



ISSN 2175-6295 Rio de Janeiro- Brasil, 12 e 13 de agosto de 2010

UMA PROPOSTA DE DECOMPOSIÇÃO PARA O PROBLEMA DE LOCALIZAÇÃO DE MÁXIMA COBERTURA

Marcos Antonio Pereira

Edson Luiz França Senne

UNESP – Univ Estadual Paulista

Av. Dr. Ariberto Pereira da Cunha, 333

12516-410 – Guaratinguetá, SP – Brasil

E-mail: {mapereira, elfsenne}@feg.unesp.br

Luiz Antonio Nogueira Lorena

INPE – Instituto Nacional de Pesquisas Espaciais

Av. dos Astronautas, 1758

12201-970 – São José dos Campos, SP – Brasil

E-mail: lorena@lac.inpe.br

RESUMO

Este trabalho apresenta uma técnica de particionamento em *clusters* para o cálculo de limitantes superiores para a solução ótima de problemas de localização de máxima cobertura. A partir de uma distância de cobertura, obtém-se um grafo cujos vértices correspondem aos locais candidatos para a localização das facilidades e cujos os arcos conectam as facilidades capazes de atender um mesmo cliente. A formulação matemática correspondente permite identificar as restrições de acoplamento, que são relaxadas no sentido lagrangiano, resultando em subgrafos independentes que correspondem a subproblemas menores, mais fáceis de serem resolvidos por métodos exatos. A técnica proposta é comparada com a abordagem tradicional, usando dados de casos reais e disponíveis na literatura.

PALAVRAS-CHAVE. Problemas de cobertura. Relaxação lagrangiana. Decomposição em *clusters*.

ABSTRACT

This paper proposes a cluster partitioning technique to calculate improved upper bounds to the optimal solution of maximal covering location problems. Given a covering distance, a graph is built considering as vertices the potential facility locations, and with an edge connecting each pair of facilities that attend a same client. Coupling constraints, corresponding to some edges of this graph, are identified and relaxed in the Lagrangean way, resulting in disconnected subgraphs representing smaller subproblems that are computationally easier to solve by exact methods. The proposed technique is compared to the classical approach, using real data and instances from the available literature.

KEYWORDS. Covering problems. Lagrangean relaxation. Clustering.

1. Introdução

Em problemas de localização de facilidades, a classe de problemas de cobertura considera a distância máxima entre um cliente e a facilidade escolhida para o atendimento de sua demanda. O problema de cobertura de conjuntos (Toregas *et al*, 1971) determina o número mínimo de facilidades necessárias para o atendimento de todos os clientes, para uma dada distância de cobertura. A formulação matemática proposta para este problema não considera a demanda individual de cada cliente. Além disso, o número de facilidades necessárias pode ser grande, implicando em elevados custos fixos de instalação. Uma formulação alternativa considera a localização de um número limitado de facilidades, mesmo que esta quantidade não seja suficiente para o atendimento da demanda total. Nesta nova formulação relaxa-se a condição de atendimento de todos os clientes e busca-se localizar p facilidades para atender a maior parte da demanda para uma dada distância de cobertura. A formulação matemática assim obtida corresponde ao problema de localização de máxima cobertura (PLMC), apresentada por Church e ReVelle (1974).

Formulações de cobertura são freqüentemente encontradas em problemas da administração pública para a localização de serviços de emergência (Eaton *et al*, 1986; Current e O'Kelley, 1981), serviços hierárquicos de saúde (Moore e ReVelle, 1982), controle de poluição atmosférica (Houglund e Stephens, 1976) e sistemas congestionados (Marianov e Serra, 1998; 2001; Corrêa e Lorena, 2006). As primeiras técnicas de resolução do PLMC baseavam-se na obtenção de soluções inteiras a partir da solução de relaxação linear da formulação proposta por Church e ReVelle (1974), que também apresenta uma heurística baseada em troca de vértices. Foram também propostos métodos de solução baseados em heurísticas gulosas (Daskin, 1995) e relaxação lagrangiana (Galvão e ReVelle, 1996). Lorena e Pereira (2002) apresentam os resultados obtidos com a relaxação lagrangean/surrogate em uma heurística baseada em otimização de subgradientes, como alternativa às heurísticas lagrangiana e *surrogate* propostas em Galvão *et al* (2000). Arakaki e Lorena (2001) apresentam um algoritmo genético construtivo capaz de resolver casos reais com até 500 vértices. Outras técnicas de solução podem ser encontradas em Chung (1986), Hale e Moberg (2003), Galvão (2004) e Serra e Marianov (2004).

Neste trabalho apresenta-se uma técnica de decomposição baseada em *clusters* para a resolução de problemas de localização de máxima cobertura de grande porte. A abordagem proposta consiste na identificação de um subconjunto de restrições que, se relaxadas, permitem o particionamento do grafo do problema original em subgrafos (*clusters*) que correspondem a subproblemas menores que podem ser resolvidos de forma exata, independentemente.

O trabalho está organizado da seguinte forma. A seção 2 apresenta a técnica de decomposição para o cálculo melhorado de limitantes superiores para a solução ótima de problemas de localização de máxima cobertura. Na seção 3 discute-se os resultados computacionais obtidos com dados reais e disponíveis na literatura. As conclusões são apresentadas na seção 4.

2. Decomposição em grafo do PLMC

O PLMC foi formulado em Church e ReVelle (1974) como um problema linear 0-1:

$$(PLMC) \quad v(PLMC) = \text{Max} \sum_{i \in N} w_i x_i \quad (1)$$

$$\text{sujeito a:} \quad \sum_{j \in S_i} y_j \geq x_i, \quad \forall i \in N \quad (2)$$

$$\sum_{j \in M} y_j = p \quad (3)$$

$$x_i \in \{0, 1\}, \quad \forall i \in N \quad (4)$$

$$y_j \in \{0, 1\}, \quad \forall j \in M \quad (5)$$

onde:

- $M = \{1, 2, \dots, m\}$ é o conjunto dos locais candidatos para a instalação das facilidades;
- $N = \{1, 2, \dots, n\}$ é o conjunto dos clientes a serem atendidos;
- $D = [d_{ij}]$ é a matriz de distâncias euclidianas entre cada par de vértices $i \in N$ e $j \in M$.
- S é a distância de cobertura;
- $S_i = \{j \in M \mid d_{ij} \leq S\}$ é o conjunto das facilidades que podem atender o cliente $i \in N$;
- w_i é a demanda (positiva) de cada cliente $i \in N$;
- p é o número de facilidades a serem localizadas;
- x_i é uma variável de decisão, com $x_i = 1$ se a demanda do cliente i é atendida, e $x_i = 0$, c.c.;
- y_j é uma variável de decisão, com $y_j = 1$ se uma facilidade é instalada em j , e $y_j = 0$, c.c.

A função objetivo busca maximizar a demanda atendida. O conjunto de restrições (2) estabelecem que um cliente será atendido se existir pelo menos uma facilidade instalada a menos de U unidades de distância. A restrição (3) limita a p o número de facilidades a serem instaladas e (4) e (5) definem a natureza binária das variáveis de decisão.

A abordagem lagrangiana tradicional (Galvão e ReVelle, 1996) relaxa o conjunto de restrições (2) com um vetor μ de multiplicadores $\mu_i \geq 0$, $i \in N$, resultando:

$$(RL_\mu) \quad v(RL_\mu) = \text{Max} \sum_{i \in N} (w_i - \mu_i) x_i + \sum_{i \in N} \sum_{j \in S_i} \mu_i y_j$$

sujeito a (3), (4) e (5).

Pela propriedade da integralidade verifica-se facilmente que $v(RL_\mu) \geq v(MCLP)$. Além disso, o limitante lagrangiano $v(RL_\mu)$ não é superior ao limitante fornecido pela relaxação de programação linear do MCLP.

Neste trabalho propõe-se uma técnica de particionamento em clusters, baseando-se na relaxação lagrangiana LagClus apresentada no trabalho de Ribeiro e Lorena (2007, 2008a, 2008b, 2008c). A relaxação LagClus é uma relaxação mais forte, que pode ser útil em várias aplicações (práticas ou teóricas) de grande porte. A primeira aplicação da relaxação LagClus foi em problemas de rotulação cartográfica, sendo depois adaptada, com bons resultados, para tratamento de problemas de carregamento de paletes e de armazenamento de polpa de madeira em navios dedicados. Recentemente, em Corrêa *et al* (2006) demonstra-se a aplicação da relaxação LagClus a problemas de localização não-capacitados, onde foi possível melhorar os limitantes conhecidos para um conjunto de instâncias de difícil resolução.

Considere o PLMC representado pela Figura 1, onde os pontos correspondem aos clientes a serem atendidos ($N = \{1, \dots, 12\}$) e os quadrados correspondem aos locais candidatos para a instalação das facilidades ($M = \{1, \dots, 7\}$). Os valores entre parênteses correspondem à demanda w_i , $\forall i \in N$. Para a distância de cobertura S dada, os conjuntos S_i , $\forall i \in N$, são definidos a seguir:

$$\begin{array}{llll} S_1 = \{1\} & S_4 = \{4\} & S_7 = \{2, 6\} & S_{10} = \{4\} \\ S_2 = \{1\} & S_5 = \{3, 4\} & S_8 = \{2, 5, 6\} & S_{11} = \{5, 7\} \\ S_3 = \{1, 2\} & S_6 = \{1, 2, 3, 5\} & S_9 = \{3, 4, 7\} & S_{12} = \{7\} \end{array}$$

Assumindo como $p = 3$ o número de facilidades a serem instaladas, a formulação matemática deste exemplo será:

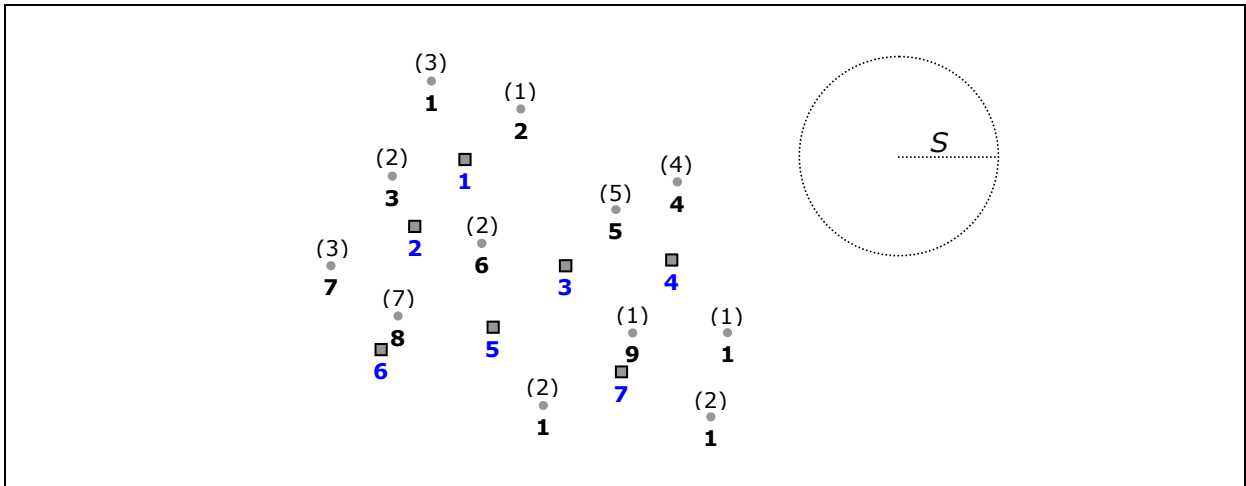


Figura 1 – Exemplo de um PLMC

$$v(PLMC) = \text{Max } 3x_1 + x_2 + 2x_3 + 4x_4 + 5x_5 + 2x_6 + 3x_7 + 7x_8 + x_9 + x_{10} + 2x_{11} + 2x_{12}$$

sujeito a:

$$y_1 \geq x_1 \quad (6)$$

$$y_1 \geq x_2 \quad (7)$$

$$y_1 + y_2 \geq x_3 \quad (8)$$

$$y_4 \geq x_4 \quad (9)$$

$$y_3 + y_4 \geq x_5 \quad (10)$$

$$y_1 + y_2 + y_3 + y_5 \geq x_6 \quad (11)$$

$$y_2 + y_6 \geq x_7 \quad (12)$$

$$y_2 + y_5 + y_6 \geq x_8 \quad (13)$$

$$y_3 + y_4 + y_7 \geq x_9 \quad (14)$$

$$y_4 \geq x_{10} \quad (15)$$

$$y_5 + y_7 \geq x_{11} \quad (16)$$

$$y_7 \geq x_{12} \quad (17)$$

$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 = 3 \quad (18)$$

$$x_i \in \{0, 1\}, \forall i \in N \quad (19)$$

$$y_j \in \{0, 1\}, \forall j \in M \quad (20)$$

Nesta abordagem para o PLMC, é preciso construir um *grafo de cobertura*, cuja definição está a seguir. Seja $G(M, A)$ um grafo onde M é o conjunto dos locais candidatos para a instalação de uma facilidade e $A = \{(p, q): p \text{ e } q \in S_i, i = 1, \dots, N\}$ é o conjunto de arestas. Assim, em um grafo de cobertura existe uma aresta conectando dois locais candidatos p e q se eles possuem, pelo menos, um cliente potencial em comum. A Figura 2 apresenta o grafo de cobertura associado ao PLMC do exemplo em questão.

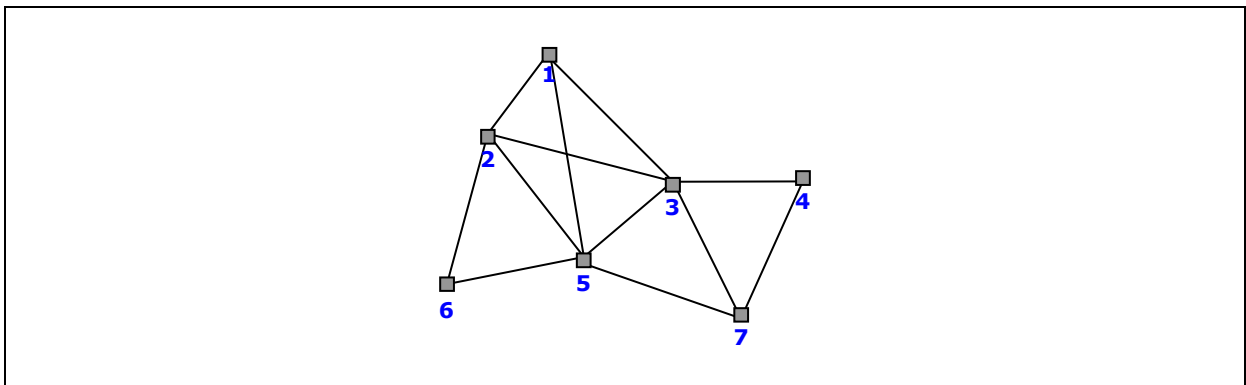
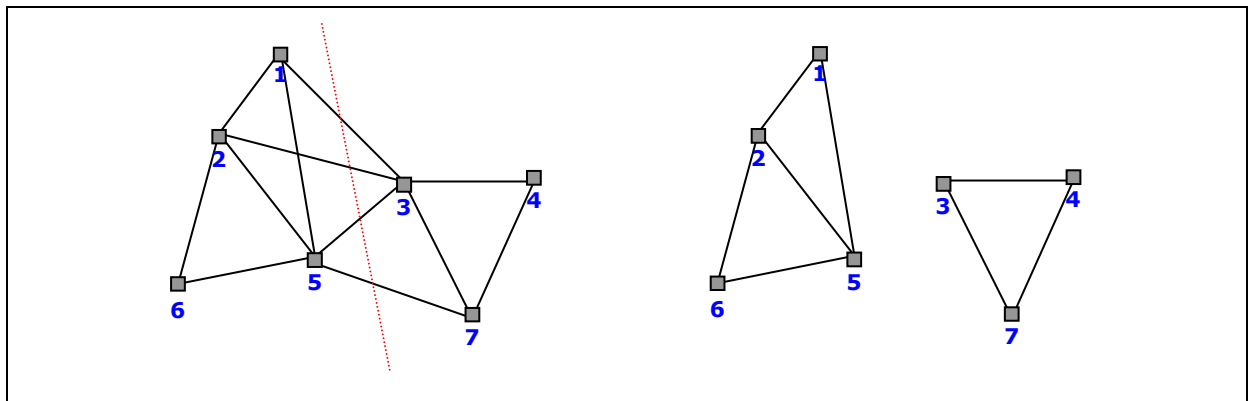


Figura 2 – O grafo de cobertura



(a) Um particionamento

(b) Os subgrafos resultantes

Figura 3 – Particionamento do grafo de cobertura

Nota-se que as arestas do grafo de cobertura estão associadas a um subconjunto das restrições (2) da formulação do PLMC. Neste exemplo, as arestas presentes na figura 2 correspondem às restrições (8), (10)-(14) e (16).

O particionamento de um grafo de cobertura consiste na remoção de um subconjunto de arestas visando obter dois ou mais subgrafos. A Figura 3(a) mostra um particionamento possível. Se as arestas (1,3), (2,3), (3,5) e (5,7) forem removidas, obtém-se os dois subgrafos mostrados na Figura 3(b).

Neste exemplo, este particionamento é obtido relaxando-se as restrições (11) e (16) da formulação do PLMC, utilizando os multiplicadores lagrangianos λ_6 e λ_{11} , respectivamente. Se a restrição (18) também for relaxada, utilizando-se o multiplicador lagrangiano μ , o problema resultante será:

$$\begin{aligned}
 v(PLMC_R) = \text{Max } & 3x_1 + x_2 + 2x_3 + 4x_4 + 5x_5 + 2x_6 + 3x_7 + 7x_8 + x_9 + x_{10} + 2x_{11} + 2x_{12} \\
 & + \lambda_6(y_1 + y_2 + y_3 + y_5 - x_6) + \lambda_{11}(y_5 + y_7 - x_{11}) \\
 & + \mu(y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 - 3) \\
 \text{sujeito a: } & (6) - (10), (12) - (15), (17), (19), (20), \text{ e} \\
 & \lambda_6, \lambda_{11} \geq 0 \\
 & \mu \in R
 \end{aligned}$$

A função objetivo pode ser reescrita como:

$$\begin{aligned}
 v(PLMC_R) = \text{Max } & 3x_1 + x_2 + 2x_3 + 4x_4 + 5x_5 + (2 - \lambda_6)x_6 + 3x_7 + 7x_8 + x_9 + x_{10} + \\
 & + (2 - \lambda_{11})x_{11} + 2x_{12} + (\lambda_6 + \mu)y_1 + (\lambda_6 + \mu)y_2 + (\lambda_6 + \mu)y_3 + \\
 & + \mu y_4 + (\lambda_6 + \lambda_{11} + \mu)y_5 + \mu y_6 + (\lambda_{11} + \mu)y_7 - 3\mu.
 \end{aligned}$$

Dessa forma, o problema original será decomposto em dois subproblemas:

$$v(PLMC_R) = v(SP_1) + v(SP_2) + (2 - \lambda_6)x_6 + (2 - \lambda_{11})x_{11} - 3\mu \quad (21)$$

onde:

$$\begin{aligned}
 v(SP_1) = \text{Max } & 3x_1 + x_2 + 2x_3 + 3x_7 + 7x_8 + \\
 & + (\lambda_6 + \mu)y_1 + (\lambda_6 + \mu)y_2 + (\lambda_6 + \lambda_{11} + \mu)y_5 + \mu y_6 \\
 \text{sujeito a: } & (6) - (8), (12), (13), (19), (20), \text{ e} \\
 & \lambda_6, \lambda_{11} \geq 0 \\
 & \mu \in R
 \end{aligned}$$

e

$$\begin{aligned}
v(SP_2) = \text{Max } & 4x_4 + 5x_5 + x_9 + x_{10} + 2x_{12} + \\
& + (\lambda_6 + \mu)y_3 + \mu y_4 + (\lambda_{11} + \mu)y_7 \\
\text{sujeito a: } & (9), (10), (14), (15), (17), (19), (20), \text{ e} \\
& \lambda_6, \lambda_{11} \geq 0 \\
& \mu \in R
\end{aligned}$$

Nota-se que estes subproblemas correspondem aos seguintes *clusters* (associados aos subgrafos do grafo de cobertura):

- *Cluster 1*: $M_1 = \{1, 2, 3, 7, 8\}$ e $N_1 = \{1, 2, 5, 6\}$;
- *Cluster 2*: $M_2 = \{4, 5, 6, 9, 10, 11, 12\}$ e $N_2 = \{3, 4, 7\}$.

A relaxação lagrangiana empregada não possui a propriedade da integralidade, sendo assim mais forte que a relaxação RL_μ de Galvão e ReVelle (1996). Além disso, o tamanho menor de cada cluster, em relação ao problema original, permite o emprego de métodos exatos para a resolução dos subproblemas, resultando no cálculo de melhores limitantes em tempo computacional reduzido.

No exemplo anterior, pode-se aplicar um método iterativo de otimização de subgradientes para determinar os valores das variáveis duais λ_6 , λ_{11} e μ , cujos valores serão utilizados para atualizar os coeficientes de custo das funções objetivo dos subproblemas, resolvidos por algum método exato.

É interessante observar neste exemplo que, com a relaxação das restrições (11) e (16), as variáveis x_6 e x_{11} não aparecem nas formulações dos subproblemas. Considerando a função objetivo (21), o valor destas variáveis será fixado em 0 ou 1, dependendo do valor do respectivo coeficiente a cada iteração, ou seja:

$$x_i = \begin{cases} 1, & \text{se } w_i - \lambda_i > 0 \\ 0, & \text{c.c.} \end{cases}$$

para todo índice $i \in I = \{i \in N \mid \text{a restrição (2) contendo a variável } x_i \text{ é relaxada}\}$.

O método de decomposição proposto pode ser sumarizado nos seguintes passos:

- a) Construa o grafo de cobertura $G(M, A)$ associado ao PLMC.
- b) Aplique uma heurística de particionamento para dividir o grafo de cobertura G em k clusters. Na formulação (1)-(5) do PLMC as restrições (2) poderão ser divididas em 2 grupos: um grupo relacionado às arestas intra-*clusters* de G e outro grupo formado por arestas inter-*clusters*.
- c) Relaxe as restrições correspondentes às arestas inter-*clusters* (definindo assim o conjunto I), usando multiplicadores lagrangianos não-negativos. Relaxe também a restrição (3).
- d) Decomponha a relaxação lagrangiana resultante em k subproblemas independentes.
- e) Aplique um método padrão de otimização de subgradientes para calcular o valor ótimo dos multiplicadores lagrangianos λ e μ .

O método de otimização de subgradientes utilizado no passo (e) está descrito a seguir:

Inicialização:

$$\lambda_i = \begin{cases} w_i, & \text{se } i \in I. \\ 0, & \text{c.c.} \end{cases}$$

$$\mu = 0;$$

$$\begin{aligned}\theta &= 2; \\ LI &= -\infty; \\ LS &= +\infty;\end{aligned}$$

Repita:

Resolva os subproblemas SP_k em x e y , com os valores correntes de λ e μ ;

Calcule:

$$v(PLMC_R) = \sum_k v(SP_k) + \sum_{i \in I} \max\{0, (w_i - \lambda_i)\} - p\mu;$$

Atualize $LS = \min\{LS, v(PLMC_R)\}$;

Se $\sum_{j \in N} y_j = p$ (solução viável para o PLMC), então:

Calcule $v(PLMC)$ com os valores correntes de x e y ;

Atualize $LI = \max\{LI, v(PLMC)\}$;

Calcule:

$$g_i^\lambda = x_i - \sum_{j \in S_i} y_j, i \in I;$$

$$g^\mu = p - \sum_{j \in N} y_j;$$

Atualize o tamanho do passo θ ,

Atualize:

$$\lambda_i = \min\{0, \lambda_i + \theta g_i^\lambda\}, i \in I;$$

$$\mu = \mu + \theta g^\mu;$$

Até (condições de parada).

O tamanho do passo θ utilizado neste algoritmo é o proposto em Held e Karp (1971), iniciando com $\theta = 2$ e dividindo-o ao meio sempre que o limitante superior se mantiver inalterado por um certo número de iterações sucessivas. As condições de parada utilizadas são:

- $\theta \leq 0.005$, ou
- $LS - LI < 1$, ou
- O vetor de subgradientes $g = [g_i^\lambda, g^\mu] = 0$.

3. Resultados computacionais

O algoritmo LagClus foi implementado em C e os testes foram realizados em um *notebook* com processador Intel Core 2 Duo 2.0 GHz e 2.0 GB RAM, Windows XP (Service Pack 3) e ILOG CPLEX 10.1.1 (ILOG, 2006). Os dados de rede e demanda utilizados foram obtidos da TSPLIB PCB3038 (Reinelt, 1994) e de casos reais de problemas de localização de facilidades em São José dos Campos - SP, Brasil (disponível para *download* em <http://www.lac.inpe.br/~lorena/instancias.html>).

A decomposição em subproblemas foi obtida pela heurística para problemas de particionamento de grafos METIS (Karypis e Kumar, 1998), executada com valores padrão. Para um dado grafo de cobertura G , a heurística METIS particiona G em k clusters, minimizando o número de arestas removidas com terminações em clusters distintos.

As tabelas 1 a 3 apresentam os resultados obtidos com a aplicação da técnica proposta. O significado das legendas das tabelas é descrito a seguir:

- k : número de *clusters*;
- n : número de clientes e de locais candidatos para a instalação de facilidades.;
- p : número de facilidades a serem instaladas;
- *Ótimo*: valor da solução ótima do PLMC obtido pelo CPLEX;

- *GapPL*: diferença da relaxação de programação linear fornecido pelo CPLEX (em %);
- *CPU*: tempo computacional para obtenção da solução ótima pelo CPLEX (em s);
- *Arcos*: número de arcos removidos (restrições relaxadas);
- $Gap = 100\% \times (LS - \acute{O}tima) / \acute{O}tima$;
- *CPUSeq*: tempo computacional total para todos os *clusters*, a cada iteração (em s);
- *CPUPar*: maior tempo computacional dentre todos os *clusters*, a cada iteração (em s).

A análise dos resultados apresentados permite afirmar que quanto menor for o número de *clusters*, melhor é a qualidade dos limitantes superiores obtidos (a coluna *Gap* apresenta valores menores). Por outro lado, se o número de *clusters* aumenta, o esforço computacional necessário para a resolução dos subproblemas diminui.

Em 92.3% dos casos apresentados, os resultados obtidos para $k = 5$ (dados da TSPLIB PCB3038) e $k = 10$ (dados de SJC) mostram valores na coluna *Gap* (em negrito) que são menores ou iguais aos obtidos pela relaxação de programação linear, o que demonstra a eficácia do método proposto. Valores ainda menores podem ser obtidos, mas deve-se considerar que isso pode aumentar o esforço computacional necessário.

Como demonstram as tabelas 1, 2 e 3, os limitantes calculados pela abordagem desenvolvida são melhores que os limitantes obtidos pela relaxação de programação linear e, conseqüentemente, por qualquer relaxação lagrangiana semelhante às apresentadas em Galvão e ReVelle (1996), Galvão *et al* (2000) e Lorena e Pereira (2002). Entretanto, ao se comparar os tempos computacionais relativos ao método proposto com os apresentados pelo CPLEX, fica evidente que esta proposta de decomposição é mais adequada para o tratamento de problemas de grande porte.

Tabela 1 – Tempos computacionais com dados de SJC, $S = 150$.

<i>n</i>	<i>p</i>	<i>Ótima</i>	<i>GapPL</i>	<i>CPU</i>	<i>k = 10</i>				<i>k = 50</i>			
					<i>Arcos</i>	<i>Gap</i>	<i>CPUSeq</i>	<i>CPUPar</i>	<i>Arcos</i>	<i>Gap</i>	<i>CPUSeq</i>	<i>CPUPar</i>
324	20	7302	0.000	0.015	296	0.000	2.543	2.537	1449	0.000	9.875	9.739
	30	9127	0.027	0.047		0.027	24.650	24.635		0.027	44.596	41.101
	40	10443	0.156	0.188		0.108	25.985	25.969		0.157	45.122	42.353
	50	11397	0.180	0.391		0.138	24.452	24.422		0.195	43.748	43.459
	60	11991	0.184	0.235		0.024	44.514	44.421		0.221	48.049	47.909
	80	12152	0.000	0.031		0.000	8.876	8.816		0.003	109.129	108.115
	108	12152	0.000	0.016		0.000	1.595	1.595		2.977	26.533	26.487
500	40	13340	0.000	0.047	108	0.000	3.453	3.361	804	0.000	17.186	11.499
	50	14773	0.014	0.047		0.000	4.938	4.514		0.014	59.019	36.293
	60	15919	0.048	0.063		0.000	8.233	7.243		0.048	57.157	34.737
	70	16908	0.000	0.031		0.000	3.723	3.370		0.000	22.891	14.421
	80	17749	0.000	0.015		0.000	5.406	4.766		0.000	26.686	16.697
	100	18912	0.098	0.109		0.000	10.276	7.171		0.056	62.071	37.748
	130	19664	0.041	0.297		0.015	30.827	24.588		0.041	69.934	43.373
167	19706	0.005	0.047		0.003	14.600	14.235		0.005	46.078	35.414	
818	80	23325	0.055	0.140	166	0.003	45.564	21.880	1649	0.061	85.922	45.819
	90	24455	0.123	0.266		0.041	56.388	24.747		0.143	87.797	47.001
	100	25435	0.127	0.344		0.012	87.279	34.481		0.140	96.124	52.060
	120	26982	0.084	0.297		0.015	69.658	31.368		0.062	105.547	54.658
	140	28002	0.140	0.359		0.095	52.966	26.271		0.128	121.127	63.713
	160	28699	0.128	0.391		0.107	58.453	24.904		0.126	96.017	50.828
	200	29153	0.018	0.234		0.011	61.531	28.301		0.039	253.908	135.048
	273	29168	0.000	0.031		0.000	3.343	2.545		0.554	46.766	37.178

Tabela 2 – Tempos computacionais com dados de SJC, $S = 200$.

<i>n</i>	<i>p</i>	<i>Optimal</i>	<i>GapPL</i>	<i>CPU</i>	<i>k</i> = 10				<i>k</i> = 50			
					<i>Arcos</i>	<i>Gap</i>	<i>CPUSeq</i>	<i>CPUPar</i>	<i>Arcos</i>	<i>Gap</i>	<i>CPUSeq</i>	<i>CPUPar</i>
324	20	9670	0.334	0.172	980	0.243	19.293	19.293	3008	0.347	32.943	32.911
	30	11737	0.087	0.484		0.060	28.943	28.943		0.094	69.959	69.959
	40	12151	0.005	0.094		0.008	31.066	31.066		0.138	77.872	77.612
	50	12152	0.000	0.015		0.000	9.926	9.926		0.000	61.456	61.395
	60	12152	0.000	0.047		0.000	4.575	4.575		0.000	14.376	14.376
	80	12152	0.000	0.016		0.000	3.670	3.670		0.000	33.936	33.936
	108	12152	0.000	0.031			0.248	11.343	11.343	0.001	897.830	893.405
500	40	17077	0.453	0.203	657	0.387	24.668	20.669	2625	0.469	55.001	51.048
	50	18361	0.014	0.109		0.003	39.109	32.248		0.025	67.626	62.596
	60	19153	0.035	0.063		0.005	52.639	35.363		0.112	85.374	76.578
	70	19551	0.110	1.078		0.069	43.946	29.817		0.170	76.671	68.721
	80	19703	0.013	0.156		0.008	35.495	27.927		0.150	102.056	95.253
	100	19707	0.000	0.078		0.000	16.624	16.501		0.001	89.858	86.698
	130	19707	0.000	0.047		0.000	1.986	1.864		0.001	26.546	25.809
	167	19707	0.000	0.016			0.016	22.379	22.225	0.859	22.314	22.133
818	80	27945	0.070	0.203	840	0.069	57.835	27.423	4910	0.121	147.155	115.605
	90	28519	0.138	1.141		0.071	114.145	45.536		0.177	128.574	99.585
	100	28910	0.103	1.391		0.036	88.885	33.153		0.175	101.875	80.758
	120	29165	0.002	1.234		0.002	55.710	31.180		0.117	141.246	115.434
	140	29168	0.000	0.125		0.000	11.643	8.940		0.021	171.961	143.737
	160	29168	0.000	0.062		0.000	9.738	7.598		0.878	39.343	36.244
	200	29168	0.000	0.032		0.000	5.762	4.610		0.847	37.205	34.871
	273	29168	0.000	0.031			0.207	24.698	18.505	2.904	25.282	23.772

Tabela 3 – Tempos computacionais com dados da TSPLIB PCB3038, $S = 400$.

<i>n</i>	<i>p</i>	<i>Ótima</i>	<i>GapPL</i>	<i>CPU</i>	<i>k</i> = 5				<i>k</i> = 10			
					<i>Arcos</i>	<i>Gap</i>	<i>CPUSeq</i>	<i>CPUPar</i>	<i>Arcos</i>	<i>Gap</i>	<i>CPUSeq</i>	<i>CPUPar</i>
3038	17	125320	0.368	802.390	165579	0.205	843.838	235.541	291363	0.470	582.528	223.245
	18	130004	0.517	10265.016		0.372	817.076	283.400		0.712	634.402	243.747
	19	134262*	0.605	20000.049		0.382	1483.237	598.653		0.793	576.821	222.087
	20	139028*	0.698	20000.156		0.500	1712.078	798.911		0.973	628.288	236.767
	21	141279*	0.853	20000.094		0.654	3117.174	1448.730		1.128	646.765	243.302
	22	143809*	1.196	20000.123		0.992	6656.267	3094.410		1.598	615.783	231.525

Na Tabela 3, os valores acompanhados de um asterisco destacam os casos em que o CPLEX não conseguiu obter a solução ótima em até 20.000 segundos. Tais valores correspondem, portanto, à soluções sub-ótimas.

Ao se considerar os tempos computacionais do método proposto para cada valor de k , nota-se que os valores de *CPUPar* tornam-se menores que os valores de *CPUSeq* à medida que o tamanho do problema aumenta, indicando que a resolução em paralelo (como em computadores com múltiplos núcleos de processamento) dos subproblemas pode reduzir substancialmente o tempo necessário para resolução de PLMCs com grande número de clientes e/ou facilidades.

4. Conclusões

Este trabalho apresenta um método de decomposição baseado em particionamento de *clusters* para calcular limitantes superiores de melhor qualidade para problemas de localização de máxima cobertura. O particionamento é realizado sobre o grafo de cobertura dos locais candidatos para atendimento de um determinado cliente. As restrições de acoplamento, uma vez identificadas, são relaxadas no sentido lagrangiano, resultando em subgrafos de tamanho

menor que o grafo do problema original. Cada subgrafo corresponde a um subproblema que pode ser resolvido, de forma independente, por métodos exatos e em tempo computacional reduzido. A eficiência desta decomposição foi comprovada com a resolução de problemas com dados reais e dados disponíveis na literatura.

Os dados de problemas com grande número de clientes e/ou facilidades permitiram observar a existência de um compromisso entre a qualidade dos limitantes e os tempos computacionais envolvidos. Dependendo da aplicação, pode-se priorizar a rapidez (aumentando-se o número desejado de *clusters*) sobre a qualidade dos limitantes. Se, no entanto, for necessário obter limitantes melhores, os tempos computacionais necessários para resolver problemas com poucos *clusters* pode aumentar significativamente.

Nos testes conduzidos, o número k de *clusters* é um parâmetro (fixado *a priori*) do algoritmo de particionamento, que busca reduzir o número de restrições a serem relaxadas, permitindo assim melhorar a qualidade dos limitantes obtidos. Nota-se que o particionamento em si desempenha um papel importante, o que estimula explorar outros critérios de decomposição, como por exemplo, minimizar o tamanho máximo de cada cluster, visando obter subproblemas com o menor tamanho possível.

O cálculo de limitantes apresentado neste trabalho pode ser usada em um método *branch-and-bound*. Como, em geral, os limitantes obtidos são melhores que os calculados pela relaxação de programação linear, poderá haver um aumento no número de podas e conseqüente redução do tamanho da árvore de busca.

Os avanços nas pesquisas em Matemática Aplicada e Ciência da Computação possibilitaram melhorias no desempenho de sistemas comerciais de otimização, permitindo a resolução de problemas complexos. Entretanto, o tempo computacional necessário para a resolução pode aumentar com o tamanho do problema, tornando-o intratável mesmo pelos sistemas mais eficientes. Nestes casos, uma abordagem que permite dividir um problema de grande porte em subproblemas menores pode ser a única alternativa.

Agradecimentos

Os autores agradecem ao CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico pelo apoio financeiro.

Referências bibliográficas

Arakaki, R.G.I.; Lorena, L.A.N. (2001) A constructive genetic algorithm for the maximal covering location problem. In: *Proceedings of the 4th Metaheuristics International Conference (MIC'2001)*. 16-20 Jul. Porto. Portugal. 13-17. Disponível para *download* em <http://www.lac.inpe.br/~lorena/arakaki/cgalap.pdf>.

Chung, C.H. (1986) Recent applications of the maximal covering location problem (MCLP) model. *Journal of the Operational Research Society* 37: 735-746.

Church, R.L.; ReVelle, C.S. (1974) The maximal covering location problem. *Papers of the Regional Science Association* 32: 101-118.

Corrêa, F.A.; Lorena, L.A.N. (2006) Using the constructive genetic algorithm for solving the probabilistic maximal covering location-allocation problem. In: *Proceedings of the 1st Workshop on Computational Intelligence/SBRN*. Disponível para *download* em http://www.lac.inpe.br/~lorena/correa/Correa_Lorena_Wci_2006.pdf.

Corrêa, F.A.; Lorena, L.A.N.; Senne, E.L.F. (2006) Lagrangean relaxation with clusters for the Uncapacitated Facility Location problem. In: *Proceedings of the XIII CLAIO - Congresso Latino-Iberoamericano de Investigación Operativa*, Montevideo, Uruguay, 27-30 Nov. Disponível para *download* em http://www.lac.inpe.br/~lorena/correa/LagClus_UFLP_CLAIO2006.pdf.

- Current, J.R.; O'Kelly, M.** (1981) Locating emergency warning sirens. *Decision Sciences* 23: 221-234.
- Daskin, M.** (1995) *Network and discrete location: models, algorithms and applications*. New York: Wiley Interscience. 500p
- Eaton, D.; Hector, M.; Sanchez, V.; Latingua, R.; Morgan, J.** (1986) Determining ambulance deployment in Santo Domingo, Dominican Republic. *Journal of the Operational Research Society* 37: 113-126.
- Galvão, R.D.; ReVelle, C.** (1996) A Lagrangean heuristic for the maximal covering location problem. *European Journal of Operational Research* 88: 114-123.
- Galvão, R.D.; Espejo, L.G.A.; Boffey, B.** (2000) A comparison of Lagrangean and surrogate relaxations for the maximal covering location problem. *European Journal of Operational Research* 124: 377-389.
- Galvão, R.D.** (2004) Uncapacitated facility location problems: contributions. *Pesquisa Operacional* 24: 7-38.
- Hale, T.S.; Moberg, C.R.** (2003) Location science review. *Annals of Operations Research* 123: 21-35.
- Held, M.; Karp, R.M.** (1971) The traveling salesman problem and minimum spanning trees: part II. *Mathematical Programming* 1: 6-25.
- Houglund, E.S.; Stephens, N.T.** (1976) Air pollutant monitor siting by analytical techniques. *Journal of the Air Pollution Control Association* 26: 52-53.
- ILOG** (2006) CPLEX 10.1.1: User's Manual. France.
- Karypis, G.; Kumar, V.** (1998) Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing* 48(1): 96-129.
- Lorena, L.A.N.; Pereira, M.A.** (2002) A Lagrangean/surrogate heuristic for the maximal covering location problem using Hillsman's edition. *International Journal of Industrial Engineering* 9: 57-67.
- Marianov, V.; Serra, D.** (1998) Probabilistic maximal covering location-allocation models for congested systems. *Journal of Regional Science* 38: 401-424.
- Marianov, V.; Serra, D.** (2001) Hierarchical location-allocation models for congested systems. *European Journal of Operational Research* 135: 195-208.
- Moore, G.C.; ReVelle, C.S.** (1982) The hierarchical service location problem. *Management Science* 28: 775-780.
- Reinelt, G.** (1994) The traveling salesman problem: computational solutions for TSP applications. *Lecture Notes in Computer Science* 840. Springer Verlag, Berlin.
- Ribeiro, G.M.; Lorena, L.A.N.** (2007) Lagrangean relaxation with clusters and column generation for the manufacturer's pallet loading problem. *Computers & Operations Research* 34: 2695-2708.
- Ribeiro, G.M.; Lorena, L.A.N.** (2008a) Lagrangean relaxation with clusters for point-feature cartographic label placement problems. *Computers & Operations Research* 35: 2129-2140.
- Ribeiro, G.M.; Lorena, L.A.N.** (2008b) Optimizing the wood pulp stowage using Lagrangean relaxation with clusters. *Journal of the Operational Research Society* 59: 600-606.
- Ribeiro, G.M.; Lorena, L.A.N.** (2008c) Column generation approach for the point-feature cartographic label placement problem. *Journal of Combinatorial Optimization* 15(2): 147-164.
- Serra, D.; Marianov, V.** (2004) *New trends in public facility location modeling*. Universitat Pompeu Fabra Economics and Business Working Paper 755. Disponível para download em <http://www.econ.upf.edu/docs/papers/downloads/755.pdf>.
- Toregas, C.; Swain, R.; ReVelle, C.; Bergman, L.** (1971) The location of emergency service facilities. *Operations Research* 19: 1363-1373.