 pontos. Para cada um dos três tipos de conjunto, são fornecidas as matrizes de distâncias e fluxos entre os pontos. No conjunto TR, é fornecido ainda a matriz de tempo de deslocamento entre os pontos.

Instâncias CAB e AP	Instâncias TR
$f(c_{ijk}^{km}) = \begin{cases} 1 & \text{se } c_{ij}^{km} \leq 0,75\beta \\ 0,75 & \text{se } 0,75\beta < c_{ij}^{km} \leq 0,8\beta \\ 0,5 & \text{se } 0,8\beta < c_{ij}^{km} \leq 0,85\beta \\ 0,25 & \text{se } 0,85\beta < c_{ij}^{km} \leq 0,9\beta \\ 0 & \text{c.c} \end{cases}$	$f(c_{ijk}^{km}) = \begin{cases} 1 & \text{se } c_{ij}^{km} \leq 12h \\ 0,75 & \text{se } 12h < c_{ij}^{km} \leq 16h \\ 0,5 & \text{se } 16h < c_{ij}^{km} \leq 20h \\ 0,25 & \text{se } 20h < c_{ij}^{km} \leq 24h \\ 0 & \text{c.c} \end{cases}$

Tabela 1: função $f(c_{ij}^{km})$ utilizada para as instâncias testadas.

Conjunto	#Inst.	N	H	#O/D pares	Valores de α	Valores de p	Valor de ϵ
CAB	16	25	25	300	{0,2; 0,4; 0,6; 0,8}	{2, 3, ..., 5}	0,1
TR	27	81	22	3240	{0,6; 0,8; 0,9}	{2, 3, ..., 10}	0,01
AP40	4	40	40	820	{0,75}	{2, 3, ..., 5}	0,005
AP50	4	50	50	1275	{0,75}	{2, 3, ..., 5}	0,005

Tabela 2: Resumos das instâncias testadas.

Nos conjuntos CAB e AP, todo ponto é considerado também um potencial hub ($H = N$) e os cálculos dos custos c_{ij}^{km} são feitos a partir das distâncias entre os pontos. Já o conjunto TR fornece 22 índices, que indicam quais são os potenciais hubs. Além disso, neste conjunto, os custos c_{ij}^{km} são calculados a partir dos tempos entre os pontos.

Com relação ao conjunto L de graus de cobertura, nós seguimos Peker e Kara [2015] e Janković et al. [2017], considerando $L = \{0; 0,25; 0,5; 0,75; 1\}$, para todos os conjuntos. O cálculo da função $f(c_{ij}^{km})$ é feito de acordo com as equações da Tabela 1. Para esta tabela, a constante β representa o custo ótimo de um outro problema de localização de hubs conhecido como problema do p -hub centro com alocação simples [Ernst et al., 2009].

Um outro ponto importante dos experimentos é que todos os conjuntos de dados possuem as matrizes de entrada (distância ou tempo) simétricas. Assim, para todo $i, j \in N$ com $i \neq j$, os pares O/D (i, j) e (j, i) são agrupados em único par O/D com fluxo $w_{ij} + w_{ji}$. Além disso, os fluxos foram normalizados, onde cada fluxo foi dividido pela soma total de todos eles.

Diversas instâncias são geradas a partir de cada conjunto, variando os valores de α e de p . A variação dos valores de α foram as mesmas usadas por Peker e Kara [2015] e Janković et al. [2017]. A Tabela 2 mostra um resumo dos experimentos. Esta tabela mostra também os valores de ϵ (2 do Algoritmo 1) arbitrados para cada conjunto. Quanto mais difícil a instância mais desigualdades (17) tendem a ser necessárias e, conseqüentemente, o valor de ϵ arbitrado é menor.

5.2. Resultados Comparativos

Nesta seção são apresentados resultados computacionais comparativos de três metodologias: a Formulação (1) - (8) proposta por Peker e Kara [2015], a Formulação (10) - (15) proposta neste trabalho sem o uso de desigualdades do tipo (17) e esta mesma formulação com a adição de desigualdades (17) sob demanda usando o algoritmo 1.

A Tabela 3 mostra um sumário dos resultados comparativos. As colunas *Conjunto* e α indicam respectivamente o nome do conjunto e o valor de α que estão sendo testados. A Coluna *#opt* mostra o número de instâncias para um dado conjunto e um dado valor de α . A Coluna *#opt* mostra o número de soluções comprovadamente ótimas obtidas com até 10.000s de execução. A Coluna *Gap₀* mostra a diferença percentual relativa entre o limite superior obtido no nó raiz e a

Cojunto	α	#Inst.	#opt	Formulação (3) - (10) Peker e Kara (2015)			Formulação Proposta sem desigualdades (17)			Formulação Proposta com desigualdades (17)				
				Média			Média			Média				
				Gap_0	#Nós	Tempo(s)	#opt	Gap_0	#Nós	Tempo(s)	#opt	Gap_0	#Nós	Tempo(s)
CAB	0,2	4	4	3,7	5634,8	194,5	4	3,1	295,3	57,2	4	3,1	521,0	78,8
CAB	0,4	4	4	4,1	4158,3	145,4	4	2,1	108,0	33,3	4	2,2	142,5	52,9
CAB	0,6	4	4	5,8	4434,3	143,5	4	2,2	111,5	28,2	4	2,2	130,5	51,6
CAB	0,8	4	4	6,1	6055,5	178,9	4	1,9	117,0	20,7	4	1,9	97,3	32,5
TR	0,6	9	0	2,8	11219,9	>10000,0	9	1,3	410,4	2922,1	9	1,0	679,0	2793,4
TR	0,8	9	0	3,8	12155,3	>10000,0	9	1,0	158,8	1665,3	9	0,9	98,6	1251,3
TR	0,9	9	0	3,8	11588,2	>10000,0	9	0,7	60,3	992,3	9	0,6	65,2	956,2
AP40	0,75	4	2	1,6	10968	>6702,8	4	0,9	1228,5	1656,3	4	0,8	258,5	893,0
AP50	0,75	4	0	1,2	22428	>10000,0	1	0,7	1854,8	>8886,9	4	0,6	566,0	4149,8

Tabela 3: Resultados comparativos

solução ótima. A Coluna *#Nós* apresenta o número de nós na árvore de *branch-and-bound*. A coluna *Tempo(s)* apresenta o tempo total de CPU em segundos. Os valores apresentados em cada uma das Colunas *Gap₀*, *#Nós* e *Tempo(s)* são uma média dentre as instâncias do conjunto. Para cada conjunto e cada valor de α , nós marcamos em negrito o menor tempo computacional médio dentre as três metodologias.

Analisando os resultados da Tabela 3, podemos notar que a formulação proposta apresenta menor tempo médio para todos os conjuntos e valores de α testados, mesmo quando as desigualdades (17) não são consideradas. Além disso, a formulação da literatura não resolve nenhuma instância dentre as 27 do conjunto TR e não resolve 6 dentre 8 instâncias AP.

Outro ponto importante a ser observado na Tabela 3 é que, para as instâncias CAB, o uso das desigualdades (17) acarretam em um maior tempo computacional quando comparado a formulação proposta sem o uso destas desigualdades. Para essas instâncias, inclusive o gap na raiz e número médio de nós são, em geral, menores ou iguais quando não foram utilizadas as desigualdades. Isso acontece não só porque nas instâncias pequenas, o tempo gasto para inclusão de cortes pode não compensar, mas também porque quando o *callback* de inclusão de desigualdades é utilizado no CPLEX, este resolvidor desabilita automaticamente algumas melhorias genéricas que ele próprio possui, o que pode levar a resultados piores. Apesar disso, nas instâncias maiores (TR e AP), o uso das desigualdades (17) compensa o não uso destas melhorias, pois acarreta em menores médias no gap da raiz, no número de nós e no tempo total, para todas as instâncias desses dois conjuntos. Além disso, 3 das 4 instâncias AP50 não foram resolvidas pela formulação proposta dentro de 1000s de execução quando desigualdades (17) não são utilizadas.

5.3. Estatísticas do Algoritmo 1

Nesta Seção apresentamos outras estatísticas para a aplicação da formulação (10) - (15) proposta neste trabalho com a adição de desigualdades (17) sob demanda usando o algoritmo 1. A Tabela 4 tais resultados. Com exceção da Coluna *#opt*, todas as demais apresentam valores médios considerando todas as instâncias do conjunto. A Coluna *#CM* indica o número de vezes que a função CORTEMINIMO (linha 5 do Algoritmo 1) foi utilizada. *#Des.* indica o número de desigualdades (17) efetivamente adicionadas a formulação (linha 7 do Algoritmo 1). *Tempo CM* indica o tempo total de CPU em segundos gasto pela função CORTEMINIMO. *Tempo total* indica o tempo total de CPU em segundos gasto em todo o algoritmo.

Observando a Tabela 4, podemos concluir que o número médio de vezes que a função CORTEMINIMO é executada (Coluna *#CM*) é significativamente grande em todos os grupos de

Conjunto	α	#Inst.	#opt	Gap ₀	#Nós	Média		Tempo	
						#CM	#Des.	CM	Total
CAB	0,2	4	4	3,1	521,0	393,0	73,3	0,2	78,8
CAB	0,4	4	4	2,2	142,5	449,8	58,8	0,2	52,9
CAB	0,6	4	4	2,2	130,5	467,3	44,5	0,2	51,6
CAB	0,8	4	4	1,9	97,3	441,8	39,3	0,2	32,5
TR	0,6	9	4	1,0	679,0	260017,0	1603,2	42,1	2793,4
TR	0,8	9	9	0,9	98,6	113316,9	791,4	18,8	1251,3
TR	0,9	9	9	0,6	65,2	41566,2	325,4	6,7	956,2
AP40	0,75	4	4	0,8	258,5	323707,3	1610,0	160,6	893,0
AP50	0,75	4	4	0,6	566,0	731199,0	2693,0	498,6	4149,8

Tabela 4: Estatísticas específicas do Algoritmo 1

instâncias testadas porém o tempo médio gasto com a execução desta função (Coluna *Tempo(s) CM*) é relativamente pequeno quando comparado ao tempo médio total de execução (Coluna *Tempo(s) CM*). Outro ponto importante é que o número médio de desigualdades (17) efetivamente adicionadas à formulação (Coluna *#Des.*) é relativamente pequeno. Apesar disso, o uso dessas desigualdades se mostra de extrema importância para a obtenção da solução ótima dentro do limite de tempo, conforme mostrou a Seção 5.2.

6. Conclusões

Nós propusemos neste trabalho uma nova formulação e conjunto de desigualdades válidas para o Problema da Cobertura Máxima p-hub com Cobertura Parcial. Como a quantidade de desigualdades propostas cresce exponencialmente com o tamanho da instância, nós propusemos gerá-las sob demanda durante o clássico algoritmo de *branch-and-bound*.

Para comprovar o ganho da inclusão das restrições propostas, diversos experimentos computacionais foram conduzidos sobre 51 instâncias de diversos tamanhos geradas com base no conjunto de pontos CAB, TR e AP. Os experimentos mostraram que o método proposto apresentou um menor tempo computacional do que o apresentado pela formulação da literatura para todas as instâncias comparadas, mesmo quando as desigualdades propostas não são utilizadas. Além disso, o uso das desigualdades resulta em menores médias no gap da raiz, no número de nós e no tempo total para todas as instâncias dos dois maiores conjuntos de instâncias.

Como trabalhos futuros, pretende-se propor desigualdades para a variante do problema com alocação múltipla, também proposta em Peker e Kara [2015].

Referências

- Button, K. (2002). Debunking some common myths about airport hubs. *Journal of Air Transport Management*, 8(3):177 – 188.
- Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):387–405.
- Contreras, I. e O’Kelly, M. (2019). Hub location problems. In *Location science*, p. 327–363. Springer.
- Dantzig, G. B. e Fulkerson, D. R. (1964). On the max flow min cut theorem of networks. *The RAND Corporation*.

- Doglio, P., Roboredo, M. C., e Pessoa, A. A. (2020). Um modelo de programação linear inteira mista para um problema de localização de hubs com cobertura máxima. In *Anais do LII SBPO*, João Pessoa-PB. SOBRAPO.
- Ernst, A. T., Hamacher, H., Jiang, H., Krishnamoorthy, M., e Woeginger, G. (2009). Uncapacitated single and multiple allocation p-hub center problems. *Computers & Operations Research*, 36(7): 2230–2241.
- Ernst, A. T. e Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location science*, 4(3):139–154.
- Farahani, R. Z., Hekmatfar, M., Arabani, A. B., e Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096–1109.
- Hwang, Y. H. e Lee, Y. H. (2012). Uncapacitated single allocation p-hub maximal covering problem. *Computers & Industrial Engineering*, 63(2):382–389.
- Ishfaq, R. e Sox, C. R. (2011). Hub location–allocation in intermodal logistic networks. *European Journal of Operational Research*, 210(2):213 – 230.
- Janković, O., Mišković, S., Stanimirović, Z., e Todosijević, R. (2017). Novel formulations and VNS-based heuristics for single and multiple allocation p-hub maximal covering problems. *Annals of Operations Research*, 259(1-2):191–216.
- Janković, O. e Stanimirović, Z. (2017). A general variable neighborhood search for solving the uncapacitated r-allocation p-hub maximal covering problem. *Electronic Notes in Discrete Mathematics*, 58:23–30.
- Kim, H. e O’Kelly, M. E. (2009). Reliable p-hub location problems in telecommunication networks. *Geographical Analysis*, 41(3):283–306.
- O’Kelly, M. E. (1986). A quadratic integer program for the location of interacting hub facilities. *European journal of operational research*, 32(3):393–404.
- Peker, M. e Kara, B. Y. (2015). The p-hub maximal covering problem and extensions for gradual decay functions. *Omega*, 54:158–172.
- Pessoa, A. (2019). Sparsemaxflowmincut. <https://github.com/artalvpes/SparseMaxFlowMinCut/>. Acessado: 2020-06-29.
- Qu, B. e Weng, K. (2009). Path relinking approach for multiple allocation hub maximal covering problem. *Computers & Mathematics with Applications*, 57(11):1890–1894.
- Silva, M. R. e Cunha, C. B. (2017). A tabu search heuristic for the uncapacitated single allocation p-hub maximal covering problem. *European Journal of Operational Research*, 262(3):954–965.
- Tan, P. Z. e Kara, B. Y. (2007). A hub covering model for cargo delivery systems. *Networks*, 49(1): 28–39.
- Weng, K., Yang, C., e Ma, Y. (2006). Two artificial intelligence heuristics in solving multiple allocation hub maximal covering problem. In *International Conference on Intelligent Computing*, p. 737–744. Springer.