UMA FORMULAÇÃO E UM ALGORITMO EXATO PARA O PROBLEMA DE LOCALIZAÇÃO COMPETITIVA COM MÁXIMA COBERTURA

José Gentile
Departamento de Engenharia de Produção – Universidade Federal Fluminense
Rua Passo da Pátria 156, 24210-240, Niterói, RJ, Brasil
josegentile@terra.com.br

Marcos Costa Roboredo
Departamento de Engenharia de Produção – Universidade Federal Fluminense
Rua Passo da Pátria 156, 24210-240, Niterói, RJ, Brasil
marcos.producao.uff@gmail.com

Artur Alves Pessoa
Departamento de Engenharia de Produção – Universidade Federal Fluminense
Rua Passo da Pátria 156, 24210-240, Niterói, RJ, Brasil
artur@producao.uff.br

RESUMO

O Problema de Localização Competitiva com Máxima Cobertura é formado por duas firmas, líder e seguidora, que competem entre si, localizando facilidades em um mercado inexplorado, de modo a atender a demanda de consumidores. A competição é sequencial, de modo que a líder inicialmente localiza suas facilidades, respeitando uma restrição orçamentária sabendo que, em seguida, a seguidora fará o mesmo. Após as decisões de localização, a demanda de cada consumidor é totalmente atendida pela facilidade mais próxima, localizada ou pela líder ou pela seguidora caso esta facilidade esteja localizada a uma distância máxima pré-especificada. O problema consiste em decidir onde a líder localizará suas facilidades, de modo a maximizar sua demanda atendida, sabendo que a seguidora também otimizará a sua. Este problema de otimização em dois níveis é resolvido de maneira exata pela primeira vez neste trabalho através de uma formulação de Programação Inteira, que é resolvida por um algoritmo *branch-and-cut*.

PALAVRAS CHAVE. Otimização em dois níveis, Programação Inteira, Localização Competitiva

Área Principal: Otimização Combinatória.

ABSTRACT

The competitive location problem with maximum coverage is composed of two firms, leader and follower, competing in an open market to serve customer demands. Initially, the leader places its facilities, under a budget constraint, knowing that the follower will react by doing the same. After the location decisions, the demand of each customer is served by the closest facility, placed by the leader or the follower, if this facility is placed within a coverage radius. The problem consists of deciding where the leader will place its facilities, considering it aims to maximize the demand it attends, knowing that the follower will try to maximize its demand too. The problem is formulated as a Mixed Integer Programming Problem, which is solved for the first time by a branch-and-cut algorithm.

KEYWORDS. Bilevel, Integer Programming, Competitive Location Problem

Main area: Combinatorial Optimization.

1. Introdução

A decisão de localização é uma questão crucial do planejamento estratégico de uma firma. Grande parte das pesquisas existentes a respeito de localização não consideram uma eventual firma concorrente oferecendo o mesmo bem ou serviço. Neste contexto, surgem os chamados problemas de localização competitiva (PLC). Nestes problemas, são consideradas duas ou mais firmas não cooperativas competindo para atender a demanda de consumidores em um dado mercado. A demanda de cada consumidor é totalmente ou parcialmente atendida por facilidades localizadas pelas firmas seguindo alguma regra de preferência do consumidor. Os PLC, assim como os problemas de localização clássicos, recebem algumas classificações como por exemplo, de acordo com o tipo de espaço de decisão (discreto ou contínuo) e de acordo com o tipo de decisão (sequencial ou simultânea). Além disso, diferentes objetivos podem ser perseguidos pelas firmas ao localizar suas facilidades, como por exemplo maximizar sua fração do mercado, seu lucro, o número de consumidores servidos, dentre outros. Uma revisão de diversos modelos e pesquisas a respeito de PLC pode ser encontrada em [Friesz et al. 1988], [Eiselot e Laporte 1993], [Eiselot e Laporte 1997] e [Kress e Pesch 2012].

O problema tratado neste artigo visa incorporar competição ao clássico problema de localização com cobertura máxima (PLCM) proposto por [Church e Velle 1974]. O PLCM considera a decisão de localização de facilidades de uma firma em um mercado. Após esta decisão, a demanda de um consumidor é dita ser atendida por esta firma se existe alguma facilidade localizada a no máximo uma distância pré-especificada deste. Quando isto acontece, a demanda do consumidor é atendida e este consumidor é dito estar coberto pela facilidade. Assim, o objetivo do problema é decidir onde uma firma deve localizar suas facilidades de modo a maximizar a demanda total atendida.

Para adicionar competição ao problema prévio, consideramos duas firmas não cooperativas, chamadas de líder e seguidora. A líder entra primeiro no mercado localizando suas facilidades respeitando uma restrição orçamentária sabendo que a seguidora em seguida reagirá fazendo o mesmo. Consideramos que a líder conhece antecipadamente o orçamento e os possíveis locais de localização da seguidora e, além disso, que a líder assume que a seguidora tomará sua decisão de localização maximizando a sua própria demanda. Após a decisão de localização das duas firmas, a demanda de cada consumidor será totalmente atendida pela firma (líder ou seguidora) que localizar a facilidade mais perto deste (empates são quebrados em favor de facilidades da seguidora), desde que esta facilidade cubra este consumidor. O objetivo do problema tratado neste artigo é otimizar a decisão da líder, em outras palavras, decidir onde a líder deve localizar suas facilidades de modo a maximizar a demanda atendida por esta. A este problema, damos o nome de Problema de Localização Competitiva com Máxima Cobertura (PLCMC).

O PLCMC foi proposto inicialmente por [Serra et al. 1994], onde os autores propuseram além do problema uma simples heurística para resolvê-lo. Mais tarde, [Plastria e Vanhaverbeck 2008] revisitaram o problema e propuseram um modelo de Programação Linear Inteira (PLI) para o caso particular onde a seguidora localiza apenas uma facilidade. Além do PLCMC, os autores trataram mais dois problemas similares também considerando o caso particular prévio. Um deles assume que a líder otimiza sua demanda atendida no pior caso enquanto o outro que a líder minimiza o seu maior arrependimento. Estes dois últimos problemas foram tratados no caso geral por [Roboredo 2012] e [Gentile et al. 2014]. Recentemente, [Seyhan 2012] tratou o problema heuristicamente propondo um modelo PLI para o PLCMC onde a resposta da seguidora é dada através de uma heurística gulosa, que é representada através de variáveis e restrições no modelo.

O PLCMC é um problema de otimização em dois níveis inteiro (2-PONI) onde o problema de primeiro nível consiste de escolher a estratégia de localização da líder enquanto o problema de segundo nível consiste de escolher a estratégia da seguidora. Na literatura, existem poucos métodos exatos genéricos para resolver os 2-PONI. Algoritmos *branch-and-bound* foram propostos por [Moore e Bard 1990] e [Bard e Moore 1992] para o caso linear, mas estes foram aptos a resolver instâncias com no máximo 10 variáveis inteiras e 35 variáveis binárias no problema de primeiro nível. Algumas técnicas são baseadas em características específicas do problema como método de decomposição em desigualdades superválidas [Israeli e Wood 2002] e [O'Hanley e Church 2011], algoritmos de plano corte [Taskin et al. 2009] e [Roboredo e Pessoa 2013], entre outras.

Este trabalho tem como objetivo propor pela primeira vez um modelo PLI exato para o PLCMC no caso geral. O modelo aqui proposto possui um número polinomial de variáveis e um número exponencial de restrições. Assim, as restrições são geradas como cortes durante um algoritmo de *branch-and-cut*. Além disso, com o objetivo de acelerar o algoritmo proposto, nós apresentamos uma heurística capaz de transformar soluções relaxadas fracionárias em inteiras, que é executada durante algumas separações de cortes. Experimentos computacionais mostram que o nosso algoritmo é capaz de resolver instâncias com até 289 consumidores com líder e seguidora localizando até no máximo 5 facilidades cada, em razoável tempo computacional.

O restante deste trabalho está dividido da seguinte maneira: A parte 2 apresenta a descrição formal do problema. Na parte 3 é apresentada a formulação para o PLCMC, enquanto na parte 4 são mostrados diversos resultados obtidos em experimentos computacionais.

2. O problema

Considere um mercado inexplorado composto por um conjunto $J = \{1,...,n\}$ de consumidores, onde cada consumidor $j \in J$ possui uma demanda w_j. Com o intuito de atender à demanda destes consumidores, duas firmas não cooperativas chamadas, uma de líder e outra de seguidora, entrarão neste mercado localizando um conjunto de facilidades de modo sequencial. Sejam L e F os conjuntos das potenciais facilidades para a líder e para a seguidora respectivamente. Para cada consumidor $j \in J$ e cada potencial facilidade $i \in L \cup F$, é conhecida a distância entre estes, que é denotada de d_{ij}. Para a localização de uma facilidade $i \in L \cup F$, é necessário gastar um custo fixo f_i. Líder e seguidora possuem respectivamente os orçamentos B^L e B^F para gastar com a localização de suas facilidades.

A líder inicialmente decide onde localizar suas facilidades, respeitando o seu orçamento, sabendo que logo após, a seguidora fará o mesmo. Após as duas firmas localizarem suas facilidades, a demanda de cada consumidor $j \in J$ somente será atendida caso exista uma facilidade da líder ou da seguidora dentro do raio de cobertura do consumidor j denotado por δ_j. Caso exista mais de uma facilidade dentro do raio de cobertura, a demanda será totalmente atendida pela facilidade mais próxima. Empates são quebrados em favor da seguidora e empates entre facilidades de uma mesma firma são quebrados arbitrariamente.

O problema de localização competitiva com máxima cobertura (PLCMC) consiste em decidir onde a líder deve localizar suas facilidades, de modo a maximizar a sua demanda, considerando-se que a decisão da seguidora será tomada visando maximizar a sua própria demanda.

3. Modelo para o PLCMC

Apesar do PLCMC ser um problema de otimização em dois níveis, vamos mostrar que este admite uma formulação de PLI. Seja Φ o conjunto de todas as possíveis estratégias da seguidora. Assim, cada $S^F \in \Phi$ é um subconjunto de facilidades de F, tal que $\sum_{i \in S^F} f_i \leq B^F$.

Define-se então as seguintes variáveis binárias:

x_i^L - indica que a líder definiu a localidade i para ser ocupada

x_k^F - indica que a seguidora definiu a localidade k para ser ocupada;

y_{ij}^L - é igual a 1 caso a localidade i tenha sido ocupada pela líder e esta é a facilidade mais próxima do consumidor j dentre as ocupadas pela líder, e igual a 0 caso contrário;

y_{kj}^F - é igual a 1 caso a localidade k tenha sido ocupada pela seguidora e esta é a facilidade mais próxima do consumidor j dentre as ocupadas pela seguidora, e igual a 0 caso contrário;
z_j^L - é igual a 1 caso o consumidor j tenha sua demanda atendida pela líder;
z_j^F - é igual a 1 caso o consumidor j tenha sua demanda atendida pela seguidora;
v_{ij} - é igual a 1 caso a localidade i não esteja ocupada pela líder, e nenhuma facilidade mais próxima do consumidor j do que i esteja ocupada pela líder; é igual a 0 caso contrário.

Para a utilização da variável v_{ij} foi necessário inicialmente considerar a constante θ(i, j), que representa a i-ésima localidade da líder mais próxima do consumidor j. Assim, para um determinado consumidor j, θ(1, j) é a localidade possível de ser ocupada pela líder mais próxima do consumidor j. θ(2, j) é a segunda localidade mais próxima do consumidor j e assim sucessivamente até todas as possíveis localidades da líder. Empates são quebrados arbitrariamente.

A formulação para o PLCMC, fica da seguinte forma:

$$\max \quad \sum_{j\in J} w_j z_j^L \tag{1}$$

$$\text{s.a.} \quad \sum_{i\in L} f_i x_i^L \le B^L \tag{2}$$

$$\sum_{k\in F} f_k x_k^F \le B^F \tag{3}$$

$$y_{ij}^L \le x_i^L \qquad \forall j\in J, \forall i\in L \tag{4}$$

$$y_{kj}^F \le x_k^F \qquad \forall j\in J, \forall k\in F \tag{5}$$

$$\sum_{i\in L} y_{ij}^L = 1 \qquad \forall j\in J \tag{6}$$

$$\sum_{k\in F} y_{kj}^F = 1 \qquad \forall j\in J \tag{7}$$

$$z_j^L + z_j^F \le 1 \qquad \forall j\in J \tag{8}$$

$$z_j^L \le \sum_{i\in L|d_{ij}\le\delta_j} y_{ij}^L \qquad \forall j\in J \tag{9}$$

$$z_j^F \le \sum_{k\in F|d_{kj}\le\delta_j} y_{kj}^F \qquad \forall j\in J \tag{10}$$

$$z_j^L \ge \sum_{k\in L|d_{kj}\le d_{ij}} y_{kj}^L - \sum_{k\in F|d_{kj}\le d_{ij}} y_{kj}^F \qquad \forall j\in J, \forall i\in L|d_{ij}\le\delta_j \tag{11}$$

$$z_j^F \ge \sum_{k\in F|d_{kj}\le d_{ij}} y_{kj}^F - \sum_{k\in L|d_{kj}< d_{ij}} y_{kj}^L \qquad \forall j\in J, \forall i\in F|d_{ij}\le\delta_j \tag{12}$$

$$v_{\theta(1,j)j} = 1 - x_{\theta(1,j)}^L \qquad \forall j\in J \tag{13}$$

$$v_{ij} \le 1 - x_i^L \qquad \forall i\in L, \forall j\in J \tag{14}$$

$$v_{\theta(i,j)j} \ge v_{\theta(i+1,j)j} \qquad \forall i=1,\dots,L-1, \forall j\in J \tag{15}$$

$$y_{\theta(1,j)j}^L = 1 - v_{\theta(1,j)} \qquad \forall j\in J \tag{16}$$

$$y_{\theta(i,j)j}^L = v_{\theta(i-1,j)j} - v_{\theta(i,j)j} \qquad \forall i=2,\dots,L, \forall j\in J \tag{17}$$

$$\sum_{j\in J} w_j z_j^F \ge \sum_{j\in J|\min\{d_{kj}|k\in S^F\}\le\delta_j} w_j \left(1 - \sum_{i\in L|d_{ij}<\min\{d_{kj}|k\in S^F\}} y_{ij}^L\right) \qquad \forall S^F\in\Phi \tag{18}$$

$$x_i^L, x_k^F, y_{ij}^L, y_{kj}^F, z_j^L, z_j^F \in \{0,1\} \qquad \forall i \in L, \forall k \in F, \forall j \in J \tag{19}$$

A função objetivo (1) visa maximizar a demanda total atendida pela líder. As restrições (2) e (3) asseguram que os orçamentos da líder e da seguidora serão respeitados. As restrições (4) e (5) asseguram a consistência entre as variáveis x^L e y^L e entre x^F e y^F respectivamente. Em outras palavras, uma variável y da líder ou da seguidora só pode assumir valor 1 caso a facilidade associada esteja realmente localizada. As restrições (6) e (7) asseguram que para cada consumidor j, existe uma única facilidade mais próxima deste localizada pela líder e pela seguidora respectivamente. As restrições (8) asseguram que um consumidor j não pode ser atendido pela líder e pela seguidora ao mesmo, apenas por uma ou por nenhuma delas. As restrições (9) e (10) asseguram a consistência entre as variáveis y^L e z^L e entre y^F e z^F respectivamente, de modo que para que um consumidor seja atendido pela líder/seguidora, é necessário, pelo menos, que a facilidade mais próxima deste consumidor localizada pela líder/seguidora esteja dentro do raio de cobertura. As restrições (11) garantem que o consumidor j será atendido pela líder, caso a facilidade mais próxima deste consumidor, localizada pela líder, esteja dentro do raio de cobertura e não exista uma facilidade localizada pela seguidora, mais próxima do consumidor j do que esta da líder. As restrições (12) asseguram o mesmo com relação ao atendimento do consumidor j pela seguidora.

As restrições (13) - (17) foram criadas para garantir as definições das variáveis y^L , ou seja, que estas só assumam valor 1 quando estiverem efetivamente associadas à facilidade mais próxima localizada pela líder. As restrições (13) e (14) asseguram a consistência entre as variáveis x^L e v. As restrições (13) garantem que se a localidade mais próxima do consumidor j foi ocupada pela líder i, v_{ij} desta localidade seja igual a 0. Caso esta localidade mais próxima não tenha sido ocupada, v_{ij} seria igual a 1. As restrições (14) asseguram que se uma localidade i foi ocupada pela líder, v_{ij} seja igual a 0 (se ela não foi ocupada, v_{ij} pode ser 0 ou 1). As restrições (15) garantem que se v_{ij} referente a i-ésima localidade mais próxima do consumidor j é 0, então todas as posteriores (de i+1 até L) também são 0. As restrições (16) e (17) asseguram as consistências entre as variáveis y^L e v. As restrições (16) asseguram que se a localidade mais próxima do consumidor j foi ocupada pela líder, y^L é igual a 1 e v é igual a 0. Caso contrário, y^L é igual a 0 e v é igual a 1. As restrições (17) garantem que se a i-ésima localidade mais próxima do consumidor j foi a ocupada mais próxima do consumidor j, então $y^L_{\theta(i,j)j}$ é igual a 1 e para tal, $v_{\theta(i-1,j)j}$ é igual a 1 e $v_{\theta(i,j)j}$ é igual a 0.

Cabe aqui mencionar que as restrições (13) - (17) poderiam ser substituídas pelas restrições (20), com a eliminação da variável v_{kj} e da constante θ(k,j). Apesar da descrição bem mais simples que as anteriores, esta família de restrições corresponde a O (|J| $|L|^2$) desigualdades. Testes realizados provaram que a utilização das restrições (13) - (17) apresentam melhores resultados que a utilização da restrição (20), bem como há rapidamente a explosão do número de restrições no modelo neste último caso.

$$t_{kj} + s_i \leq 1 \quad \forall j \in J, \forall k \in L, \forall i \in L \mid d_{ij} < d_{kj} \tag{20}$$

Finalmente, as restrições (18) são as que asseguram que a seguidora fará o melhor para si. Em outras palavras, estas restrições garantem que, dada a estratégia de localização tomada pela líder, a seguidora escolherá aquela que maximiza a sua demanda. Note que o lado direito das restrições (18) mostra a demanda atendida pela seguidora, dada a sua estratégia S^F e a estratégia da líder, definida implicitamente pelas variáveis y^L . Mais explicitamente, esta demanda é dada pela diferença entre toda a demanda que a seguidora consegue cobrir e a soma das demandas em que a líder possui facilidades localizadas mais próximas.

Para cada estratégia viável da seguidora S^F, há uma restrição em (18). Como este número de restrições tende a ser exponencial, as restrições são inseridas durante um algoritmo de *branch-and-cut*. Para tanto, será proposta uma formulação de PLI exata para problema da separação de cortes em (18).

Dada uma solução relaxada para o modelo (1) - (19) satisfazendo as restrições (1) - (17) e algumas restrições em (18), o problema da separação consiste de encontrar a estratégia da seguidora que maximiza a violação da restrição, ou seja, que maximiza o lado direito em (18). A seguir, vamos descrever uma modelo de PLI exato para a separação de (18). Seja $\bar{y}^L$ o vetor com o valor da solução relaxada associada a y^L. Sejam ainda as seguintes variáveis binárias para o modelo da separação:

s_k – é igual a 1 se a seguidora definir a localidade k para ser ocupada, e igual a 0 caso contrário;
t_{kj} - é igual a 1 caso a localidade k tenha sido ocupada pela seguidora e esta é a facilidade mais próxima do consumidor j dentre as ocupadas pela seguidora, e igual a 0 caso contrário;

O modelo PLI da separação segue a seguir:

$$\max \quad \sum_{j\in J}\sum_{k\in F}\left(w_j \sum_{i\in L | d_{kj} \le d_{ij} \wedge d_{kj} \le \delta_j} \bar{y}^L_{ij} \right) t_{kj} \tag{21}$$

$$\text{s.a.} \quad \sum_{k\in F} f_k \; s_k \le B^F \tag{22}$$

$$t_{kj} \le s_k \qquad \forall\, k \in F, \forall\, j \in J \tag{23}$$

$$\sum_{k\in F} t_{kj} = 1 \qquad \forall\, j \in J \tag{24}$$

$$s_k, t_{kj} \in \{0,1\} \qquad \forall\, k \in F, \forall\, j \in J \tag{25}$$

A função objetivo (21) visa maximizar a violação nas restrições (18), que é dada pela demanda total atendida pela seguidora. As restrições (22)-(25) garantem que as variáveis t e s representem uma solução viável para a seguidora.

Neste contexto, o nosso método exato é um algoritmo de branch-and-cut aplicado à formulação (1) - (19), onde as restrições (18) são geradas sob demanda. A implementação e todo gerenciamento da árvore de *branch and bound* são feitas através do resolvedor CPLEX.

Note que, para que uma solução seja viável para a formulação (1) - (19), é necessário que ela satisfaça duas condições: a primeira é que deve ser inteira e a segunda é que estratégia do seguidor representada por x^F e y^F deve ser ótima dada a estratégia do líder representada por x^L e y^L. A segunda condição garante que todas as restrições (18) (incluindo as que não foram inseridas pelo algoritmo *branch-and-cut*) sejam satisfeitas. Portanto, em cada nó da árvore de *branch-and-bound*, cada solução relaxada calculada pode ser inteira ou fracionária, não sendo necessariamente viável quando ela é inteira.

Com o objetivo de acelerar o método proposto, foi definido um parâmetro ε de modo que, enquanto o *gap* for maior ou igual a ε, são separados cortes tanto para soluções relaxadas inteiras quanto para fracionárias. Quando o *gap* se torna menor que ε, são separados cortes apenas para soluções relaxadas inteiras. Também limitamos o número de separações sobre soluções fracionárias a 200 separações, tendo em vista que a inserção de cortes em nós muito elevados na árvore de *branch-and-bound*, não representa praticamente nenhum ganho na solução do algoritmo, porém apresenta um custo computacional elevado. Além disso, foram propostas duas estratégias de fornecimento de soluções viáveis durante o algoritmo. A ideia de criar tais estratégias surgiu ao observarmos uma demora muito grande do algoritmo em obter soluções inteiras válidas. Uma das duas estratégias propostas é uma heurística capaz de transformar soluções relaxadas fracionárias em soluções viáveis inteiras. As estratégias são definidas como seguem:

i. Toda vez que uma solução relaxada inteira está violada, além de inserir a restrição no modelo, o algoritmo também insere a solução composta pela solução da líder relaxada e a da seguidora fornecida pela separação. Note que o tempo gasto por esta estratégia é negligível uma vez que tal solução sempre tem que ser avaliada pelo algoritmo de separação.

ii. Toda vez que uma solução relaxada fracionária não está violada, executamos uma heurística capaz de transformar esta solução em inteira da seguinte forma: No primeiro passo, a partir da solução da seguidora gerada pela separação, encontra-se a estratégia da líder que maximiza a sua demanda. Tal estratégia é obtida através de uma formulação essencialmente idêntica a (21) - (25), substituindo-se as constantes e variáveis do seguidor pelas da líder e vice-versa. No segundo passo, encontramos uma nova solução da seguidora executando o modelo (21) - (25) com base na solução da líder do passo anterior. As estratégias da líder e seguidora encontradas nos passos descritos previamente formam uma solução viável para o problema, que é fornecida ao algoritmo em caso de melhora. Note que o tempo computacional desta estratégia está associado à resolução dos dois modelos PLI descritos nos dois passos. Este tempo está contabilizado juntamente com o tempo gasto nas separações.

4. Experimentos Computacionais

Foram realizados experimentos computacionais utilizando-se CPLEX 12.5 num computador PC Intel Core I7 de 3.40 GHz, com 12Gb de RAM, na plataforma Linux, utilizando-se sempre um processador apenas. Após testarmos diversos valores para ε, identificamos $\varepsilon = 0{,}1$ como o melhor independentemente do tamanho da instância. O método foi testado em instâncias geradas aleatoriamente conforme [Plastria e Vanhaverbeke 2008], [Roboredo 2012] e [Gentile et al. 2014]. As instâncias são formadas de um grid quadrado, onde cada ponto com coordenadas inteiras positivas, é um consumidor e uma possível localidade para a líder ou para a seguidora. As localidades cuja soma das coordenadas é um múltiplo de três são possíveis localidades para a firma seguidora (conjunto F), ao passo que as demais são possíveis localidades para a líder (conjunto L). Deste modo, não existem localidades que possam ser ocupadas tanto pela líder como pela seguidora ($L \cap F = \emptyset$). O número de possíveis localidades para a líder é aproximadamente o dobro de possíveis localidades para a seguidora. Todas as distâncias são consideradas euclidianas. Os custos para instalação de uma facilidade i tanto pela líder como pela seguidora (f_i) são valores inteiros gerados aleatoriamente e distribuídos uniformemente no intervalo [5, 10]. A demanda de cada consumidor (w_j) também é inteira e gerada aleatoriamente e distribuída uniformemente no intervalo [50, 250]. As instâncias foram testadas para valores de orçamento $B^L = B^F = 15$ e $B^L = B^F = 25$. O raio de cobertura (δ_j) foi considerado o mesmo para todos os consumidores e testado para os valores 1, $\sqrt{2}$, 2, $\sqrt{5}$, $\sqrt{8}$, 3, $\sqrt{10}$, $\sqrt{13}$ e 4. Os tamanhos dos grids testados foram: 5x5, 7x7, 10x10, 12x12, 15x15 e 17x17. Todas as instâncias testadas apresentaram solução ótima dentro do limite máximo de 3h de execução.

Inicialmente comparamos o nosso algoritmo com um de força bruta, também por nós implementado. O algoritmo de força bruta enumera todas as soluções possíveis, retornando a ótima da seguinte forma: inicialmente o algoritmo constrói todas as combinações viáveis para uma estratégia com uma facilidade, depois para duas facilidades e finalmente para três facilidades (máximo possível dado o orçamento $B^F = 15$), tanto para a líder como para a seguidora. Para o orçamento $B^L = B^F = 25$, o algoritmo também constrói estratégias com quatro e até cinco facilidades, o que leva a uma explosão combinatória. Os tempos comparativos estão apresentados na Tabela 1, onde para cada grid são mostrados os tempos médios de execução considerando todas as possibilidades de raio, tendo em vista que este praticamente não influencia no tempo de execução do algoritmo de força bruta.

Tabela 1 – Algoritmo de Força Bruta

	$B^L = B^F = 15$		$B^L = B^F = 25$	
Grid	Tempo médio Força Bruta (s)	Tempo médio Modelo Proposto(s)	Tempo médio Força Bruta (s)	Tempo médio Modelo Proposto(s)
5x5	0,02	0,56	3,07	0,60
7x7	0,11	3,49	73,64	17,39
10x10	17,70	49,44	84.062,20	289,40
12x12	163,72	185,71	>500.000,00	813,69
15x15	3.515,24	405,67	-	2.689,09

Observando a Tabela 1, podemos notar que os tempos obtidos pelo modelo proposto crescem de maneira mais suave quando comparados aos obtidos pelo algoritmo força bruta a medida que o tamanho de grid aumenta. Além disso, a tabela mostra que, para orçamento $B^L = B^F = 25$, o modelo proposto supera com facilidade o algoritmo força bruta.

Agora apresentamos estatísticas detalhadas referente ao método proposto. A Tabela 2 apresenta os resultados para instâncias com $B^L = B^F = 15$, enquanto a Tabela 3 apresenta os resultados para $B^L = B^F = 25$. Para cada instância testada foi contabilizada a demanda total, a demanda atendida pela líder, a demanda atendida pela seguidora, o *gap* e o tempo na raiz, o número de nós na árvore *branch-and-bound*, o número de separações fracionárias, o número total de separações, o número de cortes fracionários, o número total de cortes, o número de soluções calculadas pela heurística, o número de soluções inseridas no modelo pela heurística, o tempo gasto nas separações e o tempo total de execução. Todos os tempos considerados são tempos de CPU em segundos.

Fruto da observação das Tabelas 2 e 3, algumas conclusões foram tiradas. Primeiramente, observamos que o método foi capaz de resolver todas as instâncias em razoáveis tempos computacionais. A instância mais difícil foi a com grid 15x15, $\delta_j = \sqrt{10}$ e orçamentos $B^L = B^F = 25$, que demorou um pouco mais de 2h para a resolução. Outro ponto é que o *gap* da raiz foi baixo para a maior parte das instâncias. Por outro lado, algumas instâncias apresentaram um elevado valor neste quesito. Isto indica o tempo computacional final poderia ser melhorado investindo na melhoria da formulação. Outra observação importante é o baixo tempo gasto acumulado com as separações, que foi pequeno em relação ao tempo total para a maioria das instâncias. Apesar disso, não notamos melhora no tempo total quando tentamos resolver mais separações diminuindo o valor de ε. O número de nós também foi pequeno para a maioria das instâncias. Para as instâncias com $B^L = B^F = 15$, por exemplo, todas as instâncias foram resolvidas com menos de 1000 nós. O baixo número de nós deve-se, entre outros fatores, ao elevado número de cortes associados a soluções fracionárias inseridos. Uma última observação diz respeito as estratégias de fornecimento de soluções propostas. Apesar do baixo número de soluções inseridas, estas foram essenciais para a aceleração do método. Diversas instâncias não eram resolvidas em 3h antes de aplicarmos tais estratégias.

5. Conclusão

Neste artigo propusemos pela primeira vez um método exato para o Problema de Localização Competitiva com Máxima Cobertura. O método proposto consistiu de um algoritmo branch-and-cut aplicado a um modelo PLI.

Para acelerar o método proposto, foram propostas duas estratégias de fornecimento de soluções viáveis ao algoritmo, onde uma delas consistia de uma heurística capaz de transformar soluções relaxadas fracionárias em uma solução viável inteira completa para o problema.

Com o intuito de mostrar a robustez do método proposto, diversos resultados computacionais foram apresentados, onde o método mostrou-se capaz de resolver instâncias significativamente grandes em razoáveis tempos computacionais.

Em trabalhos futuros, pretendemos criar cortes válidos para a formulação, o que poderia diminuir os *gaps* obtidos no nó raiz e, consequentemente, também diminuir o tempo computacional final. Além disso, temos a intenção de testar a formulação proposta nas instâncias utilizadas por [Seyhan 2012].

6. Referências

Bard, J. F., Moore, J. T. (1992), An algorithm for the discrete bilevel programming problem. *Naval Research Logistics (NRL)* v. 39, n. 3, p. 419-435.

Church, R., Velle, C. R. (1974), The maximal covering location problem. *Papers in regional science*, 32(1), 101-118.

Eiselt, H., Laporte, G. (1997), Sequential location problems. *European Journal of Operational Research*, v. 96, n. 3, p. 217-231.

Eiselt, H., Laporte, G., Thisse, J.(1993), Competitive location models: A framework and bibliography. *Tansportation Science*, v. 27, n.1, p. 44-54.

Friesz, T.; Miller, T.; Tobin, R. (1988), Competitive network facility location models: a survey. *Papers in Regional Science*, v. 65, n. 1, p. 47-57.

Gentile, J.; Pessoa, A. A.; Roboredo, M. C. (2014) Um algoritmo branch-and-cut para o problema da cobertura máxima competitiva minimizando o maior arrependimento. *Anais do XLVI Simpósio Brasileiro de Pesquisa Operacional* (SBPO).

Israeli, E. e Wood, R. K. (2002), Shortest-path network interdiction. *Networks*, v. 40, p. 97–111.

Kress, D., Pesch, E. (2012), Sequential competitive location on networks. *European Journal of Operational Research*, v. 217, n. 3, p. 483–499.

Moore, J. e Bard, J. (1990), The mixed integer linear bilevel programming problem. *Operations Research*, v. 38, p. 911-921.

O'Hanley, J. R. e Church, R. L. (2011), Designing robust coverage networks to hedge against worst-case facility losses. *European Journal of Operational Research*, v. 209, p. 23-36.

Pessoa, A. A., Poss, M., Roboredo, M. C., Aizemberg, L. (2013), Solving bilevel combinatorial optimization as bilinear min-max optimization via a branch-and-cut algorithm. *Anais do XLV Simpósio Brasileiro de Pesquisa Operacional* (SBPO).

Plastria, F. e Vanhaverbeke, L. (2008), Discrete models for competitive location with foresight. *Computers & Operations Research*, v. 35, n. 3, p. 683-700.

Roboredo, M. C. (2012), A branch-and-cut algorithm for a budget constrained centroid problem. *Anais do XLIV Simpósio Brasileiro de Pesquisa* Operacional (SBPO).

Roboredo, M. C. e Pessoa, A. A. (2013), A branch-and-cut algorithm for the discrete (r|p)-centroid problem. *European Journal of Operational Research*, v. 224, n. 1, p. 101-109.

Serra D, ReVelle C. (1994), Market capture by two competitors: the preemptive location problem. *Journal of Regional Science,* v. 34, p. 549–561.

Seyhan, Tolga Han, (2012) "Network Design Under Competition". Theses and Dissertations. Disponível em: http://preserve.lehigh.edu/cgi/viewcontent.cgi?article=2266&context=etd.

Taskin, Z. C., Smith, J. C., Ahmed, S., Schaefer, A. J. (2009), Cutting plane algorithms for solving a stochastic edge-partition problem. *Discrete Optimization*, v. 6, p. 420-435.

Tabela 2 : Resultados com $B^L = B^F = 15$

grid	δ	$\sum w_j$	Opt líder	Opt seguidor	Gap raiz (%)	tempo raiz (s)	# nós	# sep frac	# sep total	# cut frac	# cut total	# sols calc	# sols inser	tempo sep	tempo total
5x5	1	3542	1380	1221	9,15	0,98	9	14	15	2	3	13	1	0,39	1,00
5x5	√2	3542	1938	1604	4,95	0,3	5	1	2	1	2	1	1	0,02	0,32
5x5	2	3542	1938	1604	1,77	0,59	3	7	8	6	7	2	2	0,12	0,59
5x5	√5	3542	1938	1604	0,00	0,47	0	7	8	3	4	5	3	0,17	0,47
5x5	√8	3542	1938	1604	0,00	0,42	0	6	7	5	6	2	2	0,09	0,42
5x5	3	3542	1938	1604	0,00	0,40	0	6	7	5	6	2	2	0,10	0,40
5x5	√10	3542	1938	1604	0,00	0,55	0	10	11	8	9	3	3	0,20	0,55
5x5	√13	3542	1938	1604	0,00	0,40	0	7	8	6	7	2	2	0,12	0,40
5x5	4	3542	1938	1604	2,34	0,92	2	14	15	7	8	8	3	0,34	0,91
7x7	1	6531	1227	1508	12,27	3,46	21	17	19	2	3	16	1	1,78	4,06
7x7	√2	6531	2291	2536	4,25	1,42	5	1	2	0	1	2	1	0,17	1,49
7x7	2	6531	2475	2761	19,33	3,08	78	34	34	7	7	27	2	5,32	8,31
7x7	√5	6531	3091	3440	6,51	3,53	54	10	13	2	3	9	2	1,67	4,38
7x7	√8	6531	3160	3371	6,11	2,37	21	7	7	5	5	2	2	0,49	2,76
7x7	3	6531	3160	3371	6,56	1,97	55	4	7	3	5	3	2	0,43	2,91
7x7	√10	6531	3170	3361	0,00	1,64	0	4	4	3	3	1	1	0,23	1,64
7x7	√13	6531	3170	3361	1,61	2,39	1	6	7	5	6	2	2	0,36	2,40
7x7	4	6531	3170	3361	0,00	3,46	0	5	6	4	5	2	2	0,45	3,46
10x10	1	15247	2402	2705	0,00	3,37	0	1	2	0	1	2	1	0,52	3,37
10x10	√2	15247	4553	4617	0,00	2,44	0	1	2	1	1	0	0	0,21	2,44
10x10	2	15247	4993	5617	9,31	12,13	68	14	14	5	5	9	3	4,70	16,38
10x10	√5	15247	5910	7386	13,01	11,39	147	63	65	6	6	57	2	36,61	56,36
10x10	√8	15247	5942	8454	17,60	27,49	212	68	71	6	6	62	2	47,40	81,88
10x10	3	15247	6385	7818	10,97	19,96	72	27	29	4	5	24	3	15,18	35,44
10x10	√10	15247	6386	8564	10,70	21,17	197	44	47	5	5	39	3	30,00	63,27
10x10	√13	15247	6425	8822	7,72	23,5	128	74	75	6	6	68	2	67,10	100,82
10x10	4	15247	6290	8957	8,22	27,89	162	68	69	8	8	60	4	41,74	84,99
12x12	1	22977	2853	2551	0,00	9,26	0	1	3	1	2	1	0	0,68	9,26
12x12	√2	22977	4824	4578	0,00	8,42	0	1	2	0	1	2	1	1,27	8,42
12x12	2	22977	6108	6091	6,69	31,94	78	2	3	1	2	2	1	1,56	39,66
12x12	√5	22977	8685	7354	9,68	29,16	96	3	4	2	3	2	1	2,25	35,78
12x12	√8	22977	8967	9387	16,67	71,65	432	95	97	5	5	90	2	192,66	298,06
12x12	3	22977	9976	8360	11,86	50,28	360	53	55	5	6	49	3	95,91	157,30
12x12	√10	22977	10621	9753	12,82	58,18	600	199	201	7	7	192	1	520,00	660,52
12x12	√13	22977	11021	10390	15,73	134,71	538	75	75	7	7	68	1	202,25	340,33
12x12	4	22977	11381	10555	9,78	61,02	381	8	9	2	3	7	2	20,01	122,06
15x15	1	31354	2770	2313	2,04	55,98	71	0	2	0	1	1	1	3,18	69,42
15x15	√2	31354	4765	4209	0,00	18,46	0	0	2	0	1	1	0	1,82	18,46
15x15	2	31354	6408	5579	1,76	57,24	7	2	6	2	4	2	2	3,79	71,97
15x15	√5	31354	9487	8648	5,55	98,93	89	3	7	2	4	3	2	7,06	150,51
15x15	√8	31354	10833	9706	3,98	125,46	34	5	7	3	4	3	1	8,93	179,35
15x15	3	31354	11767	10647	7,96	104,18	269	14	20	4	8	14	5	42,86	246,14
15x15	√10	31354	13146	12690	10,72	186,49	839	73	83	17	22	61	4	278,97	1.101,06
15x15	√13	31354	14536	13085	8,44	195	241	72	75	11	13	63	2	278,43	753,06
15x15	4	31354	14958	13661	7,80	192,65	308	130	134	13	16	120	1	467,27	1.061,06
17x17	1	43429	3005	2315	0,25	150,49	9	0	2	0	1	1	1	3,15	245,04
17x17	√2	43429	4912	4441	3,93	204,11	52	4	9	3	6	4	2	23,26	257,17
17x17	2	43429	6775	6128	0,00	87,07	0	1	2	1	1	0	0	2,49	87,07
17x17	√5	43429	10538	9585	1,74	287,97	12	3	5	2	3	2	1	18,39	381,95
17x17	√8	43429	11960	10796	5,48	238,88	319	2	13	1	10	10	1	28,45	677,71
17x17	3	43429	13588	12142	3,82	339,16	215	4	12	3	10	8	2	18,48	610,20
17x17	√10	43429	16528	14760	6,00	254,36	410	10	22	3	13	17	4	53,44	935,90
17x17	√13	43429	18867	17084	8,57	482,99	337	32	44	10	20	32	4	146,49	1.395,52
17x17	4	43429	19266	17293	12,27	591,91	446	38	46	20	26	24	4	158,00	1.368,63

Tabela 3 : Resultados com $B^L = B^F = 25$

grid	δ	$\sum w_j$	Opt líder	Opt seguidor	Gap raiz (%)	tempo raiz (s)	# nós	# sep frac	# sep total	# cut frac	# cut total	# sols calc	# sols inser	tempo sep	tempo total
5x5	1	3542	1537	2005	10,69	0,49	36	21	25	5	7	18	1	0,53	0,95
5x5	√2	3542	1432	2110	7,35	0,26	45	3	6	3	6	3	2	0,05	0,42
5x5	2	3542	1428	2114	5,80	0,58	9	15	16	6	7	10	4	0,42	0,73
5x5	√5	3542	1428	2114	5,02	0,36	25	5	7	3	5	4	3	0,11	0,49
5x5	√8	3542	1428	2114	5,64	0,38	11	7	8	6	7	2	2	0,12	0,40
5x5	3	3542	1428	2114	6,30	0,36	55	6	8	5	7	3	2	0,11	0,51
5x5	√10	3542	1428	2114	6,30	0,59	76	10	12	5	7	7	4	0,33	0,82
5x5	√13	3542	1428	2114	6,12	0,36	50	6	8	5	7	3	2	0,11	0,52
5x5	4	3542	1428	2114	5,79	0,5	15	9	10	8	9	2	2	0,15	0,56
7x7	1	6531	1722	2602	10,28	11,25	36	35	37	0	1	36	1	8,93	14,57
7x7	√2	6531	2259	3590	18,71	6,36	493	186	189	4	5	183	2	36,82	45,51
7x7	2	6531	2398	3980	16,65	4,56	418	179	182	2	3	178	2	36,36	43,01
7x7	√5	6531	2462	4069	6,89	2,2	318	4	9	2	5	5	4	0,60	6,51
7x7	√8	6531	2462	4069	6,28	3,79	55	4	8	3	5	3	2	0,69	5,53
7x7	3	6531	2462	4069	6,97	2,55	67	5	10	2	4	5	4	0,97	4,40
7x7	√10	6531	2462	4069	6,69	5,3	28	24	27	5	6	20	4	6,48	9,69
7x7	√13	6531	2462	4069	5,67	7,62	31	26	28	7	8	20	4	7,03	10,61
7x7	4	6531	2462	4069	6,98	5,96	67	52	54	9	10	44	4	12,74	16,67
10x10	1	15247	3612	3870	5,45	6,94	144	2	2	1	1	1	1	0,66	14,01
10x10	√2	15247	5598	6651	13,24	15,65	1077	31	31	4	4	27	1	15,36	63,50
10x10	2	15247	6580	7287	10,22	20,58	823	18	21	4	6	16	2	23,16	77,96
10x10	√5	15247	6557	8460	15,41	34,19	5029	200	210	34	38	170	2	477,28	844,63
10x10	√8	15247	6568	8679	11,94	45,87	1503	75	82	19	24	61	4	200,77	379,64
10x10	3	15247	6568	8679	10,80	37,59	1600	84	90	19	23	69	6	194,68	438,27
10x10	√10	15247	6726	8521	5,46	39,67	360	50	51	24	24	26	3	100,70	210,26
10x10	√13	15247	6726	8521	4,94	42,47	310	137	139	34	35	104	4	328,70	407,55
10x10	4	15247	6726	8521	4,64	83,9	427	27	28	15	15	12	3	60,75	168,78
12x12	1	22977	4216	3887	4,57	15,51	107	5	7	1	2	5	1	5,43	20,77
12x12	√2	22977	7097	6007	8,63	44,73	1276	22	28	3	6	22	2	40,67	148,86
12x12	2	22977	9007	7695	9,22	95,45	1079	200	205	22	24	180	3	494,89	644,19
12x12	√5	22977	10664	9884	11,83	67,89	4294	200	206	27	29	175	3	695,89	1.272,46
12x12	√8	22977	11216	11105	8,42	68,74	1440	71	76	15	17	58	2	318,45	611,32
12x12	3	22977	11334	11083	8,08	56,1	1029	142	144	25	26	118	6	605,66	813,19
12x12	√10	22977	10937	11516	10,56	98,15	3112	136	141	45	47	93	3	698,38	1.578,44
12x12	√13	22977	11047	11878	5,30	87,26	1738	100	103	20	20	80	4	412,77	1.207,71
12x12	4	22977	11047	11878	5,08	109,49	985	86	89	26	26	60	4	469,40	1.026,31
15x15	1	31354	4382	3657	2,44	51,51	24	2	4	1	3	3	1	6,02	67,33
15x15	√2	31354	7358	5981	2,79	58,93	108	8	17	7	13	7	4	14,54	116,31
15x15	2	31354	10127	8729	2,51	81,73	214	2	9	1	6	6	3	9,07	154,66
15x15	√5	31354	13570	11635	8,01	247,68	3578	37	59	12	30	43	4	351,48	2.318,62
15x15	√8	31354	14668	12955	12,33	252,23	6362	63	74	26	34	45	5	614,19	3.961,52
15x15	3	31354	15412	13266	9,07	310,52	4634	55	74	28	45	44	4	805,07	4.048,15
15x15	√10	31354	15622	14868	9,58	367,89	8283	120	133	46	56	84	5	2.192,50	8.915,51
15x15	√13	31354	15869	15258	6,41	547,55	2035	42	47	23	27	23	3	750,76	2.574,81
15x15	4	31354	15990	15084	4,25	492,86	2038	13	17	12	14	3	1	167,46	2.044,93
17x17	1	43429	4934	3735	0,00	71,55	0	2	2	1	1	1	1	10,69	71,55
17x17	√2	43429	7826	6783	3,01	153,14	241	6	15	5	14	10	3	21,06	260,89
17x17	2	43429	10517	9496	2,84	181,45	394	10	21	6	15	13	3	39,42	386,10
17x17	√5	43429	16145	14331	4,49	293,99	532	9	14	8	13	6	3	30,57	653,33
17x17	√8	43429	18146	14665	8,53	491,78	5528	32	47	18	32	28	3	623,86	4.631,86
17x17	3	43429	19151	16242	10,25	448,42	2163	63	63	28	28	35	3	427,30	2.671,29
17x17	√10	43429	20661	19306	13,20	462,91	5145	142	152	59	66	90	5	701,55	7.787,07
17x17	√13	43429	21860	19570	10,80	616,92	5072	33	35	18	19	16	1	195,74	6.716,57
17x17	4	43429	21949	21022	9,40	514,67	4853	11	14	9	12	5	2	48,59	6.939,47