

# Abordagens complementares para problemas de $p$ -medianas

**EDSON LUIZ FRANÇA SENNE, DR.**  
Professor Adjunto do DMA/FEG/UNESP  
E-mail: [elfsenne@feg.unesp.br](mailto:elfsenne@feg.unesp.br)

**LUIZ ANTONIO NOGUEIRA LORENA, DR.**  
Pesquisador Titular do LAC/INPE  
E-mail: [lorena@lac.inpe.br](mailto:lorena@lac.inpe.br)

## Resumo

A localização de  $p$ -medianas é um problema clássico de otimização combinatória. O objetivo é localizar em uma rede  $p$  nós (denominados medianas), de forma a minimizar a soma das distâncias de cada nó de demanda até sua mediana mais próxima. Neste trabalho aborda-se a relaxação lagrangeana/*surrogate* como técnica para resolver tais problemas. Discute-se a utilização desta relaxação em combinação com métodos de otimização por subgradientes e com métodos de geração de colunas. O trabalho apresenta testes computacionais que demonstram a eficiência dos algoritmos propostos, considerando problemas obtidos da literatura e problemas reais obtidos a partir de Sistemas de Informações Geográficas.

## Palavras-chave

Problemas de localização, problemas de  $p$ -medianas, relaxação lagrangeana/*surrogate*, geração de colunas, programação inteira.

## Complementary approaches for $p$ -median location problems

### Abstract

*The search for  $p$ -median vertices on a network is a classical combinatorial optimization problem. The objective is to locate  $p$  facilities (medians) such as the sum of the distances from each demand vertex to its nearest facility is minimized. This work presents the lagrangean/*surrogate* relaxation as a technique for solving such combinatorial problems. The paper discusses the use of this relaxation combined with subgradient optimization methods and with column generation methods. Computational tests which demonstrate the efficiency of the proposed approaches for solving  $p$ -median instances taken from the literature and obtained from Geographical Information Systems are presented.*

### Key words

*Location problems,  $p$ -median problems, lagrangean/*surrogate* relaxation, column generation, integer programming.*

## INTRODUÇÃO

Em um problema de localização deseja-se estabelecer os locais onde serão sediadas facilidades (fábricas, depósitos, hospitais, escolas, etc.) para atender, da melhor maneira possível, um conjunto espacialmente distribuído de pontos de demanda. Dada a sua variedade e importância prática, os problemas de localização vêm sendo estudados por muitos pesquisadores, existindo uma extensa literatura a respeito (BRANDEAU; CHIU, 1989).

O problema de p-medianas é um problema clássico de localização. O objetivo é determinar os locais de p facilidades (denominadas medianas) em uma rede de n nós, de modo a minimizar a soma das distâncias entre cada nó de demanda e a mediana mais próxima. O problema de p-medianas tem grande importância prática como, por exemplo, na localização de escolas (PIZZOLATO *et al.*, 2002) e de antenas de telecomunicação (LORENA; PEREIRA, 2002). A Figura 1 ilustra, no Sistema de Informações Geográficas ArcView (ESRI, 1996), a solução obtida por um algoritmo de p-medianas para um problema de máxima cobertura relativo à instalação de três antenas de telecomunicação de curto alcance (800 m), considerando uma rede com 708 nós da região central da cidade de São José dos Campos, SP.

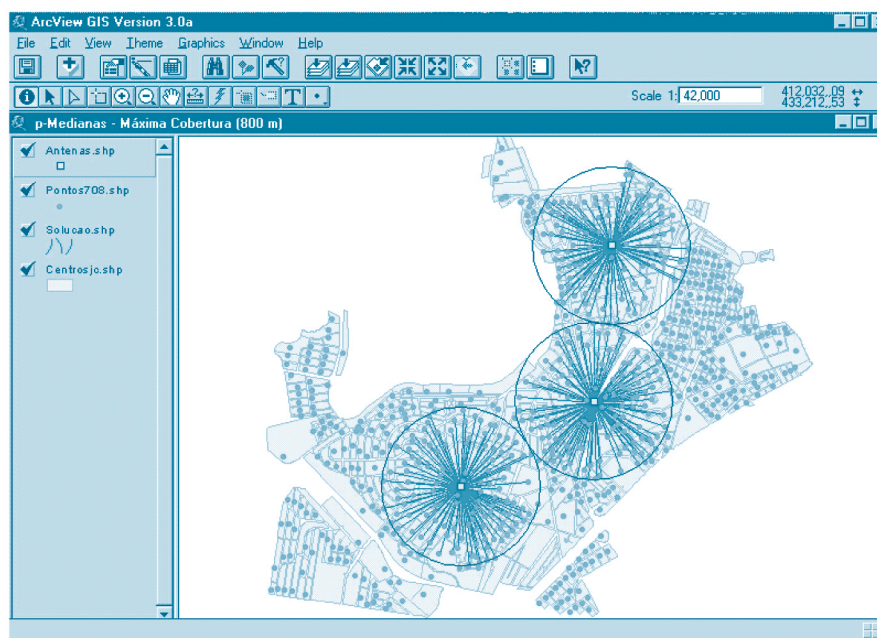
A localização de p-medianas é reconhecida como um problema difícil (GAREY; JOHNSON, 1979). Boas soluções podem requerer tempos computacionais excessivos para que possam ser consideradas, por exemplo, no contexto de tomadas de decisão. Discutem-se, neste trabalho, duas abordagens para a solução de problemas de p-medianas. Estas abordagens baseiam-se na combina-

O problema de p-medianas tem grande importância prática como, por exemplo, na localização de escolas e de antenas de telecomunicação.

ção de duas heurísticas bem conhecidas – a relaxação lagrangeana e a relaxação *surrogate* – que, aliadas a um processo de otimização por subgradientes ou a um método de geração de colunas, leva a algoritmos heurísticos eficientes.

Consideradas individualmente, tais abordagens já foram discutidas em trabalhos anteriores (LORENA; SENNE, 1999; SENNE; LORENA, 2000; SENNE; LORENA, 2002). Neste trabalho discute-se o comportamento computacional destas abordagens, em função do tamanho do problema e do número de medianas.

Figura 1: Solução de problema de máxima cobertura.



Mostra-se, a partir de testes computacionais, que o algoritmo baseado no método de otimização por subgradientes, para um dado valor de  $n$ , é muito eficiente para valores pequenos de  $p$ , perdendo eficiência à medida que o valor de  $p$  aumenta, enquanto que para o algoritmo baseado no método de geração de colunas ocorre o inverso, ou seja, maior eficiência é obtida para grandes valores de  $p$ . O comportamento complementar destes algoritmos pode ser muito importante para a construção de Sistemas de Apoio à Decisão.

### O PROBLEMA DE P-MEDIANAS E A RELAXAÇÃO LAGRANGEANA/SURROGATE

Nesta seção discute-se a aplicação de heurísticas de relaxação a problemas de  $p$ -medianas, introduzindo uma nova relaxação que combina as relaxações lagrangeana e *surrogate*.

O problema de  $p$ -medianas pode ser modelado como o seguinte problema (P) de programação inteira binária:

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

sujeito a:

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

$$\sum_{i \in N} x_{ii} = p \quad (2)$$

$$x_{ij} \leq x_{ii}; \quad i, j \in N \quad (3)$$

$$x_{ij} \in \{0,1\}; \quad i, j \in N \quad (4)$$

em que:

- $N = \{1, \dots, n\}$
- $[d_{ij}]_{n \times n}$  é uma matriz simétrica de custos (ou distâncias), com  $d_{ii} = 0, \forall i \in N$ ;
- $[x_{ij}]_{n \times n}$  é uma matriz de alocação, com  $x_{ij} = 1$  se o nó  $j$  é alocado à mediana  $i$ , e  $x_{ij} = 0$ , caso contrário;  $x_{ij} = 1$  se o nó  $i$  é uma mediana e  $x_{ii} = 0$ , caso contrário;
- $p$  é o número de medianas;
- $n$  é o número de nós.

As restrições (1) e (3) garantem que cada nó  $j$  é alocado a apenas um nó  $i$ , o qual deve ser uma mediana. A restrição (2) determina o número exato de medianas a ser localizado e a restrição (4) corresponde às condições de integralidade.

Esta formulação do problema de  $p$ -medianas é de difícil solução. Mesmo com os avanços nos *softwares* dedicados à solução de problemas de Programação Matemática, tal como o CPLEX (1999), a formulação (P) pode ser resolvida de forma ótima somente para problemas pequenos. Desta forma, outras opções têm sido tentadas para a solução do problema (P). Uma destas tentativas é a relaxação lagrangeana/*surrogate*, descrita nesta seção e que pode ser resolvida através de um método de subgradientes, descrito na seção 3. Na seção 4 apresenta-se uma outra formulação possível para o problema de  $p$ -medianas, que também emprega a relaxação lagrangeana/*surrogate*.

A relaxação lagrangeana/*surrogate* é uma combinação de duas relaxações bem conhecidas. Descreve-se a seguir como obter tal relaxação para o problema P. Para um dado vetor  $\lambda \in R_+^n$ , uma relaxação *surrogate* de (P) – denotada por  $(SP^\lambda)$  – pode ser definida como (GLOVER, 1968):

$$v(SP^\lambda) = \text{Min} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}$$

sujeito a:

$$\sum_{j \in N} \lambda_j \sum_{i \in N} x_{ij} = \sum_{j \in N} \lambda_j \quad (5)$$

e (2), (3) e (4)

Nesta formulação relaxa-se, no sentido *surrogate*, a restrição (1), ou seja, a de que cada nó é alocado a apenas uma mediana. Como  $SP^\lambda$  é uma relaxação de um problema de minimização, sabe-se que  $v(SP^\lambda) \leq v(P)$  e que seu melhor valor pode resultar do problema conhecido como dual *surrogate*, ou seja:

$$\text{Max}_{\lambda > 0} v(SP^\lambda)$$

O problema  $(SP^\lambda)$ , no entanto, ainda não pode ser resolvido facilmente, pois é um problema linear inteiro sem uma estrutura especial que possa ser explorada. Assim, pode-se relaxar o problema novamente, agora no sentido lagrangeano. Para um dado  $t \geq 0$ , relaxa-se a restrição (5), obtendo a relaxação lagrangeana/*surrogate* – denotada por  $(L'SP^\lambda)$  – que é dada por:

$$v(L'SP^\lambda) = \text{Min} \sum_{j \in N} \sum_{i \in N} (d_{ij} - t\lambda_j) x_{ij} + t \sum_{j \in N} \lambda_j$$

sujeito a (2), (3) e (4). Novamente, por se tratar de uma relaxação de um problema de minimização, sabe-se que, para um dado  $t \geq 0$  e  $\lambda \in \mathbb{R}_+^n$ ,  $v(L^tSP^\lambda) \leq v(SP^\lambda) \leq v(P)$ .

Relaxado desta forma, o problema pode ser resolvido facilmente. Para isto, basta considerar como implícita a restrição (2) e decompor o problema  $(L^tSP^\lambda)$  para o índice  $i$ , o que leva aos seguintes  $n$  subproblemas:

$$\text{Min} \sum_{j \in N} (d_{ij} - t\lambda_j)x_{ij} \quad i \in N$$

sujeito a (3) e (4). Cada um destes  $n$  subproblemas pode ser resolvido facilmente fazendo:

$$\beta_i = \sum_{j \in N} \{ \text{Min} (0, d_{ij} - t\lambda_j) \} \quad i \in N$$

e escolhendo  $I$  como o conjunto dos índices dos  $p$  menores  $\beta_i$ . Neste caso, a solução  $x^{t,\lambda}$  para o problema  $(L^tSP^\lambda)$  é dada por:

$$x_{ii} = 1, \text{ se } i \in I \text{ (caso contrário, } x_{ii} = 0)$$

e, para todo  $j \neq i$ ,  $x_{ij} = 1$ , se  $i \in I$  e  $d_{ij} - t\lambda_j < 0$  (caso contrário,  $x_{ij} = 0$ ).

Assim, a solução da relaxação lagrangeana/surrogate é dada por:

$$v(L^tSP^\lambda) = \sum_{i \in N} \beta_i x_{ii} + t \sum_{j \in N} \lambda_j$$

A solução  $x^{t,\lambda}$  não é necessariamente viável para o problema (P), pois como relaxou-se a restrição (1), um nó pode estar alocado a mais de uma mediana. Entretanto, o conjunto  $I$  identifica quais nós devem ser considerados como medianas para produzir soluções viáveis para (P). Duas heurísticas são usadas para, a partir de  $x^{t,\lambda}$ , determinar uma solução primal viável  $x_f$ . A primeira procura alocar cada nó  $j$  à mediana  $i$  mais próxima, ou seja, calcula simplesmente:

$$v(x_f) = \sum_{j \in N} (\text{Min}_{i \in I} d_{ij})$$

A segunda, é uma heurística de localização-alocação considerada por Cooper (1963) e Taillard (1996) que se baseia na observação de que após a definição de  $x_p$ , exatamente  $p$  aglomerados (*clusters*) podem ser identificados, correspondendo às  $p$ -medianas e suas não-medianas alocadas. A solução  $x_f$  pode ser melhorada procurando-se por uma nova mediana dentro de cada aglomerado, trocando-se a mediana atual por um nó não-mediana e recalculando-se as alocações. Caso o conjunto  $I$  se altere,

recalcula-se  $v(x_f) = \sum_{j \in N} (\text{Min}_{i \in I} d_{ij})$  e, se a nova solução for melhor, pode-se repetir o processo de realocação dentro dos novos aglomerados. Este processo se repete até que não seja mais possível obter novas melhorias.

É interessante observar que, para um dado  $\lambda \in \mathbb{R}_+^n$  a relaxação lagrangeana é um caso particular da relaxação  $L^tSP^\lambda$ , bastando para isso considerar o fator  $t$  como unitário.

É interessante observar que, para um dado  $\lambda \in \mathbb{R}_+^n$  a relaxação lagrangeana é um caso particular da relaxação  $L^tSP^\lambda$ , bastando para isso considerar o fator  $t$  como unitário.

### A RELAXAÇÃO LAGRANGEANA/SURROGATE E O MÉTODO DE OTIMIZAÇÃO POR SUBGRADIENTES

O uso combinado da relaxação lagrangeana e otimização por subgradientes de um ponto de vista primal-dual tem sido considerado como uma boa técnica de solução para o problema de  $p$ -medianas (BEASLEY, 1993).

O método de subgradientes é usado para solução de problemas de otimização de funções que podem possuir pontos não diferenciáveis. Nestes pontos o gradiente é substituído pelo conceito de subgradiente. É interessante notar que para problemas formulados como em (P) e relaxados através da relaxação lagrangeana (ou lagrangeana/surrogate) a identificação dos subgradientes é imediata, tomando o valor das restrições relaxadas nos pontos de busca (NEMHAUSER; WOLSEY, 1988; WOLSEY, 1998). O algoritmo LSOS, apresentado a seguir, mostra o método de subgradientes usado neste trabalho para obter a solução do problema dual lagrangeano/surrogate.

Considerando a relaxação  $L^tSP^\lambda$  para um multiplicador  $\lambda$  fixo, o melhor valor de  $t$  pode ser obtido do seguinte dual lagrangeano:

$$v(D^{t,\lambda}) = \text{Max}_{t \geq 0} v(L^tSP^\lambda)$$

Observe que, como  $v(D^{t,\lambda}) \geq v(L^tSP^\lambda)$ , o limite obtido pela relaxação lagrangeana/surrogate é melhor do que o obtido pela relaxação lagrangeana usual (que considera  $t = 1$ ).

Sabe-se que  $v(L^tSP^\lambda)$  em função de  $t$  é côncava e linear por partes. Assim, uma boa aproximação para o fator  $t$  pode ser obtida por um procedimento de busca dicotômica.

A combinação da relaxação lagrangeana/surrogate com o método de otimização por subgradientes pode ser

estabelecida pelo seguinte algoritmo:

**Algoritmo LSOS (relaxação lagrangeana/surrogate com otimização por subgradientes)**

Dados  $\lambda > 0$ , fazer  $lb = -\infty$ ,  $ub = +\infty$ ;

Repetir:

Determinar o fator  $t$  usando o procedimento de busca dicotômica apresentado em Senne; Lorena (2000);

Resolver a relaxação  $L^1SP^\lambda$ , obtendo a solução  $x^{t,\lambda}$  e  $v(L^1SP^\lambda)$ ;

Obter a solução viável  $x_f$ , por meio das heurísticas apresentadas na seção anterior;

Atualizar o limite inferior, fazendo:  $lb = \text{Max} [lb, v(L^1SP^\lambda)]$ ;

Atualizar o limite superior, fazendo:  $ub = \text{Min} [ub, v(x_f)]$ ;

Calcular  $g_i = 1 - \sum_{j \in N} x_{ij}^{t,\lambda}$ ,  $i \in N$ ;

Atualizar o tamanho do passo, fazendo:  $\theta = \pi (ub - lb) / \|g^\lambda\|^2$ ;

Atualizar os multiplicadores lagrangeanos, fazendo  $\lambda_i = \lambda_i + \theta g_i$ ,  $i \in N$ ;

Enquanto (as condições de parada não forem verificadas).

Para este algoritmo:

- o vetor  $\lambda$  inicial usado é dado por:  $\lambda_i = \text{Min}_{j \in N} \{d_{ij}\}$ ,  $i \in N$ ;
- o controle do parâmetro  $\pi$ , usado para atualizar o tamanho do passo  $\theta$ , é como o proposto em Held; Karp (1971), ou seja,  $0 < \pi \leq 2$ , iniciando com  $\pi = 2$  e dividindo  $\pi$  ao meio sempre que o limite inferior  $lb$  não aumentar por 30 iterações sucessivas;
- as condições de parada utilizadas são:
  - a)  $ub - lb < 1$  ou  $\|g^\lambda\|^2 = 0$  (solução ótima), e, por razões práticas,
  - b)  $\pi \leq 0,005$ , que indica uma convergência lenta,
  - c) número de iterações  $> 1000$ .

**A RELAXAÇÃO LAGRANGEANA/SURROGATE E O MÉTODO DE GERAÇÃO DE COLUNAS**

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 (DANTZIG; WOLFE, 1960), onde as colunas correspondem aos pontos extremos do conjunto convexo de soluções factíveis do problema.

A primeira aplicação prática desta técnica foi na determinação de padrões de corte unidimensionais (GILMORE;

GOMORY, 1961; GILMORE; GOMORY, 1963) e, desde então, seu uso difundiu-se de forma intensa (DESROCHERS *et al.*, 1992; VANCE, 1993; VANCE *et al.*, 1994; du MERLE *et al.*, 1999).

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 (BARNHART *et al.*, 1998). Sabe-se, entretanto, que a aplicação direta do método de geração de colunas freqüentemente 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 *Bboxstep* (MARSTEN *et al.*, 1975), 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* (du MERLE *et al.*, 1999), 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* (NEAME, 1999), 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 (1999).

No processo tradicional de geração de colunas, as iterações do algoritmo consideram um problema-mestre restrito e um subproblema de geração de colunas. A solução do problema-mestre fornece uma solução dual que é usada no subproblema para determinar se existem novas colunas que podem ser acrescentadas ao problema. São bem conhecidas as equivalências entre a decomposição de Dantzig-Wolfe, o processo de geração de colunas, e a relaxação lagrangeana (KELLEY, 1960). Entretanto, em muitos casos a aplicação do processo de geração de colunas resulta em uma convergência lenta. Nesta seção discute-se como usar a heurística lagrangeana/surrogate como uma alternativa de estabilização do processo de geração de colunas para acelerar a solução de problemas de p-medianas.

O uso de geração de colunas para resolver problemas de p-medianas ainda não foi suficientemente explorado. Algumas tentativas iniciais aparecem em Garfinkel *et al.* (1974) e Swain (1974). Estes autores relatam problemas de convergência, mesmo para problemas em que o número de medianas é pequeno quando comparado ao número total de nós. Esta observação foi confirmada mais tarde

por Galvão (1981). A solução de grandes problemas usando uma abordagem estabilizada aparece em du Merle *et al.* (1999).

Na formulação descrita nesta seção, o problema de p-medianas considera a formação de agrupamentos (*clusters*) de pontos medianas e os respectivos pontos alocados a estas medianas. Tais agrupamentos podem ser representados por colunas de zeros (pontos que não fazem parte do agrupamento) e uns (pontos que fazem parte do agrupamento), que são usadas em uma formulação de particionamento de conjuntos com restrição de cardinalidade (número de medianas). Como o número de agrupamentos pode ser muito grande, a formulação de particionamento é resolvida gerando as colunas somente quando necessário. É interessante notar que esta abordagem, embora conhecida há cerca de 30 anos, não foi suficientemente explorada, pois os *softwares* de Programação Matemática não eram competitivos em tempo computacional com o método de subgradientes. Neste trabalho mostra-se que, dependendo do problema, esta abordagem pode ser mais rápida do que a do método de subgradientes.

O problema de p-medianas pode ser modelado como o seguinte problema (Q) de cobertura de conjuntos:

$$v(Q) = \text{Min} \sum_{j \in M} c_j x_j$$

sujeito a:

$$\begin{aligned} \sum_{j \in M} A_j x_j &= 1 \\ \sum_{j \in M} x_j &= p \\ x_j &\in \{0,1\}, j \in M \end{aligned}$$

em que:

$$M = \{1, \dots, m\};$$

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

$[a_{ij}]_{n \times m}$  é uma matriz com  $a_{ij} = 1$  se  $i \in S_j$ , e  $a_{ij} = 0$ , caso

$$\text{contrário; e } c_j = \text{Min}_{i \in S_j} \left( \sum_{k \in S_j} d_{ik} \right).$$

Esta é a formulação encontrada em Minoux (1987). A mesma formulação pode ser obtida do problema (P)

definido anteriormente, pela aplicação da decomposição de Dantzig-Wolfe considerada em Garfinkel *et al.* (1974) e Swain, (1974). Se S é o conjunto de todos os subconjuntos de N, a formulação pode obter uma solução ótima do problema. No entanto, como o número de subconjuntos pode ser muito grande, na prática, somente um conjunto parcial de colunas pode ser considerado.

O problema (PM) a ser resolvido pelo método de geração de colunas é a versão de Programação Linear deste problema de cobertura de conjuntos, ou seja:

$$v(\text{PM}) = \text{Min} \sum_{j \in M} c_j x_j$$

sujeito a:

$$\sum_{j \in M} A_j x_j \geq 1 \tag{6}$$

$$\sum_{j \in M} x_j = p \tag{7}$$

$$x_j \in [0,1]; j \in M$$

O problema (PM) é conhecido como problema-mestre restrito no contexto do processo de geração de colunas (BARNHART *et al.*, 1998). Na abordagem tradicional, após definir um conjunto inicial de colunas, o problema (PM) é resolvido e os custos duais finais  $\pi_j$  ( $j = 1, \dots, n$ ), referentes às restrições (6), e  $\alpha$ , referente à restrição (7), são usados para gerar novas colunas resolvendo-se o seguinte subproblema:

$$v(\text{SubP}) = \text{Min}_{i \in N} \left[ \text{Min}_{y_j \in \{0,1\}} \sum_{j \in N} (d_{ij} - \pi_j) y_j \right]$$

A solução do subproblema (SubP) é obtida estabelecendo-se (para cada  $i = 1, \dots, n$ )  $y_j = 1$ , se  $d_{ij} - \pi_j \leq 0$  e,

caso contrário,  $y_j = 0$ . A coluna  $\left[ \frac{y_j}{1} \right]$  é acrescentada ao

problema (PM) se  $v(\text{SubP}) < |\alpha|$ . Para acelerar o processo, todas as colunas que satisfazem a expressão

$$\left[ \text{Min}_{y_j \in \{0,1\}} \sum_{j \in N} (d_{ij} - \pi_j) y_j \right] < |\alpha|$$

podem ser acrescentadas ao problema (PM) e não apenas à coluna correspondente ao valor mínimo da expressão.

Como observado anteriormente, para um dado  $t \geq 0$  e  $\pi \in R_+^n$  o melhor valor de  $t$  para a relaxação (L'SP $^\pi$ ) pode

ser obtido resolvendo-se o problema dual  $\text{Max}_{t \geq 0} v(L^t SP^\pi)$ .

Assim, pode-se integrar a relaxação lagrangeana/*surrogate* ao processo de geração de colunas transferindo os multiplicadores  $\pi_j$  ( $j = 1, \dots, n$ ) obtidos do problema

(PM) para o problema  $\text{Max}_{t \geq 0} v(L^t SP^\pi)$ , a fim de descobrir

o melhor valor de  $t$  a ser usado para modificar o subproblema gerador de colunas. Com isso, a identificação de novas colunas para o problema (PM) passa a ser feita pelo seguinte subproblema:

$$v(\text{SubP}^t) = \text{Min}_{i \in N} \left[ \text{Min}_{y_j \in \{0,1\}} \sum_{j \in N} (d_{ij} - t \cdot \pi_j) y_j \right]$$

Assim, a combinação da relaxação lagrangeana/*surrogate* com o método de geração de colunas pode ser estabelecida pelo seguinte algoritmo:

**Algoritmo LSGC (relaxação lagrangeana/*surrogate* com geração de colunas)**

1. Estabelecer um conjunto inicial de colunas para o problema (PM);
2. Resolver (PM) usando o *software* CPLEX, obtendo os custos duais  $\pi_j$  ( $j = 1, \dots, n$ ) e  $\alpha$ ;
3. Resolver o dual lagrangeano/*surrogate*  $\text{Max}_{t \geq 0} v(L^t SP^\pi)$ , obtendo o melhor valor de  $t$ ;
4. Resolver o subproblema (SubP<sup>t</sup>), acrescentando ao problema (PM) todas as colunas  $\begin{bmatrix} y_j \\ 1 \end{bmatrix}$  que satisfazem  $\left[ \text{Min}_{y_j \in \{0,1\}} \sum_{j \in N} (d_{ij} - \pi_j) y_j \right] < |\alpha|$  ( $i = 1, \dots, n$ );
5. Parar, se o passo (4) não acrescentar novas colunas ao problema (PM);
6. Executar os testes de remoção de colunas e retornar ao passo (2).

Para estabelecer o conjunto inicial de colunas para o problema (PM) utiliza-se o seguinte algoritmo:

Seja  $Num\_Cols = 2000$ , o número máximo de colunas no conjunto inicial de colunas.  
 $ncols = 0$ ;  
 Enquanto ( $ncols < Num\_Cols$ ), repetir:  
 Seja  $M = \{n_1, n_2, \dots, n_p\} \subset N$  um conjunto de nós gerado aleatoriamente.  
 Para cada  $k = 1, 2, \dots, p$ , fazer:

$$S_k = \{n_k\} \cup \{i \in N - M \mid d_{in_k} = \text{Min}_{j \in M} (d_{ij})\}$$

$$c_k = \text{Min}_{j \in S_k} \left( \sum_{i \in S_k} d_{ij} \right)$$

Para cada  $j = 1, \dots, n$ , fazer:

$y_j = 1$  se  $j \in S_k$   
 $y_j = 0$ , caso contrário.

Incluir a coluna  $\begin{bmatrix} y_j \\ 1 \end{bmatrix}$  no conjunto inicial de colunas.

$ncols = ncols + p$ ;  
 Fim\_da\_repetição.

Para a remoção de colunas improdutivas pode-se utilizar o seguinte algoritmo:

Para  $i = 1, \dots, m$ , fazer:

Remover a coluna  $i$  do problema (PM) atual se:  $cr_i > cr\_med$ , em que:

- $m$  é o número total de colunas do problema (PM) atual;
- $cr_i$  é o custo reduzido da coluna  $i$  do problema (PM) atual ( $i = 1, \dots, m$ );
- $cr\_med$  é o valor médio dos custos reduzidos para o conjunto inicial de colunas do problema (PM).

**TESTES COMPUTACIONAIS**

Os algoritmos apresentados anteriormente foram programados na linguagem C e executados em uma estação de trabalho Sun Ultra30. Foram considerados, inicialmente, problemas da *OR-Library* (BEASLEY, 1990) para os quais a solução ótima é conhecida. Os resultados estão mostrados na Tabela 1. Esta tabela contém:

- o número de nós ( $n$ ) e o número de medianas ( $p$ );
- a solução inteira ótima do problema;
- a relação  $n/p$ ;
- $d\_prim = 100 * |(v(x_p) - \text{ótimo})| / \text{ótimo}$ , ou seja, o desvio percentual da melhor solução primal encontrada em relação ao valor da solução ótima;
- $d\_dual = 100 * (\text{ótimo} - v(L^t SP^\lambda)) / \text{ótimo}$ , ou seja, o desvio percentual da melhor solução dual encontrada em relação ao valor da solução ótima;
- $d\_cplx$  = desvio percentual da melhor solução do problema (PM), obtida pelo CPLEX (ILOG, 1999), em relação ao valor da solução ótima;
- o tempo computacional (em segundos).

A Tabela 1 mostra, para valores decrescentes da razão

$n/p$ , os comportamentos complementares dos algoritmos LSOS e LSGC: o algoritmo LSOS mostra-se muito eficiente para grandes valores de  $n/p$ , ao passo que o algoritmo LSGC mostra-se muito eficiente para valores pequenos de  $n/p$  (é bem conhecida a observação de que os problemas de p-medianas mais difíceis para abordagens como a do algoritmo LSOS ocorre exatamente quando  $p = n/3$  (CHRISTOFIDES; BEASLEY, 1982)).

Para tornar mais evidentes os comportamentos dos algoritmos LSOS e LSGC foram realizados testes computacionais considerando o problema PCB3038 (3038 nós), obtido da biblioteca TSPLIB (REINELT, 1994), para alguns valores de  $p$ . Pode-se notar pelos resultados

mostrados na Tabela 2 que à medida que o número de medianas aumenta, o desempenho do algoritmo LSGC melhora, a tal ponto que, para  $p = 500$ , já é melhor do que o algoritmo LSOS. Deve-se observar que, na Tabela 2, os desvios são calculados em função da melhor solução conhecida.

Os comportamentos dos algoritmos LSOS e LSGC também se comprovam com testes realizados com problemas reais obtidos a partir do Sistema de Informações Geográficas para a cidade de São José dos Campos, SP. A Tabela 3 mostra os resultados obtidos para um problema de 708 nós (ver Figura 1) para alguns valores de  $p$ . Esta tabela contém:

Tabela 1: Resultados para problemas da *OR-Library*.

				LSOS			LSGC		
n	p	n/p	ótimo	d_prim	d_dual	tempo	d_cplx	d_dual	tempo
400	5	80	8162	-	0,866	18,48	0,686	1,662	52807,93
300	5	60	7696	-	0,046	12,94	0,246	1,796	17889,12
200	5	40	7824	-	0,523	5,61	-	-	902,77
400	10	40	6999	-	0,440	24,23	-	-	36829,25
300	10	30	6634	-	0,131	14,43	-	-	10749,91
100	5	20	5819	-	-	0,76	-	-	36,35
200	10	20	5631	-	-	5,87	-	-	996,00
300	30	10	4374	-	-	15,74	-	-	831,22
400	40	10	4809	-	-	46,62	-	-	1055,20
100	33	3	1355	-	-	1,14	-	-	0,37
200	67	3	1255	-	-	12,33	-	-	1,29
300	100	3	1729	-	-	57,73	0,116	0,058	4,55
400	133	3	1789	-	-	231,51	0,112	0,950	6,21
500	167	3	1828	-	-	377,14	0,055	0,310	11,00
600	200	3	1989	-	-	879,95	0,302	0,285	15,81
700	233	3	1847	-	-	494,52	0,081	0,379	21,50
800	267	3	2026	-	-	1360,49	0,518	0,346	26,14
900	300	3	2106	0,047	0,004	2994,11	0,518	0,827	33,37

Tabela 2: Resultados para o problema PCB3038.

		LSOS			LSGC		
p	Solução conhecida	d_prim	d_dual	tempo	d_cplx	d_dual	tempo
300	187723,46	1,305	0,056	719,04	0,043	0,044	22235,02
350	170973,34	2,067	0,050	731,50	0,044	0,045	10505,93
400	157030,46	1,630	0,012	919,79	0,008	0,008	4686,27
450	145422,94	1,612	0,056	745,86	0,052	0,053	1915,84
500	135467,85	2,344	0,040	684,82	0,036	0,036	597,86



**Tabela 3: Resultados obtidos para problemas reais.**

p	1_sup	1_inf	gap1	tempo	1_cplx	1_inf	gap2	tempo
50	118863,44	117691,84	0,986	11,29	117720,57	117719,83	0,001	2100,08
100	79105,96	77932,72	1,483	13,08	77937,25	77937,25	-	54,51
150	60132,28	59595,13	0,893	15,49	59595,88	59595,88	-	15,46
200	49400,39	48013,53	2,807	15,61	48018,74	48018,74	-	7,48
250	41777,86	39779,90	4,782	18,12	39782,85	39782,85	-	5,50
300	36611,90	32987,72	9,899	16,55	32993,96	32993,53	0,001	3,69

- o número de medianas (p);
- 1\_sup = o valor da melhor solução primal encontrada pelo algoritmo LSOS;
- 1\_inf = v(LSP<sup>p</sup>), ou seja, o valor da melhor solução dual encontrada;
- 1\_cplx = o valor da melhor solução do problema (PM), obtida pelo CPLEX;
- gap1 = 100 \* (1\_sup - 1\_inf) / 1\_sup;
- gap2 = 100 \* |1\_cplx - 1\_inf| / 1\_cplx;
- o tempo computacional (em segundos).

## CONCLUSÃO

A relaxação lagrangeana/surrogate tem sido explorada como uma alternativa computacional mais rápida do

que a relaxação lagrangeana tradicional para a solução de problemas de otimização combinatória.

Os algoritmos de localização de p-mediana apresentados neste artigo vêm sendo utilizados para desenvolver ferramentas úteis para a solução de diversos problemas relativos à análise de redes urbanas (<http://www.lac.inpe.br/~lorena/ArsigIndex.html>).

Como se nota pelos resultados computacionais, o comportamento dos algoritmos LSOS e LSGC é muito interessante para o desenvolvimento de Sistemas de Apoio à Decisão que devem considerar grandes problemas obtidos a partir de dados reais. Os dois algoritmos podem estar disponíveis para serem usados nas situações que lhes forem mais favoráveis, permitindo oferecer a um planejador soluções de boa qualidade e rapidamente.

Artigo recebido em 21/01/2003

Aprovado para publicação em 07/08/2003

## Referências Bibliográficas

- 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.
- BEASLEY, J.E. (1990). OR-Library: Distributing test problems by electronic mail. *Journal of Operational Research Society*, 41(11), 1069-1072.
- BEASLEY, J.E. (1993) Lagrangean heuristics for location problems, *European Journal of Operational Research*, 65, 383-399.
- BRANDEAU, M.L.; CHIU, S.S. (1989). An Overview of Representative Problems in Location Research. *Management Science*, 35(6), 645-674.
- CHRISTOFIDES, N.; BEASLEY, J. E. (1982). A tree search algorithm for the p-median problems. *European Journal of Operational Research*, 10: 196-204.
- COOPER, L. (1963). Location-allocation problems. *Operations Research*, 11, 331-343.
- DANTZIG, G.B.; WOLFE, P. (1960) Decomposition principle for linear programs, *Operations Research*, 8, 101-111.
- DESROCHERS, M.; DESROSIERS, J.; SOLOMON, M. (1992). A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. *Operations Research*, 40, 342-354.
- DU MERLE, O.; VILLENEUVE, D.; DESROSIERS, J.; HANSEN, P. (1999). Stabilized column generation. *Discrete Mathematics*, 194, 229-237.
- ESRI (1996). Environmental Systems Research Institute, Inc. *Avenue Customization and Application Development for ArcView*.
- GALVÃO, R.D. (1981). A Note on Garfinkel, Neebe and Rao's LP Decomposition for the p-Median Problem, *Transportation Science*, 15(3), 175-182.
- GAREY, M. R. AND JOHNSON, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*, W. H. Freeman and Co., San Francisco.

- GARFINKEL, R.S.; NEEBE, W.; RAO, M.R. (1974). An Algorithm for the M-median Location Problem. *Transportation Science*, 8, 217-236.
- GILMORE, P.C.; GOMORY, R.E. (1961). A linear programming approach to the cutting stock problem. *Operations Research*, 9, 849-859.
- GILMORE, P.C.; GOMORY, R.E. (1963). A linear programming approach to the cutting stock problem - part ii. *Operations Research*, 11, 863-888.
- GLOVER, F. (1968). Surrogate constraints. *Operations Research*, 16(4), 741-749.
- HELD, M.; KARP, R. M. (1971). The traveling-salesman problem and minimum spanning trees: part II. *Mathematical Programming*, 1, 6-25.
- ILOG (1999). Cplex Division, ILOG Inc., *Cplex 6.5*.
- KELLEY, J.E. (1960). The Cutting Plane Method for Solving Convex Programs. *Journal of the SIAM*, 8, 703-712.
- 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(1), 57-67.
- LORENA, L. A. N.; SENNE, E. L. F. (1999). Improving Traditional Subgradiente Scheme for Lagrangean Relaxation: An Application to Location Problems, *International Journal of Mathematical Algorithms*, 1: 133-151.
- MARSTEN, R. M.; HOGAN, W.; BLANKENSHIP, J. (1975). The Boxstep method for large-scale optimization. *Operations Research*, 23, 389-405.
- MINOUX, M. (1987). A Class of Combinatorial Problems with Polynomially Solvable Large Scale Set Covering/Set Partitioning Relaxations. *RAIRO*, 21(2), 105-136.
- NEAME, P. J. (1999). *Nonsmooth Dual Methods in Integer Programming*. Phd Thesis - Department of Mathematics and Statistics, The University of Melbourne.
- NEMHAUSER, G.L.; WOLSEY, L.A. (1988). *Integer and Combinatorial Optimization*. Wiley, New York.
- PIZZOLATO, N. D.; BARCELOS, F. B.; LORENA, L. A. N. (2002). *School Location Methodology in Urban Areas of Developing Countries*. IFORS2002 - The sixteenth triennial conference of the International Federation of Operational Research Societies.
- REINELT, G. (1994). *The traveling salesman problem: computational solutions for TSP applications*. Lecture Notes in Computer Science 840, Springer Verlag, Berlin.
- 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.
- SENNE, E. L. F.; LORENA, L. A. N. (2002). Stabilizing column generation using Lagrangean/surrogate relaxation: An application to p-median location problems. *European Journal of Operational Research* (submetido).
- SWAIN, R.W. (1974). A Parametric Decomposition Approach for the Solution of Uncapacitated Location Problems, *Management Science*, 21, 955-961.
- TAILLARD, E.D. (1996). *Heuristic methods for large centroid clustering problems*, Technical report IDSIA96-96, IDSIA.
- VANCE, P. (1993). *Crew scheduling, cutting stock and column generation: solving huge integer programs*. PhD thesis, Georgia Institute of Technology.
- 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.
- WOLSEY, L. A. (1998). *Integer Programming*. John Wiley & Sons, New York; H. Freeman and Co., San Francisco.

## ■ Agradecimentos

Os autores agradecem à FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo e ao CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, pelos apoios financeiros recebidos. Agradecem também as sugestões recebidas de dois revisores anônimos.