

ABORDAGENS DE GERAÇÃO DE COLUNAS PARA UM PROBLEMA DE P-MEDIANAS CAPACITADO

Luiz Antonio Nogueira Lorena
LAC/INPE
Caixa Postal 515
12.201-970 São José dos Campos – SP
lorena@lac.inpe.br

Edson Luiz França Senne
FEG/UNESP
Caixa Postal 205
12515-410 Guaratinguetá – SP
elfsenne@feg.unesp.br

Resumo: O problema de p-medianas capacitado (PPMC) considera decisões ótimas sobre a localização de p facilidades levando-se em conta distâncias e capacidades de serviços. Apresentamos neste trabalho abordagens de geração de colunas para o PPMC. Os problemas mestres restritos consideram colunas formadas de clusters que satisfazem as restrições de capacidades, para os quais foram resolvidos problemas de 1-mediana. Os subproblemas geradores de novas colunas são então do tipo mochila. A relaxação Lagrangeana/surrogate é usada para gerar novas colunas através de um problema da mochila modificado, e também para fornecer bons limitantes ao processo. O processo de geração de colunas é acelerado mesmo no caso em que múltiplas colunas são selecionadas para entrar no problema mestre restrito. Resultados computacionais são apresentados para instancias geradas a partir de dados da cidade de São José dos Campos, SP.

Palavras-chave: Problemas de localização, Problema de p-medianas capacitado, Geração de colunas, Relaxação Lagrangeana/surrogate.

Abstract: The Capacitated p-median problem (CPMP) objectives the optimal location of p facilities, considering distances and capacities for the service to be given by each median. We present column generation approaches to CPMP. The identified restricted master problems optimize the covering of 1-median clusters satisfying the capacity constraints, and new columns are generated considering knapsack subproblems. The Lagrangean/surrogate relaxation has been used recently to accelerate subgradient like methods. In this work the Lagrangean/surrogate relaxation is directly identified from the master problem dual and provides new bounds and new productive columns through a modified knapsack subproblem. The overall column generation process is accelerated, even when multiple pricing is observed. Computational tests are presented running instances taken from real data of São José dos Campos city.

Keywords: Location problems, Capacitated p-median problems, Column generation, Lagrangean/surrogate relaxation.

1. Introdução

O problema de *p*-medianas capacitado (PPMC) considera decisões ótimas sobre a localização de *p* facilidades levando-se em conta distâncias e capacidades de serviços. Aparentemente o PPMC não foi estudado tão intensivamente como o problema clássico de *p*-medianas. Problemas relacionados foram tratados em Bramel and Simchi-Levi [3], Klein and Aronson [18], Mulvey and Beck [28] e Osman and Christofides [31]. Um ampla revisão bibliográfica e também um conjunto de problemas testes são apresentados em [31]. O PPMC é reconhecidamente *NP-difícil* e algumas abordagens anteriores aplicam *heurísticas Lagrangeanas*, tais como em Koskosidis and Powell [19] e em [29]. Abordagens recentes exploram o uso de meta-*heurísticas*, tais como *simulated annealing* e *busca tabu* (como em França *et al.* [13] e em [31]), e *algoritmos genéticos* (Maniezzo *et al.* [24]). Bons resultados foram apresentados para um conjunto padrão de instâncias (<http://mscmga.ms.ic.ac.uk/info.html> [2]).

A técnica de geração de colunas pode ser aplicada a problemas lineares de grandes dimensões, no caso de não se dispor de todas as colunas a priori, ou quando se pretende resolver um problema utilizando a decomposição de *Dantzig-Wolfe*, onde as colunas correspondem aos pontos extremos do conjunto convexo de soluções factíveis do problema. Neste caso, o algoritmo para resolução alterna entre um subproblema e um problema mestre restrito. A partir de um conjunto inicial de colunas, resolve-se o problema mestre, obtendo-se as variáveis duais que serão utilizadas pelo subproblema para determinar novas colunas a serem consideradas no problema mestre. Baseado no trabalho de Dantzig e Wolfe [6], a primeira aplicação prática desta técnica foi na determinação de padrões de corte unidimensionais (Gilmore and Gomory [14, 15]) e, desde então, seu uso difundiu-se de forma intensa [1, 4, 7, 8, 9, 30, 33, 34].

Abordagens baseadas na técnica de geração de colunas têm aparecido em um grande número de trabalhos recentes como alternativa aos métodos não lineares baseados em relaxação Lagrangeana (métodos de subgradientes e métodos bundle) para resolver problemas inteiros de grande porte [1]. Sabe-se, entretanto, que a aplicação direta do método de geração de colunas frequentemente produz um número muito grande de colunas que não são relevantes para a solução final, comprometendo assim a convergência para a solução do problema. Nestes casos, observa-se que as variáveis duais oscilam em torno da solução dual ótima. Logo, métodos que previnam tal comportamento podem acelerar a resolução do problema. Dentre estes merecem destaque: Método Boxstep [25], que restringe o espaço de busca de soluções duais a uma região limitada tendo a solução dual atual como centro; Método Analytic Centre Cutting Plane [10, 11], que usa o centro analítico de uma região da função dual como solução, no lugar da solução ótima, não permitindo assim mudanças muito drásticas entre as soluções duais de duas iterações consecutivas; Métodos Bundle [30], que combinam regiões de confiança e penalizações para que as soluções duais não variem muito de uma iteração para outra. Outros métodos estão descritos em Neame [30].

Neste trabalho explora-se a equivalência entre o método de geração de colunas, resultante da decomposição de Dantzig-Wolfe do problema original, e o método de Planos de Corte (Kelley [17]) para resolver o problema Lagrangeano associado, através do uso da relaxação Lagrangeana/surrogate descrita em Narciso e Lorena [29]. Esta abordagem pode ser vista como uma alternativa de estabilização do processo de geração de colunas, obtendo soluções duais e colunas de melhor qualidade, e acelerando a resolução do problema original.

O trabalho está organizado da seguinte forma. Na Seção 2 apresentam-se as formulações matemáticas dos problemas PPMC que são usadas neste trabalho. A próxima seção apresenta as abordagens de geração de colunas e a forma pela qual a relaxação Lagrangeana/surrogate é usada para gerar as novas colunas do problema mestre restrito. Na seção 4 apresentam-se os algoritmos e na seção seguinte os resultados dos experimentos computacionais.

2. Formulações do PPMC

O PPMC considerado neste trabalho é modelado de duas maneiras. A primeira considera o seguinte problema de programação linear inteira 0-1:

$$(P) \quad v(P) = \text{Min} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij} \quad (1)$$

$$\text{sujeito a} \quad \sum_{j \in N} x_{ij} = 1 ; i \in N \quad (2)$$

$$\sum_{j \in N} x_{jj} = p \quad (3)$$

$$\sum_{i \in N} q_i x_{ij} \leq Q x_{jj} ; j \in N \quad (4)$$

$$x_{ij} \in \{0,1\}; i \in N, j \in N \quad (5)$$

onde:

$N = \{1, \dots, n\}$ é o conjunto dos índices das entidades a serem alocadas e também o conjunto das possíveis medianas, das quais p medianas devem ser localizadas;

q_i é a demanda de cada entidade e Q a capacidade de cada possível mediana;

$[d_{ij}]_{n \times n}$ é a matriz de distâncias;

$[x_{ij}]_{n \times n}$ é a matriz de alocações, com $x_{ij}=1$ se a entidade i é alocada à mediana j , e $x_{ij}=0$, caso contrário; $x_{jj} = 1$ se a mediana j é selecionada e $x_{jj} = 0$, caso contrário.

As restrições (2) e (3) asseguram que cada entidade é alocada a apenas uma mediana. A restrição (4) assegura que o total de capacidades das medianas deve ser respeitado, e as restrições (5) apresentam condições de integralidades.

São assumidas capacidades iguais para as medianas no intuito de simplificar a formulação de particionamento de conjuntos que está definida a seguir.

O PPMC também pode ser formulado como o seguinte problema de particionamento de conjuntos:

$$(PC) \quad v(PC) = \text{Min} \sum_{k=1}^m c_k x_k$$

$$\text{sujeito a} \quad \sum_{k=1}^m A_k x_k = 1 \quad (6)$$

$$\sum_{k=1}^m x_k = p \quad (7)$$

$$x_k \in \{0,1\},$$

onde:

$S = \{S_1, S_2, \dots, S_m\}$, é um conjunto de subconjuntos de N ;

$A = [a_{ik}]_{n \times m}$, é uma matriz com $a_{ik} = \begin{cases} 1, & \text{se } i \in S_k \\ 0, & \text{caso contrário} \end{cases}$, satisfazendo $\sum_{i \in N} q_i a_{ik} \leq Q$;

e $c_k = \text{Min}_{i \in S_k^1} \left(\sum_{j \in S_k^1} d_{ij} \right)$, considerando $S_k^1 = \{i \in S_k \mid a_{ik} = 1\}$.

Esta é a formulação apresentada em Minoux [27]. A mesma formulação pode ser obtida do problema (P) aplicando a decomposição de Dantzig-Wolfe. Para cada subconjunto S_k^1 , as medianas abertas são decididas quando o custo da coluna c_k é calculado, e portanto as colunas de (PC) consideram implicitamente as restrições (4) de (P). As restrições (1) e (2) são conservadas e atualizadas respectivamente em (6) e (7), de acordo com o processo de decomposição de Dantzig-Wolfe.

Se S é o conjunto de todos os subconjuntos de N , a formulação apresentada pode levar à solução ótima de PPMC. Mas, como o número de subconjuntos pode ser muito grande, um conjunto parcial de colunas geralmente é considerado, levando o problema (PC) a ser chamado de problema mestre restrito dentro do contexto de geração de colunas [1].

3. A relaxação Lagrangeana/surrogate e o processo de geração de colunas

É bem conhecida a equivalência entre a decomposição de Dantzig-Wolfe, geração de colunas e a otimização da relaxação Lagrangeana. Resolver um programa linear através da decomposição de Dantzig-Wolfe resulta ser o mesmo que resolver o Lagrangeano pelo método de planos de corte de Kelley [17]. Entretanto em muitos casos a aplicação direta do método de geração de colunas pode resultar em convergência lenta.

A relaxação Lagrangeana/surrogate será usada em conjunto com o método de geração de colunas identificando os problemas de otimização que são comuns na solução do Lagrangeano/surrogate e no subproblema de custo reduzido. Desta forma além de proporcionar limitantes inferiores de boa qualidade, a relaxação Lagrangeana/surrogate estará gerando também colunas de qualidade para o processo de geração de colunas. Mais informações sobre a relaxação Lagrangeana/surrogate podem ser obtidas nos trabalhos de Lorena e Lopes [20], Lorena e Narciso [21], Lorena e Senne [22], Narciso e Lorena [29] e Senne e Lorena [32].

Para um dado vetor $\lambda \in R_+^n$ e $t \geq 0$ a relaxação Lagrangeana/surrogate de PPMC é dada por:

$$\begin{aligned} (L_t P \lambda) \quad v(L_t P \lambda) = \text{Min} \quad & \sum_{i \in N} \sum_{j \in N} (d_{ij} - t I_i) x_{ij} + t \sum_{i \in N} I_i \\ \text{sujeito a} \quad & (3), (4), \text{ e } (5). \end{aligned}$$

O problema $(L_t P \lambda)$ pode ser resolvido considerando a restrição (3) de forma implícita e decompondo (no índice j) o problema nos seguintes n sub-problemas da mochila 0-1:

$$\begin{aligned} v(knap^{tI})_j = \text{Min} \quad & \sum_{i \in N} (d_{ij} - t I_i) x_{ij} \\ \text{sujeito a} \quad & (4) \text{ e } (5). \end{aligned} \quad (8)$$

Cada sub-problema pode ser resolvido pelo algoritmo exato de Horowitz and Sahni (veja Martello e Toth [26]). Seja J o conjunto dos índices correspondentes aos p menores $v(knap^{tI})_j$, $j \in N$ (a restrição (3) está sendo considerada neste momento). O valor da relaxação Lagrangeana/surrogate é então dado por:

$$v(L_t P \lambda) = \sum_{j \in J} v(knap^{tI})_j + t \sum_{i \in N} I_i \quad (9)$$

A característica que torna interessante a relaxação $(L_t P \lambda)$ é que no caso particular em que $t = 1$, a expressão (9) dá o valor da relaxação Lagrangeana usual considerando o mesmo multiplicador λ .

Dois duais são então identificados: um dual externo que considera a variável multidimensional λ , usualmente otimizado por métodos do tipo subgradientes, e, considerando o multiplicador λ fixo, o melhor valor para a variável unidimensional t pode ser encontrado otimizando o dual Lagrangeano interno $v(D_t^I) = \underset{t \geq 0}{Max} v(L_t P \lambda)$. O melhor valor da relaxação Lagrangeana/surrogate é melhor que o da relaxação Lagrangeana. Uma forma aproximada de encontrar o melhor valor de t é resolver o dual interno por busca dicotômica (veja o algoritmo SH usado no trabalho [32]).

O problema a ser resolvido por geração de colunas é a versão de programação linear do problema de cobertura de conjuntos de (PC):

$$(CC) \quad v(CC) = \underset{Min}{\sum_{k=1}^m} c_k x_k$$

sujeito a

$$\sum_{k=1}^m A_k x_k \geq 1$$

$$\sum_{k=1}^m x_k = p$$

$$x_k \in [0,1].$$

Depois de definido um subconjunto inicial de colunas, o problema (CC) é resolvido e os custos duais finais $p_i, i = 1, \dots, n$ e μ são usados no seguinte subproblema para a geração de novas colunas (onde para simplificar a notação estamos usando para cada $j, x_{ij} = z_i$):

$$(Sub_{\pi}P) \quad v(Sub_{\pi}P) = \underset{j \in N}{Min} \{v(knap^P)_j\}$$

O subproblema (Sub $_{\pi}$ P) é resolvido decompondo no índice j e obtendo n problemas da mochila em variáveis 0-1 descritos na expressão (8). A coluna $\begin{bmatrix} z_i \\ 1 \end{bmatrix}$ é adicionada ao problema (CC) se $v(Sub_{\pi}P) < \lfloor m \rfloor$. Na prática, para $j = 1, \dots, n$, todas as colunas encontradas no processo de solução do subproblema que satisfaçam:

$$\sum_{i \in N} q_i z_i \leq Q \quad e \quad \sum_{i \in N} (d_{ij} - p_i) z_i < \lfloor m \rfloor \quad (10)$$

poderão ser adicionadas ao conjunto de colunas, acelerando o processo de geração de colunas.

Considerando o uso das variáveis duais $p_i, i = 1, \dots, n$, no problema $(L_t P^{\pi})$ (para o caso Lagrangeano, onde $t = 1$), as mesmas colunas identificadas em (Sub $_{\pi}$ P) estarão sendo identificadas. Portanto o problema Lagrangeano pode ser usado para gerar novas colunas e proporcionar um limitante inferior ao PPMC.

A relaxação Lagrangeana/surrogate é integrada ao processo de geração de colunas transferindo os multiplicadores p_i ($i = 1, \dots, n$) do problema (CC) ao problema $\underset{t \geq 0}{Max} v(L_t P^{\pi})$, e retornando as respectivas colunas. Se as novas colunas satisfazem a expressão (10), então podem ser adicionadas ao conjunto de colunas de (CC). O efeito resultante é uma aceleração da convergência do processo de geração de colunas, mesmo no caso em que múltiplas colunas são adicionadas em cada iteração do processo.

A relaxação Lagrangeana/surrogate é um limitante válido para o processo de geração de colunas. Em particular é melhor que o limitante Lagrangeano usual (caso em que $t = 1$) e pode ser útil para definir parâmetros de convergência no problema mestre restrito. A Figura 1 mostra um comportamento típico de limitantes duais integrados ao processo de geração de colunas.

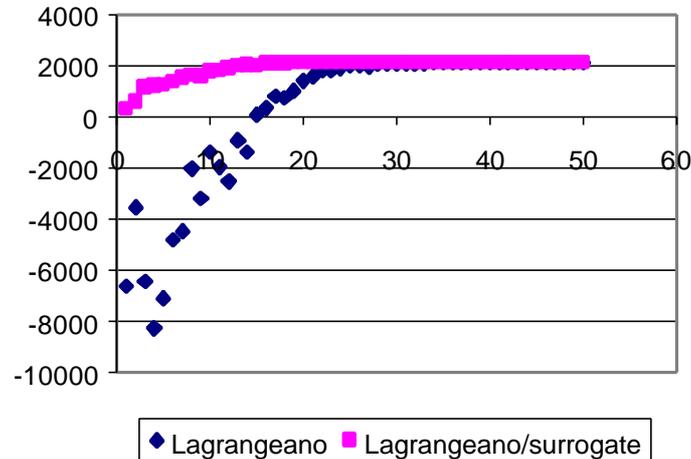


Figura 1. Comportamento típico dos limitantes duais

4. Algoritmos de geração de colunas

Os algoritmos de geração de colunas podem ser apresentados como:

CG (t)

- (i) Defina um conjunto inicial de colunas para (CC);
- (ii) Resolva (CC) usando o software CPLEX [16] e retorne os multiplicadores duais \mathbf{p}_j ($j = 1, \dots, n$) e μ ;
- (iii) Resolva aproximadamente (por busca dicotômica) o dual Lagrangeano/surrogate $\text{Max}_{t \geq 0} v(L_t P^P)$, retornando as colunas identificadas;
- (iv) Adicione em (CC) as colunas $\begin{bmatrix} z_i \\ 1 \end{bmatrix}$ que satisfazem a expressão (10);
- (v) Se não forem encontradas novas colunas no passo (iv) então pare;
- (vi) Faça os testes para remover colunas e volte para o passo (ii).

Os passos (i) e (vi) estão descritos a seguir. O algoritmo CG(1) reproduz o processo tradicional de geração de colunas, ou seja, no passo (iii) calcula o limite Lagrangeano $v(L_1 P^P)$. Em qualquer caso os limites $v(\text{CC})$ e $v(L_t P^P)$ são calculados a cada iteração. O seguinte algoritmo é usado no passo (i):

Seja $nc = 0$.
 Enquanto $nc < \text{NCOLS}$ fazer:
 $dtotal = 0$;

$S = \{ 1, \dots, n \};$
 Enquanto $Q \geq dtotal$ fazer:
 Seja k um valor escolhido aleatoriamente de S ;
 $C = C \cup \{ k \}$
 $S = S - \{ k \}$
 $dtotal = dtotal + q_k$
 Fim enquanto;
 Determinar a melhor mediana do *cluster* C ;
 Acrescentar ao problema uma coluna referente ao *cluster* C ;
 $nc = nc + 1$;
 Fim enquanto.

Nos testes computacionais descritos a seguir utilizou-se $NCOLS = 1000$. Para garantir a viabilidade do problema inicial considerou-se ainda um *cluster* adicional contendo todos os nós e com custo bem alto. Para o processo de remoção de colunas usado no passo (vi) do algoritmo foram consideradas em cada iteração somente as 3000 colunas de menor custo reduzido.

5. Experimentos computacionais

Os algoritmos descritos neste trabalho foram implementados na linguagem C e testes computacionais foram realizados em uma estação de trabalho Sun Ultra30.

O conjunto de instâncias é composto de dados reais coletados com o auxílio do Sistema de Informações Geográficas ArcView (ESRI [12]), e representam a área central da cidade de São José dos Campos. Seis instâncias (100×10), (200×15), (300×25), (300×30), (402×30) e (402×40) foram criadas. Cada ponto é localizado em um quarteirão que apresenta uma demanda e também é um local possível de localização de medianas. A demanda foi estimada considerando o número de casas (apartamentos) em cada quarteirão. Um quarteirão vazio recebeu o valor demanda igual a 1. As

capacidades são então estimadas como $C = \left\lceil \frac{\sum \text{demandas}}{\text{número de medianas} \times \alpha} \right\rceil$, onde α é 0.9 ou 0.8. Estas

instâncias estão disponíveis em <http://www.lac.inpe.br/~lorena/instancias.html>.

Nos experimentos computacionais realizados utilizou-se um limite máximo de 300 iterações. As tabelas 1 e 2 mostram os resultados computacionais obtidos pelos algoritmos CG(t) e CG(1) considerando estas instâncias. As tabelas apresentam como colunas:

- Instância – a identificação da instância;
- n – número de nós da rede;
- p – número de medianas a serem localizadas;
- Primal – o melhor valor obtido para (CC);
- Dual – o melhor valor obtido no dual Lagrangeano/surrogate;
- Gap = $100 * | \text{Primal} - \text{Dual} | / \text{Primal}$;
- Col.G – número de colunas geradas;
- Tempo – Tempo computacional total (em segundos).

Tabela 1 – Resultados obtidos por CG(t)

Instância	n	p	Primal	Dual	Gap	Col.G	Tempo
sjc1	100	10	17149.57	17149.56	–	3297	10.80
sjc2	200	15	33232.59	33231.89	–	9277	114.20
sjc3a	300	25	45244.70	45243.83	–	16480	351.51
sjc3b	300	30	40634.36	40634.34	–	17011	316.40
sjc4a	402	30	61850.64	61850.20	–	30931	1125.08
sjc4b	402	40	52403.89	52403.89	–	28643	729.01

Tabela 2 – Resultados obtidos por CG(1)

Instância	n	p	Primal	Dual	Gap	Col.G	Tempo
sjc1	100	10	16889.02	16893.91	–	4287	14.05
sjc2	200	15	33232.59	33231.78	–	12167	151.59
sjc3a	300	25	45240.03	45239.55	–	22499	553.76
sjc3b	300	30	40635.90	40635.89	–	21196	382.24
sjc4a	402	30	61816.25	61815.36	–	37240	1522.31
sjc4b	402	40	52369.48	52369.26	–	34769	894.25

Para efeito de comparação, a Tabela 3 apresenta os resultados obtidos com a heurística LSLSH(t), descrita em [23], aplicada sobre as mesmas instâncias. A heurística LSLSH(t) usa a relaxação Lagrangeana/surrogate com otimização do dual externo empregando um método de subgradientes, e também a viabilização de soluções duais que empregam heurísticas de localização-alocação (propostas em Cooper [5]). Esta tabela contém como colunas:

- P-limite – o melhor limite superior (primal) obtido para (PPMC);
- D-limite – o melhor limite inferior (dual) obtido para (PPMC);
- $\text{Gap} = 100 * | \text{P-limite} - \text{D-limite} | / \text{P-limite}$;
- Instancias, n, p e Tempo com as mesmas definições de colunas das Tabelas 1 e 2.

Tabela 3 – Resultados obtidos pelo algoritmo LSLSH(t)

Instância	n	p	P-limite	D-limite	Gap	Tempo
sjc1	100	10	17288.99	17252.12	0.213	68.62
sjc2	200	15	33395.38	33223.66	0.514	2083.37
sjc3a	300	25	45364.30	45313.43	0.112	2604.92
sjc3b	300	30	40635.90	40634.91	0.002	867.68
sjc4a	402	30	62000.23	61842.49	0.254	27717.11
sjc4b	402	40	52641.79	52396.54	0.466	4649.47

Das tabelas 1 e 2 nota-se que o algoritmo CG(t) é mais rápido e capaz de obter bons resultados gerando um número menor de colunas do que a abordagem tradicional dada pelo algoritmo CG(1). Ambos são ainda muito mais rápidos que a abordagem Lagrangeana/surrogate otimizada por um método de subgradientes (Tabela 3).

A capacidade do algoritmo CG(t) em obter bons resultados gerando poucas colunas torna-se ainda mais evidente quando o número de colunas úteis é limitado, conforme mostram os resultados da tabela a seguir. A Tabela 4 reporta os resultados obtidos pelos algoritmos CG(t) e CG(1) (valores entre parênteses) para a instância (402×30) e contém:

- N°Cols – número de colunas de menor custo reduzido mantidas a cada iteração do algoritmo;
- iter – número de iterações;
- Primal – o melhor valor obtido para (CC);
- Dual – o melhor valor obtido no dual Lagrangeano/surrogate;
- Gap = $100 * | \text{Primal} - \text{Dual} | / \text{Primal}$;
- Col.G – número de colunas geradas;
- Tempo – Tempo computacional total (em segundos).

Tabela 4 – Comportamentos de CG(t) e CG(1) com limitação do número de colunas

N°Cols	iter	Primal	Dual	Gap	Col.G	Tempo
3500	101 (120)	61850.64 (61665.98)	61849.90 (61666.28)	– (–)	30149 (39334)	1169.96 (1672.67)
3300	102 (126)	61850.64 (61844.14)	61850.64 (61843.67)	– (–)	30544 (38273)	1097.56 (1578.46)
3100	113 (134)	61849.29 (61667.16)	61849.26 (61667.16)	– (–)	32678 (40982)	1175.25 (1638.28)
3000	106 (119)	61850.64 (61816.25)	61850.20 (61815.36)	– (–)	30931 (37240)	1112.29 (1525.35)
2900	96 (300)	61850.64 (62108.74)	61850.63 (54691.75)	– (11.942)	29156 (82790)	1079.85 (1855.60)
2800	111 (300)	61850.64 (62483.74)	61850.23 (44403.46)	– (28.936)	32429 (90360)	1115.98 (1832.64)

6. Conclusões

Este trabalho apresentou abordagens de geração de colunas para o PPMC. Estas abordagens consideram o processo tradicional de geração de colunas integrado ao contexto da relaxação Lagrangeana/surrogate, onde novas colunas e limites de qualidade são identificados, acelerando o processo computacional.

Os resultados computacionais mostraram que a aplicação da relaxação Lagrangeana/surrogate gera um número menor de colunas quando comparado ao processo tradicional de solução e que também o problema mestre restrito continua operante mesmo com um número pequeno de colunas. Os limites Lagrangeano/surrogate podem ser úteis em árvores de busca *branch-and-price* e estão sendo objeto de pesquisa.

Agradecimentos: Os autores agradecem a FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo (proc. 99/06954-7) e CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico (processos 302408/88-6 e 300837/89-5) pelo apoio.

7. Referências

1. Barnhart, C.; Johnson, E.L.; Nemhauser, G.L.; Savelsbergh, M.W.P.; Vance, P.H. (1998) Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research* 46, 316-329.
2. Beasley, J.E. (1990) OR-Library: Distributing test problems by electronic mail. *Journal Operational Research Society* 41, 1069-1072.
3. Bramel, J.; Simchi-Levi, D. (1995) A location based heuristic for general routing problems. *Operations Research* 43, 649-660.
4. Carvalho, J.M.V. (1996) *Exact Solution of Bin-Packing Problems Using Column Generation and Branch-and-Bound*. Universidade do Minho, Departamento Produção e Sistemas, Working Paper.
5. Cooper, L. (1963) Location-allocation problems. *Operations Research* 11, 331-343.
6. Dantzig, G.B.; Wolfe, P. (1960) Decomposition principle for linear programs. *Operations Research* 8, 101-111.
7. Day, P.R.; Ryan, D.M. (1997) Flight Attendant Rostering for Short-Haul Airline Operations. *Operations Research* 45, 649-661.
8. Desrochers, M.; Desrosiers, J.; Solomon, M. (1992) A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. *Operations Research* 40, 342-354.
9. Desrochers, M.; Soumis, F. (1989) A Column Generation Approach to the Urban Transit Crew Scheduling Problem. *Transportation Science* 23, 1-13.
10. du Merle, O.; Goffin, J.L.; Vial, J.P. (1998) On Improvements to the Analytic Centre Cutting Plane Method. *Computational Optimization and Applications* 11, 37-52.
11. du Merle, O.; Villeneuve, D.; Desrosiers, J.; Hansen, P. (1999) Stabilized column generation. *Discrete Mathematics* 194, 229-237.
12. ESRI (1996) *Avenue Customization and Application Development for ArcView*. Environmental Systems Research Institute, Inc.
13. França, P.M.; Sosa, N.M.; Pureza, V. (1999) An adaptive tabu search algorithm for the capacitated p-median problem. *International Transactions in Operations Research* 6, 665-678.
14. Gilmore, P.C.; Gomory, R.E. (1961) A linear programming approach to the cutting stock problem. *Operations Research* 9, 849-859.
15. Gilmore, P.C.; Gomory, R.E. (1963) A linear programming approach to the cutting stock problem-part II. *Operations Research* 11, 863-888.
16. ILOG (1999) *CPLEX 6.5*. ILOG Inc. CPLEX Division.
17. Kelley, J.E. (1960) The Cutting Plane Method for Solving Convex Programs. *Journal of the SIAM* 8, 703-712.
18. Klein, K.; Aronson, J.E. (1991) Optimal clustering: a model and method. *Naval Research Logistics* 38, 447-461.
19. Koskosidis, Y.A.; Powell, W.R. (1992) Clustering algorithms for consolidation of costumes orders into vehicle shipments. *Transportation Research B* 26, 365-379.
20. Lorena, L.A.N.; Lopes, F.B. (1994) A surrogate heuristic for set covering problems. *European Journal of Operational Research* 79, 138-150.
21. Lorena, L.A.N.; Narciso, M.G. (1996) Relaxation heuristics for a generalized assignment problem. *European Journal of Operational Research* 91, 600-610.
22. Lorena, L.A.N.; Senne, E.L.F. (1999) Improving traditional subgradient scheme for Lagrangean relaxation: an application to location problems. *International Journal of Mathematical Algorithms* 1, 133-151.
23. Lorena, L.A.N.; Senne, E.L.F. (2001) *Local search heuristics for capacitated p-median problems*. Submitted to: Networks and Spatial Economics.
24. Maniezzo, V.; Mingozzi, A.; Baldaci, R. (1998) A bionomic approach to the capacitated p-median problem. *Journal of Heuristics* 4, 263-280.

25. Marsten, R.M.; Hogan, W.; Blankenship, J. (1975) The Boxstep method for large-scale optimization. *Operations Research* 23, 389-405.
26. Martello, S.; Toth, P. (1990) *Knapsack problems: Algorithms and computer implementations*. John Wiley & Sons.
27. Minoux, M. (1987) A Class of Combinatorial Problems with Polynomially Solvable Large Scale Set Covering/Set Partitioning Relaxations. *RAIRO* 21 (2), 105–136.
28. Mulvey, J.M.; Beck, M.P. (1984) Solving capacitated clustering problems. *European Journal of Operational Research* 18, 339-348.
29. Narciso, M.G.; Lorena, L.A.N. (1999) Lagrangean/surrogate relaxation for generalized assignment problems. *European Journal of Operational Research* 114, 165-177.
30. Neame, P.J. (1999) *Nonsmooth Dual Methods in Integer Programming*. PhD Thesis, Department of Mathematics and Statistics, The University of Melbourne.
31. Osman, I.H.; Christofides, N. (1994) Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research* 1, 317-336.
32. Senne, E.L.F.; Lorena, L.A.N. (2000) *Lagrangean/Surrogate Heuristics for p-Median Problems*. In *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, M. Laguna and J.L. Gonzalez-Velarde (Eds.), Kluwer Academic Publishers, 115-130.
33. Vance, P. (1993) *Crew scheduling, cutting stock and column generation: solving huge integer programs*. PhD thesis, Georgia Institute of Technology.
34. Vance, P.H.; Barnhart, C.; Johnson, E.L.; Nemhauser, G.L. (1994) Solving Binary Cutting Stock Problems by Column Generation and Branch-and-Bound. *Computational Optimization and Applications* 3, 111-130.