UM ALGORITMO *BRANCH-AND-CUT* PARA O PROBLEMA DA COBERTURA MÁXIMA COMPETITIVA MINIMIZANDO O MAIOR ARREPENDIMENTO

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

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

Marcos Costa Roboredo
Instituto de Matemática e Estatística – Universidade do Estado do Rio de Janeiro
Rua São Francisco Xavier 524, 20550-900, Maracanã, Rio de Janeiro, RJ, Brasil
marcos.roboredo@ime.uerj.br

RESUMO

O problema da cobertura máxima competitiva minimizando o maior arrependimento é formado por duas firmas, líder e seguidora, que competem entre si, em um determinado mercado, de modo a atender a demanda de clientes. A demanda de cada cliente é dita coberta caso exista uma facilidade localizada a uma distância máxima pré-especificada. A demanda de um cliente é atendida totalmente pela facilidade mais próxima da localizada pela líder ou pela seguidora que o cobre. 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. O problema consiste em decidir onde a líder localizará suas facilidades, de modo a minimizar o seu maior arrependimento. O problema é formulado em dois níveis e resolvido por um algoritmo *branch-and-cut*. Diversos experimentos computacionais são apresentados comprovando a eficiência e robustez do método.

PALAVRAS CHAVE. Programação Inteira, Cobertura Máxima Competitiva, Mínimo arrependimento.

Área Principal: Otimização Combinatória, Programação Matemática.

ABSTRACT

The competitive maximum coverage problem minimizing the greatest regret is composed of two firms, leader and follower, competing in a given market, to serve customers demand. The demand of each client is served if there is one facility located within a coverage radius. The demand of a client is fully served by the closest facility, placed by the leader or the follower. Competition is sequential. Initially the leader places its facilities, under a budget constraint. The follower will then react by doing the same. The problem consists of deciding where the leader will place its facilities, considering it aims to minimize its greatest regret. The problem is formulated as a bi-level one and solved by a branch-and-cut algorithm. Computational experiments are presented, proving the efficiency and robustness of the method.

KEYWORDS. Integer Programming, Competitive Maximum Coverage, Min regret.

Main area: Combinatorial Optimization, Mathematical Programming.

1. Introdução

Os problemas de localização de facilidades (PLF) são intensamente explorados devido à sua relevância tanto prática quanto teórica. Tais problemas consistem basicamente em decidir onde uma firma deve localizar as suas facilidades otimizando algum critério. Os PLF podem ser classificados em contínuos e discretos. Problemas contínuos são aqueles em que as facilidades podem ser localizadas em qualquer ponto da região considerada. Problemas discretos são aqueles em que a há um conjunto finito de potenciais localidades para receber as facilidades.

Os problemas de localização de facilidades competitiva (PLC) visam incorporar competição à decisão de localização. Neste contexto, tal decisão não será feita de forma independente, mas deverá levar em conta que já existem facilidades de outras firmas não cooperativas no mercado e que poderão vir a surgir novas facilidades localizadas por estas firmas. Os PLC podem ser classificados em sequenciais ou simultâneos. Nos PLC sequenciais as decisões tomadas pelas firmas são feitas de maneira sequencial enquanto nos PLC simultâneos tais decisões são tomadas simultaneamente. Diversos PLC discretos sequenciais são discutidos e revisados em Kress e Pesch (2012) e em Eiselt e Laporte (1997).

Neste trabalho, incorporamos competição ao clássico Problema da Cobertura Máxima (PCM) desenvolvido por Church e Velle (1974). Uma facilidade cobre a demanda de um cliente quando está localizada a no máximo uma distância pré-especificada deste cliente. O problema consiste em decidir onde uma firma deve localizar suas facilidades de modo a maximizar a demanda total coberta.

No problema deste trabalho, é considerado um mercado inexplorado, onde uma firma chamada líder deve decidir onde localizar suas facilidades sabendo que uma firma não cooperativa chamada seguidora entrará no mercado localizando suas facilidades. A demanda de cada cliente é atendida se existe alguma facilidade localizada pela líder ou pela seguidora a no máximo uma distância pré-especificada deste cliente. Caso exista mais de uma facilidade dentro deste 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. Um outro aspecto importante a ser mencionado diz respeito ao critério a ser utilizado para a tomada de decisão, o que referencia a teoria dos jogos. Neste trabalho, vamos considerar que a líder localiza suas facilidades visando minimizar o seu maior arrependimento. O arrependimento é dado pela diferença entre a demanda total coberta pela líder caso ela soubesse onde a seguidora vai localizar e a demanda total realmente coberta pela líder. Nós referenciamos este problema como o Problema da Cobertura Máxima Competitivo minimizando o Maior Arrependimento (PCMC-MA).

Plastria e Vanhaverbeke (2008) propuseram especificamente o PCMC-MA. Além deste critério, os autores propuseram mais dois: a maximização do pior caso (max-min) e o critério onde líder e seguidora fazem o que é melhor para si (Stackelberg). No critério de Stackelberg, assume-se uma decisão racional por parte da seguidora, o que faz sentido apenas caso toda a decisão da seguidora seja tomada de forma centralizada. Em caso contrário, uma alternativa é assumir o pior caso para a líder (max-min), o que pode evitar cenários desastrosos mas, por outro lado, pode reduzir muito os ganhos em cenários não tão pessimistas. Um critério que pode ser considerado mais equilibrado é o mínimo arrependimento, porque pode privilegiar tanto cenários otimistas quanto pessimistas. Plastria e Vanhaverbeke propuseram para os três casos um Modelo de Programação Linear Inteira (PLI) considerando o caso particular onde a seguidora pode localizar apenas uma facilidade. Roboredo (2012) considerou o caso geral do problema para o critério do pior caso onde um algoritmo *branch-and-cut* foi apresentado.

O PCMC-MA é um problema de otimização em dois níveis inteiros (2-PONI), onde o problema de otimização de primeiro nível consiste em escolher as localidades onde a líder localizará suas facilidades, minimizando o seu maior arrependimento, enquanto o problema do segundo nível consiste em se determinar onde a seguidora localizará suas facilidades, visando maximizar o arrependimento da líder. Os 2-PONI, em geral, são problemas difíceis de resolver de maneira exata. Moore e Bard (1990) propuseram um algoritmo *branch-and-bound* capaz de resolver instâncias relativamente pequenas. Em geral os procedimentos exatos para 2-PONI são gerados a partir de características específicas dos problemas, como por exemplo métodos de decomposição baseados em desigualdades superválidas (Israeli e Wood, 2002; O'Hanley e Church, 2011), algoritmos de plano de corte (Taskin *et al*.,2009), dentre outros. Mais recentemente, alguns trabalhos vem mostrando que, para a classe de problemas minimax bilineares, a formulação de Programação Linear Inteira Mista (PLIM) obtida pela substituição da maximização mais interna por um conjunto de desigualdades lineares pode ser resolvida de maneira eficiente através de um algoritmo de *branch-and-cut.* Esta ideia já foi aplicada ao problema da mediana com fortificação e interdições (Pessoa *et al*.,(2013)), ao problema (r,p)-centróide (Roboredo e Pessoa, 2013) e ao PCMC, considerando o critério do pior caso, anteriormente mencionado (Roboredo, 2012). Para os três casos, o método conseguiu resolver instâncias em aberto e obteve resultados satisfatórios quando comparado a outros métodos exatos.

Neste trabalho, mostramos inicialmente que o problema admite uma formulação minimax bilinear e, em seguida, derivamos um algoritmo *branch-and-cut* para o mesmo, substituindo o problema de otimização do 2º nível por um conjunto de desigualdades lineares. No melhor do nosso conhecimento, pela primeira vez está sendo proposto um modelo exato pata o PCMC-MA, no caso geral.

O artigo está dividido da seguinte maneira. A seção 2 apresenta a definição do problema, sendo em seguida apresentado um exemplo ilustrativo. A seção 3 mostra a modelagem do problema por meio de um algoritmo de branch-and-cut. A seção 4 apresenta resultados de experimentos computacionais realizados.

2. O problema

Considere um mercado inexplorado composto por um conjunto $J = \{1,...,n\}$ de clientes, onde cada cliente $j \in J$ possui uma demanda w_j. Com o intuito de atender a demanda destes clientes, duas firmas não cooperativas chamadas, uma de líder e outra de seguidora, entrarão neste mercado localizando um conjunto de facilidades de modo seqüencial. Sejam L e F os conjuntos das potenciais facilidades para a líder e para a seguidora respectivamente. Para cada cliente $j \in J$ 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 Bl e Bf 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 cliente $j \in J$ somente será atendida caso exista uma facilidade da líder ou da seguidora dentro do raio de cobertura do cliente 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 cobertura máxima competitiva minimizando o maior arrependimento da líder (PCMC-MA) consiste em decidir onde a líder deve localizar suas facilidades, de modo a minimizar o seu maior arrependimento. O arrependimento é calculado pela diferença entre a demanda atendida por esta firma, se ela soubesse qual decisão a seguidora iria tomar e a demanda que esta firma realmente atende com a decisão escolhida. Considera-se que a decisão da seguidora será tomada visando maximizar esta diferença.

2.1 Exemplo ilustrativo

A fim de se compreender melhor o que representa minimizar o arrependimento da líder, considere o seguinte exemplo, com 12 clientes ($J=\{J_1,J_2,...,J_{12}\}$, 2 potenciais facilidades para a líder ($L=\{L_1,L_2\}$) e 3 potenciais facilidades para a seguidora ($F=\{F_1,F_2,F_3\}$). Os custos fixos f_i

para se localizar uma facilidade, os orçamentos B_l e B_f e a demanda w_j de cada cliente são todos considerados unitários. Deste modo, devido a seus orçamentos, líder e seguidora poderão localizar no máximo uma facilidade. A Tabela 1 apresenta as demandas atendidas pela líder para cada combinação entre uma estratégia da líder e uma da seguidora. Os valores foram arbitrariamente atribuídos.

Tabela 1 : Demanda atendida

		Seguidora		
		F_1	F_2	F_3
Líder	L_1	5	4	4
	L_2	3	5	5

Pode-se então derivar uma segunda tabela, que representa o arrependimento que a líder teria, levando-se em conta a decisão da firma seguidora. A Tabela 2 é montada da seguinte forma. Suponhamos que a líder escolha a estratégia L_2 e a seguidora a estratégia F_1. A demanda atendida pela líder na Tabela 1 é 3. Porém se a líder soubesse que a seguidora escolheria F_1, ela teria escolhido L_1 e atenderia uma demanda de 5. A líder atendeu 3, mas poderia ter atendido 5. O arrependimento é a diferença (5-3=2), que é colocado na Tabela 2. Na situação da líder escolher a estratégia L_1 e a seguidora F_1, a líder já teria escolhido a melhor possibilidade. O arrependimento, neste caso, é igual a 0. Deste modo, monta-se a tabela, com todas as combinações possíveis de estratégia para a líder e para a seguidora.

O objetivo da líder, de acordo com o PCMC-MA é minimizar o seu maior arrependimento. Para a estratégia L_1, o maior arrependimento é 1 (seguidor adota F_2 ou F_3). Para a estratégia L_2, o maior arrependimento é 2 (seguidor adota F_1). Logo a líder deve escolher L_1, ficando com um arrependimento de no máximo 1.

Tabela 2: Arrependimento

		Seguidora		
		F_1	F_2	F_3
Líder	L_1	0	1	1
	L_2	2	0	0

3. Modelo para o PCMC-MA

Nesta seção é mostrado que o PCMC-MA pode ser formulado como um problema min-max bilinear. Para isto, seja S o conjunto de todas as possíveis estratégias da seguidora. Assim, cada $S_0 \in S$ é um subconjunto de facilidades de F, tal que $\sum_{i \in S_0} f_i \leq B_f$. Para cada $S_0 \in S$, seja $Z(S_0)$ a maior demanda possível que a líder conseguiria atender se soubesse que a estratégia escolhida pela seguidora seria S_0. Para efeito de facilitar a nomenclatura neste artigo, a estratégia da líder escolhida para se calcular $Z(S_0)$ será chamada de estratégia da líder virtual. O arrependimento é então calculado pela diferença entre $Z(S_0)$ e a demanda efetivamente atendida pela líder.

Considere agora as seguintes variáveis binárias:

- x_i - representa que a líder definiu a localidade i para ser ocupada;
- y_{ij} - É igual a 1 caso a localidade i tenha sido ocupada pela líder e seja a facilidade mais próxima do cliente j dentre as ocupadas pela líder, e igual a 0 caso contrário.

Uma formulação natural em dois níveis para o PCMC-MA segue abaixo:

$$\min_{x,\,y} \; Z(S_0) - \sum_{j \in J} \; \sum_{i \in L | d_{ij} < \min\{d_{kj} | k \in S_0\} \wedge d_{ij} \le \delta_j} w_j \; y_{ij} \tag{1}$$

$$s.a. \quad \sum_{i \in L} x_i \; f_i \le B_l \tag{2}$$

$$y_{ij} \le x_i \qquad \forall\, i \in L, \forall\; j \in J \tag{3}$$

$$\sum_{i \in L} y_{ij} = 1 \qquad \forall\; j \in J \tag{4}$$

$$y_{ij} \in \{0,1\} \qquad \forall\, i \in L, \forall\; j \in J \tag{5}$$

$$x_i \in \{0,1\} \qquad \forall\, i \in L \tag{6}$$

$$\max_{S_0} \; Z(S_0) - \sum_{j \in J} \; \sum_{i \in L | d_{ij} < \min\{d_{kj} | k \in S_0\} \wedge d_{ij} \le \delta_j} w_j \; y_{ij} \tag{7}$$

$$s.a. \quad \sum_{i \in S_0} f_i \le B_f \tag{8}$$

A função objetivo da líder (1) e da seguidora (7) são idênticas, porém opostas. Enquanto a líder deseja minimizar o arrependimento, a seguidora deseja maximizá-lo. A função objetivo possui dois termos: o primeiro refere-se à líder virtual, que representa a maior demanda possível de ser atendida pela líder com a estratégia S_0 definida pela seguidora. O segundo, negativo, representa a demanda real que é atendida pela líder e a diferença representa o arrependimento. A restrição (2) garante que a estratégia da líder respeita o seu orçamento. A restrição (3) garante a consistência entre as variáveis y e x. A restrição (4) assegura que para cada cliente j, haja exatamente uma facilidade mais perto de j localizada pela líder. A restrição (8), do segundo nível, assegura que o orçamento da seguidora está sendo respeitado.

Para se aplicar o método proposto, faz-se necessário linearizar o modelo (1) - (8). Para tanto, vamos propor a criação de algumas variáveis auxiliares para a descrição de S_0. As variáveis do primeiro nível são as 2 anteriormente descritas, x e y. Para o segundo nível foram criadas as seguintes variáveis binárias:

- s_k - representa que a seguidora definiu a localidade k para ser ocupada;
- t_{kj} - é igual a 1 caso a localidade k tenha sido ocupada pela seguidora e esta é a facilidade mais próxima do cliente j dentre as ocupadas pela seguidora, e igual a 0 caso contrário;
- x'_i - representa que a líder virtual definiu a localidade i para ser ocupada;
- y'_{ij} - é igual a 1 se a líder virtual atende o cliente j pela facilidade i; é igual a 0 caso contrário. Somente criamos esta variável se $d_{ij} \le \delta_j$;
- v_{kj} - é igual a 1 caso a localidade k não esteja ocupada pela seguidora, e nenhuma facilidade mais próxima do cliente j do que k esteja ocupada pela seguidora; é igual a 0 caso contrário.

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

A formulação em dois níveis linear para o PCMC-MA, fica da seguinte forma:

$$\min_{x,y} \sum_{j \in J} \; \sum_{i \in L | \, d_{ij} \le \delta_j} w_j \; y'_{ij} - \sum_{j \in J} \sum_{k \in F} \left(w_j \sum_{i \in L | d_{ij} < d_{kj} \wedge d_{ij} \le \delta_j} y_{ij} \right) t_{kj} \tag{9}$$

$$s.a. \quad \sum_{i \in L} x_i \; f_i \le B_l \tag{10}$$

$$y_{ij} \le x_i \qquad \forall\, i \in L, \forall\; j \in J \tag{11}$$

$$\sum_{i \in L} y_{ij} = 1 \qquad \forall\ j \in J \tag{12}$$

$$0 \le y_{ij} \le 1 \qquad \forall\ i \in L, \forall\ j \in J \tag{13}$$

$$x_i \in \{0,1\} \qquad \forall\ i \in L \tag{14}$$

$$\max_{s,t,x',y'} \sum_{j \in J} \sum_{i \in L \mid d_{ij} \le \delta_j} w_j\, y'_{ij} - \sum_{j \in J} \sum_{k \in F} \left(w_j \sum_{i \in L \mid d_{ij} < d_{kj} \wedge d_{ij} \le \delta_j} y_{ij} \right) t_{kj} \tag{15}$$

$$s.a. \quad \sum_{k \in F} f_k\, s_k \le B_f \tag{16}$$

$$\sum_{i \in L} f_i\, x'_i \le B_l \tag{17}$$

$$t_{kj} \le s_k \qquad \forall\ k \in F, \forall\ j \in J \tag{18}$$

$$y'_{ij} \le x'_i \qquad \forall\ j \in J, \forall\ i \in L | d_{ij} \le \delta_j \tag{19}$$

$$\sum_{k \in F} t_{kj} = 1 \qquad \forall\ j \in J \tag{20}$$

$$\sum_{i \in L | d_{ij} \le \delta_j} y'_{ij} \le 1 \qquad \forall\ j \in J \tag{21}$$

$$y'_{ij} + \sum_{k \in F | d_{kj} \le d_{ij}} t_{kj} \le 1 \qquad \forall\ j \in J, \forall\ i \in L\, | d_{ij} \le \delta_j \tag{22}$$

$$v_{\theta(1,j)j} = 1 - s_{\theta(1,j)} \qquad \forall\ j \in J \tag{23}$$

$$v_{kj} \le 1 - s_k \qquad \forall\ k \in F, \forall\ j \in J \tag{24}$$

$$v_{\theta(k,j)j} \ge v_{\theta(k+1,j)j} \qquad \forall\ k = 1,\ldots,F-1, \forall\ j \in J \tag{25}$$

$$t_{\theta(1,j)j} = 1 - v_{\theta(1,j)} \qquad \forall\ j \in J \tag{26}$$

$$t_{\theta(k,j)j} = v_{\theta(k-1,j)j} - v_{\theta(k,j)j} \qquad \forall\ k = 2,\ldots,F, \forall\ j \in J \tag{27}$$

$$x'_i, s_k, y'_{ij}, t_{kj}, v_{kj} \in \{0,1\} \qquad \forall\ i \in L, \forall\ k \in F, \forall\ j \in J \tag{28}$$

As restrições (10) a (12) são as anteriormente mencionadas. As variáveis y foram definidas como contínuas, de modo a que o CPLEX ramifique a árvore de *branch-and-bound* apenas nas variáveis x. Isto facilita a resolução, uma vez que os números de variáveis x é significativamente menor do que o número de variáveis y. Tal relaxação somente pode ser feita pois as variáveis y sempre resultam em valores binários quando as variáveis x são declaradas binárias.

No segundo nível as restrições (16) e (17) asseguram que as estratégias da seguidora e da líder virtual respeitam seus orçamentos. As restrições (18) e (19) garantem as consistências entre as variáveis t_{kj} e s_k e entre y'_{ij} e x'_i respectivamente. As restrições (20) asseguram que para cada cliente j, existe uma facilidade mais próxima localizada pela seguidora. As restrições (21) garantem que para cada cliente j, existe no máximo uma facilidade da líder virtual para servir o cliente. As restrições (22) asseguram o anteriormente mencionado, de modo que para que y'_{ij} seja 1, não exista nenhuma facilidade localizada pela seguidora mais próxima do cliente j do que a facilidade i localizada pela líder virtual. As restrições (23) e (24) asseguram a consistência entre as variáveis s e v. A restrição (23) garante que se a localidade mais próxima do cliente j foi ocupada pela seguidora, v_{kj} desta localidade seja igual a 0. Caso esta localidade mais próxima não tenha sido ocupada, v_{kj} seria igual a 1. A restrição (24) assegura que se uma localidade k foi ocupada pela seguidora, v_{kj} seja igual a 0 (se ela não foi ocupada, v_{kj} pode ser 0 ou 1). A restrição (25) garante que se v_{kj} referente a k-ésima localidade mais próxima do cliente j é 0, então todas as posteriores (de k+1 até F) também são 0. As restrições (26) e (27) asseguram as consistências entre as variáveis t_{kj} e v_{kj}. A restrição (26) assegura que se a localidade mais próxima do cliente j foi ocupada pela seguidora, t_{kj} é igual a 1 e v_{kj} é igual a 0. Caso contrário, t_{kj} é igual a 0 e v_{kj} é igual a 1. A restrição (27) garante que se a k-ésima localidade mais próxima do cliente j foi a ocupada mais próxima do cliente j, então $t_{\theta(k,j)j}$ é igual a 1 e para tal, $v_{\theta(k-1,j)j}$ é igual a 1 e $v_{\theta(k,j)j}$ é igual a 0.

Cabe aqui mencionar que as restrições (23) - (27) poderiam ser substituídas pelas restrições (29), com a eliminação da variável v_{kj} e da constante $\theta(k,j)$, no entanto, apesar da descrição bem mais simples que as anteriores, esta família de restrições corresponde a $O(|J|\,|F|^2)$ desigualdades. Testes realizados provaram que a utilização das restrições (23) - (27) apresentam melhores resultados que a utilização da restrição (29) para instâncias maiores.

$$t_{kj} + s_i \leq 1 \quad \forall\, j \in J, \forall\, k \in F, \forall\, i \in F \mid d_{ij} < d_{kj} \tag{29}$$

Seguindo Pessoa *et al.* (2013), substituímos o problema de otimização do segundo nível por um conjunto de desigualdades lineares:

$$\min_{x,y} z \tag{30}$$

$$\text{s. a.} \quad (10) - (14) \tag{31}$$

$$z \geq \sum_{j \in J} \sum_{i \in L \mid d_{ij} \leq \delta_j} w_j\, y'_{ij} - \sum_{j \in J} \sum_{k \in F} \left(w_j \sum_{i \in L \mid d_{ij} < d_{kj} \wedge d_{ij} \leq \delta_j} y_{ij} \right) t_{kj} \tag{32}$$

$$\forall\, s, t, x', y', v \quad \text{que satisfaz (16) - (28)}$$

O número de restrições (32) é dado pelo número de estratégias da seguidora. Como este número tende a ser exponencial, tais restrições devem ser geradas durante um algoritmo de *branch-and-cut*. O problema da separação consiste de, dada uma solução relaxada ($\bar{x}$, $\bar{y}$), encontrar a estratégia da seguidora (x', y', s, t, v) que maximiza o arrependimento da líder. Tal estratégia pode ser encontrada pela resolução do PLI (15) - (28) considerando os valores constantes $\bar{y}$ na função objetivo (15).

O modelo de PLI para o problema da separação somente foi resolvido para soluções relaxadas inteiras. Apesar dele também ser apto a separar soluções relaxadas fracionárias. Tal decisão foi tomada pela dificuldade em se resolver as separações.

4. Experimentos Computacionais

Foram realizados experimentos computacionais utilizando-se CPLEX 12.5 num computador PC AMD Phenom X4 9600 de 2.31 GHz, com 3Gb de RAM, utilizando-se sempre um processador apenas. O método foi testado em instâncias geradas aleatoriamente conforme Plastria e Vanhaverbeke (2008) e Roboredo e Pessoa (2012). As instâncias são formadas de um grid quadrado, onde cada ponto com coordenadas inteiras positivas, é um cliente e uma possível localidade para a líder ou para a seguidora. As localidades cuja soma das coordenadas seja múltipla 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 tanto pela líder como pela seguidora (f) são valores gerados aleatoriamente e distribuídos uniformemente no intervalo [5, 10]. A demanda de cada cliente (w_j) também é gerada aleatoriamente e distribuída uniformemente no intervalo [50, 250]. As instâncias foram testadas para valores de orçamento $B_l = B_f = 15$ e $B_l = B_f = 25$. A máxima distância viajada (δ_j) foi considerada a mesma para todos os clientes e testada para os valores 1, $\sqrt{2}$, 2, $\sqrt{5}$, $\sqrt{8}$, 3 e 4. O tamanho dos grids a serem testados foram: 5x5, 7x7, 10x10, 12x12 e 15x15. Todos os testes foram limitados em no máximo 6h de execução. Algumas instâncias não apresentaram solução ótima dentro deste tempo, sendo apresentada a melhor solução viável.

Para melhorar o tempo da solução do problema da separação, foi introduzido um *cutoff* no valor da solução relaxada, tendo em vista que a solução ótima do modelo PLI da separação é no mínimo igual à solução relaxada, ocasião em que nenhum corte é inserido, ou é superior a este valor, ocasião em que um corte é inserido no problema principal.

A Tabela 3 apresenta os resultados para instâncias com Bl = Bf = 15, enquanto a Tabela 4 apresenta os resultados para Bl = Bf = 25. Para cada instância testada foi contabilizada a demanda total, o mínimo arrependimento, a demanda atendida pela líder, o número total de separações, o número de cortes, o *gap* na raiz, o número de nós da árvore *branch-and-bound*, o tempo gasto nas separações, o tempo total para a solução ótima (em alguns casos viável), e nos casos de solução viável, foi incluído o *gap* desta solução em relação a melhor relaxação. Todos os tempos considerados são tempos de CPU em segundos. Nove instâncias com orçamento 25 não apresentaram solução ótima, sendo apresentada a melhor solução viável. Destas, em cinco delas o modelo não conseguiu sequer sair da raiz, porém em outras quatro, o modelo abriu a árvore de *branch-and-bound*. Para estas instâncias foi apresentado o *gap* raiz em relação a melhor solução ótima encontrada.

Foi observado que a máxima distância influencia consideravelmente no tempo para a solução. O algoritmo não conseguiu resolver a instância 10x10, orçamento 25 e máxima distância 4. Já a instância 15x15, com o mesmo orçamento e máxima distância 1 foi resolvida com facilidade. Foi observado que o número de separações e cortes em geral é pequeno. No entanto, mesmo com poucas separações, o algoritmo gasta muito tempo para resolver. Comparando-se o tempo total com o tempo da separação, conclui-se que quase a totalidade do tempo é gasto para a solução do modelo PLI da separação. Tal fato decorre da complexa modelagem que envolve a montagem destes problemas, do seu elevado número de restrições, bem como do fato de que a função objetivo da separação é uma diferença de dois termos, apresentando valores ótimos bem inferiores a demanda total, o que dificulta a convergência para esta solução. Há aqui oportunidade para a melhoria da modelagem do problema da separação.

Uma adaptação ao algoritmo foi efetuada de modo a se comparar os resultados do atual modelo com o modelo apresentado por Plastria e Vanhaverbeke (2008), que trata do PCMC-MA no caso particular do seguidor localizar apenas uma facilidade. Foi verificado que para instâncias menores (5x5 raios 2, 3 e 4, 7x7 raios 2, 3 e 4, 10x10 raio 2 e 12x12 raio 2) o modelo proposto por Plastria e Vanhaverbeke (2008) supera o nosso. Para as instâncias 10x10 raios 3 e 4, 12x12 raio 3 e 15x15 raio 2, o nosso modelo é comparável ao modelo acima mencionado. No entanto para instâncias maiores (12x12 raio 4 e 15x15 raios 3 e 4), o nosso modelo apresenta melhor desempenho que o apresentado por Plastria e Vanhaverbeke (2008) comprovando a robustez do método.

Foram construídos também algoritmos de força bruta, testando todas as soluções possíveis, de modo a se comparar com o modelo. O algoritmo constrói todas as combinações para uma estratégia com uma facilidade, depois para duas facilidades e finalmente para três facilidades. Algumas técnicas de modo a se descartar rapidamente soluções fora do orçamento e estratégias do seguidor que produzem arrependimento maior que o mínimo até o momento são utilizadas, de modo a se melhorar o seu desempenho. O algoritmo foi testado apenas com orçamento 15, tendo em vista que com orçamento 25, haveria a possibilidade de se localizar quatro ou até cinco facilidades, o que causaria uma explosão combinatória. Para o grid 5x5, o algoritmo levou em média menos de 0,5 s. Para as instâncias 7x7, o algoritmo levou cerca de 3s, para 10x10, 1.751s e para 12x12, mais de 7h, o que prova que para a maior parte das instâncias 10x10 e para instâncias de tamanho 12x12 em diante, o modelo proposto supera com facilidade a força bruta.

Uma outra possibilidade a ser explorada em futuras pesquisas é a criação de uma metaheurística para a separação de (32), por meio de, por exemplo, uma busca local iterada, ou de um algoritmo genético, de modo a se encontrar cortes violados sem a necessidade de solução de um modelo de PLI para a separação. Como a solução destes modelos exatos é bastante demorada, a utilização de metaheurísticas se mostra bastante promissora. Ainda que a heurística não encontrasse a solução ótima do problema da separação, bastaria que ela chegasse a um valor superior a solução relaxada do problema principal, para que um corte fosse inserido.

5. Conclusão

Neste trabalho foi apresentado um modelo de *branch-and-cut* para a solução do problema de cobertura máxima competitiva minimizando o maior arrependimento. Diversos testes computacionais foram realizados comprovando a robustez do método. Como pesquisa futura é intenção se aprofundar na modelagem do problema da separação, que ainda apresenta um tempo elevado para sua solução, bem como na pesquisa de heurísticas para se encontrar cortes violados sem a necessidade de se resolver o problema da separação por um método exato.

Tabela 3 : Resultado com $B_l = B_f = 15$

grid	δ	$\sum w_j$	mínimo arrepen-dimento	demanda líder	#sep	#corte	Gap raíz (%)	# nós	tempo sep	tempo total	Gap (%)
5x5	1	3542	19	1217	3	2	0,00	0	5,67	5,75	0,00
5x5	√2	3542	151	2385	4	3	0,00	0	7,33	7,50	0,00
5x5	2	3542	187	2064	7	5	0,00	0	18,81	19,13	0,00
5x5	√5	3542	371	1961	6	4	0,00	0	14,94	15,25	0,00
5x5	√8	3542	371	1961	6	4	0,00	0	11,17	11,61	0,00
5x5	3	3542	371	1961	6	4	0,00	0	13,16	13,48	0,00
5x5	4	3542	442	2812	6	4	0,00	0	8,56	8,89	0,00
7x7	1	6531	353	952	9	7	16,79	3	21,45	22,34	0,00
7x7	√2	6531	770	1267	12	8	56,09	75	27,78	30,69	0,00
7x7	2	6531	711	2344	9	6	21,68	9	51,91	53,22	0,00
7x7	√5	6531	918	2157	8	5	48,33	17	54,41	55,78	0,00
7x7	√8	6531	1010	2369	6	5	48,53	23	41,66	43,34	0,00
7x7	3	6531	884	3192	6	5	45,17	19	45,47	47,36	0,00
7x7	4	6531	855	3311	8	6	28,17	2	77,64	78,61	0,00
10x10	1	15247	596	1747	13	10	35,61	24	133,20	140,92	0,00
10x10	√2	15247	972	4110	9	6	0,00	0	89,03	93,52	0,00
10x10	2	15247	1121	5003	9	6	18,50	4	149,25	155,42	0,00
10x10	√5	15247	1872	5867	13	8	40,81	30	369,75	384,70	0,00
10x10	√8	15247	1899	5159	19	11	27,47	42	616,75	631,44	0,00
10x10	3	15247	1879	7362	11	8	41,67	44	356,06	371,39	0,00
10x10	4	15247	1741	6690	8	5	44,08	23	589,06	601,09	0,00
12x12	1	22977	654	2188	28	22	43,31	65	788,95	816,66	0,00
12x12	√2	22977	1208	4066	18	14	37,78	90	536,31	578,80	0,00
12x12	2	22977	1406	5330	9	6	37,00	58	437,33	471,78	0,00
12x12	√5	22977	2413	7834	11	8	33,33	64	804,05	845,55	0,00
12x12	√8	22977	2678	9142	11	7	31,82	27	844,73	880,00	0,00
12x12	3	22977	2568	10133	12	7	22,99	22	1.304,91	1.342,77	0,00
12x12	4	22977	2320	13932	9	6	30,80	16	2.028,89	2.060,19	0,00
15x15	1	31354	503	2222	10	8	40,39	17	1.226,48	1.282,70	0,00
15x15	√2	31354	1122	3586	10	8	31,69	75	1.362,98	1.521,41	0,00
15x15	2	31354	1487	4802	21	18	72,91	100	2.827,78	3.022,14	0,00
15x15	√5	31354	2339	7820	12	9	40,89	44	2.869,45	3.004,81	0,00
15x15	√8	31354	2654	10180	11	9	37,93	76	2.447,50	2.645,83	0,00
15x15	3	31354	2987	9061	10	8	77,73	89	2.876,78	3.053,31	0,00
15x15	4	31354	4159	13503	13	9	81,52	168	7.844,95	8.213,67	0,00

Tabela 4 : Resultado com $B_l = B_f = 25$

grid	δ	$\sum w_j$	mínimo arrepen- dimento	demanda líder	#sep	#corte	Gap raíz (%)	# nós	tempo sep	tempo total	Gap (%)
5x5	1	3542	462	1161	14	11	40,21	40	23,09	23,98	0,00
5x5	√2	3542	326	1547	7	6	33,05	15	12,11	12,61	0,00
5x5	2	3542	441	2106	20	14	37,87	49	36,08	37,38	0,00
5x5	√5	3542	597	2346	18	13	33,15	77	19,39	21,09	0,00
5x5	√8	3542	597	2346	19	14	30,29	99	22,28	24,06	0,00
5x5	3	3542	597	2346	19	14	33,12	70	18,36	19,80	0,00
5x5	4	3542	653	2838	22	18	26,85	94	23,83	25,72	0,00
7x7	1	6531	525	1687	12	10	42,11	110	44,05	47,70	0,00
7x7	√2	6531	1137	2228	13	11	45,70	144	81,03	85,66	0,00
7x7	2	6531	605	3021	9	7	51,22	72	99,88	102,78	0,00
7x7	√5	6531	778	2432	11	9	31,64	41	178,88	180,94	0,00
7x7	√8	6531	984	2432	21	15	31,89	49	276,30	279,47	0,00
7x7	3	6531	830	3371	10	9	42,48	97	145,78	149,47	0,00
7x7	4	6531	1047	3570	20	16	42,57	189	266,14	273,67	0,00
10x10	1	15247	1082	2616	28	22	45,81	381	399,00	437,17	0,00
10x10	√2	15247	1315	4334	7	5	35,72	135	150,69	164,52	0,00
10x10	2	15247	1460	5458	11	7	29,10	104	446,14	461,30	0,00
10x10	√5	15247	2231	6365	24	18	43,78	629	8.902,14	8.969,22	0,00
10x10	√8	15247	2449	7745	32	27	50,31	580	10.880,23	10.952,13	0,00
10x10	3	15247	<2253	7066	>25	>21	51,30	>499	>21.600,00	>21.600,00	20,60
10x10	4	15247	<2721	9032	>21	>19	-	0	>21.600,00	>21.600,00	60,51
12x12	1	22977	1037	3393	31	25	37,79	461	708,67	793,23	0,00
12x12	√2	22977	1871	5663	23	18	38,44	517	809,30	897,28	0,00
12x12	2	22977	1978	6737	29	24	34,66	199	2.958,53	3.016,73	0,00
12x12	√5	22977	<2576	11295	>16	>13	45,73	>19	>21.600,00	>21.600,00	42,96
12x12	√8	22977	<2775	9495	>9	>6	-	0	>21.600,00	>21.600,00	80,68
12x12	3	22977	<4859	9157	>6	>5	-	0	>21.600,00	>21.600,00	100,00
12x12	4	22977	<3815	11478	>8	>7	-	0	>21.600,00	>21.600,00	72,97
15x15	1	31354	863	3415	28	25	30,93	180	2.570,05	2.779,86	0,00
15x15	√2	31354	1843	5155	28	21	32,08	313	3.171,77	3.391,42	0,00
15x15	2	31354	2398	7729	20	18	27,75	213	2.383,34	2.598,52	0,00
15x15	√5	31354	3314	11623	20	18	47,74	860	6.893,59	7.503,50	0,00
15x15	√8	31354	<3803	11641	>24	>20	36,02	>154	>21.600,00	>21.600,00	19,15
15x15	3	31354	<3304	13166	>13	>11	49,07	>79	>21.600,00	>21.600,00	45,79
15x15	4	31354	<19431	4434	>5	>4	-	0	>21.600,00	>21.600,00	100,00

6. Referências

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.

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

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

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. (2001), Static competitive facility location: An overview of optimisation approaches, *European Journal of Operational Research*, v. 129, p. 461-470.

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.

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.