



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

UM MÉTODO *BRANCH-AND-PRICE* PARA PROBLEMAS DE  
LOCALIZAÇÃO DE  $p$ -MEDIANAS.

Marcos Antonio Pereira

Tese de Doutorado em Computação Aplicada, orientada pelo Dr. Luiz Antônio Nogueira  
Lorena e pelo Prof. Dr. Edson Luiz França Senne.

INPE  
São José dos Campos  
2005

519.8

PEREIRA, M. A.

Um Método *Branch-and-Price* para Problemas de Localização de  $p$ -Medianas / M. A. Pereira - São José dos Campos: INPE, 2005.

88p. - ().

1. Otimização Combinatória. 2. Localização de Facilidades  
3. *Branch-and-Price*. 4. Relaxação Lagrangeana/*Surrogate*.

1. Combinatorial Optimization. 2. Facility Location. 3. Branch-  
and-Price. 4. Lagrangean/*Surrogate* Relaxation.

## **FOLHA DE APROVAÇÃO**



*De todos os infortúnios que afligem a humanidade, o mais amargo é que temos de ter consciência de muito e controle de nada.*

(Heródoto, historiador grego, séc V a.C.)

*O último esforço da razão é reconhecer que existe uma infinidade de coisas que a ultrapassam.*

(Blaise Pascal, filósofo e matemático francês, 1623-1662)



*A meus pais,  
MAURO PEREIRA e  
ANTÔNIA PEREIRA.*



## **AGRADECIMENTOS**

Agradeço a Deus, cujas bênçãos possibilitaram a realização deste trabalho.

Ao Dr. Luiz Antônio Nogueira Lorena e ao Prof. Dr. Edson Luiz França Senne, pelos conhecimentos adquiridos e pela competência, dedicação, paciência e amizade demonstrados o tempo todo.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES, pelo auxílio financeiro.

Aos docentes do curso de Computação Aplicada - CAP, pesquisadores do Laboratório Associado de Computação e Matemática Aplicada - LAC, e colegas de curso, pelas conversas proveitosas e pelas outras também.

À Maria Cristina Peloggia de Araújo, secretária do LAC, e Vanessa Aparecida de Oliveira, secretária da CAP, pela dedicação com que atenderam cada solicitação.

Ao Instituto Nacional de Pesquisas Espaciais - INPE, pela oportunidade de estudos e utilização de suas instalações.

A meus pais, por sempre acreditarem na importância do estudo.



## RESUMO

Este trabalho apresenta a implementação de um algoritmo *branch-and-price* para resolver problemas de localização de facilidades baseados no modelo matemático do problema de  $p$ -medianas. A abordagem tradicional de geração de colunas é comparada com uma nova proposta, onde o critério de custos relativos empregado na seleção de colunas é modificado pelo multiplicador da relaxação lagrangeana/*surrogate*. A eficiência da nova abordagem foi comprovada por testes computacionais envolvendo instâncias com até 900 vértices. Também foram realizados estudos com dados reais de problemas de máxima cobertura, formulados como problemas de  $p$ -medianas, cuja esparsidade nos vetores de custos e na matriz de restrições representam grande dificuldade para métodos baseados em geração de colunas.



# **A BRANCH-AND-PRICE METHOD FOR $p$ -MEDIAN LOCATION PROBLEMS.**

## **ABSTRACT**

This work presents a branch-and-price algorithm to solve facility location problems that are based on the mathematical formulation of  $p$ -median problems. The traditional column generation approach is compared to a new proposal, where the reduced cost criterion employed at the column selection is modified by the lagrangean/surrogate multiplier. The efficiency of the new approach is tested with instances up to 900 vertices. Tests with real data for maximal covering location problems, formulated as  $p$ -median problems, were conducted and showed the impact of sparsity on column generation based methods.



## SUMÁRIO

Pág.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SÍMBOLOS

LISTA DE SIGLAS E ABREVIATURAS

CAPÍTULO 1 - INTRODUÇÃO.....	23
CAPÍTULO 2 - O PROBLEMA DE <i>P</i> -MEDIANAS .....	29
2.1 - Levantamento Bibliográfico.....	29
2.2 - Formulações Matemáticas.....	31
2.2.1 - Problema Inteiro Binário.....	31
2.2.2 - Problema de Particionamento de Conjuntos.....	32
2.3 - Relaxação Lagrangeana/ <i>Surrogate</i> .....	34
2.3.1 - Cálculo do Multiplicador Lagrangeano/ <i>Surrogate</i> .....	36
CAPÍTULO 3 - APLICAÇÃO DA RELAXAÇÃO LAGRANGEANA/ <i>SURROGATE</i> AO MÉTODO DE GERAÇÃO DE COLUNAS .....	39
3.1 - O Problema Mestre Restrito.....	40
3.1.1 - O Conjunto Inicial de Colunas.....	41
3.2 - O Subproblema Gerador de Colunas.....	42
3.2.1 - Gerenciamento do Tamanho do Problema.....	43
3.3 - O Algoritmo de Geração de Colunas.....	44
3.4 - Resultados Computacionais.....	46
CAPÍTULO 4 - IMPLEMENTAÇÃO DO ALGORITMO <i>BRANCH-AND-PRICE</i> E RESULTADOS COMPUTACIONAIS .....	55
4.1 - Condições de Poda.....	55
4.2 - Regra de Ramificação.....	56
4.2.1 - Identificação de <i>q</i> .....	56
4.2.2 - Identificação de <i>r</i> .....	57
4.3 - Definição dos Subproblemas.....	57
4.4 - Resultados Computacionais.....	58
CAPÍTULO 5 - INSTÂNCIAS DIFÍCEIS: O PROBLEMA DE LOCALIZAÇÃO DE MÁXIMA COBERTURA.....	63
5.1 - Formulação Matemática.....	63
5.2 - Formulação de <i>p</i> -Medianas para o PLMC.....	64
5.3 - Geração de Colunas com Ampliação e Perturbação.....	66
5.4 - Resultados Computacionais.....	70

CAPÍTULO 6 - CONCLUSÃO.....	77
6.1 - Trabalhos Futuros.....	78
REFERÊNCIAS BIBLIOGRÁFICAS .....	81

## LISTA DE FIGURAS

2.1 - Limitantes lagrangeano/ <i>surrogate</i> . .....	36
2.2 - Algoritmo de busca do multiplicador lagrangeano/ <i>surrogate</i> .....	37
3.1 - Rotina de geração do conjunto inicial de colunas. ....	41
3.2 - Rotina de eliminação de colunas.....	44
3.3 - Esquema geral do algoritmo de geração de colunas. ....	45
3.4 - Algoritmo de geração de colunas.....	46
3.1 - Resultados obtidos para instâncias da <i>OR-Library</i> . ....	47
3.5 - Evolução de $v(\overline{PCC})$ para $fator\_rc = 0,5$ . ....	50
3.6 - Evolução de $v(\overline{PCC})$ para $fator\_rc = 0,4$ . ....	50
3.7 - Evolução de $v(\overline{PCC})$ para $fator\_rc = 0,3$ . ....	50
3.8 - Convergência do multiplicador lagrangeano/ <i>surrogate</i> .....	51
4.1 - Algoritmo <i>Branch-And-Price</i> .....	56
4.2 - Estrutura das colunas dos problemas segundo a regra de ramificação. ....	57
5.1 - Localização de antenas em São José dos Campos - SP.....	65
5.2 - $GC(t)$ sem ampliação ou perturbação: $v(L_t S_\lambda PPM) = 2121,40$ . ....	69
5.3 - $GC(t)$ com ampliação: $v(L_t S_\lambda PPM) = 2862,21$ .....	69
5.4 - $GC(t)$ com ampliação e controle de perturbação: $v(L_t S_\lambda PPM) = 2936,49$ . ....	70



## LISTA DE TABELAS

3.1 - Resultados obtidos para instâncias da <i>OR-Library</i> . .....	47
3.2 - Comparação entre <i>SG</i> e <i>GC(t)</i> . .....	49
3.3 - Influência do parâmetro <i>fator_rc</i> . .....	49
3.4 - Resultados computacionais para instâncias PCB3038 ( <i>fator_rc</i> = 1,0). .....	52
3.5 - Resultados computacionais para instâncias PCB3038 ( <i>fator_rc</i> = 0,5). .....	52
3.6 - Resultados computacionais para instâncias PCB3038 ( <i>fator_rc</i> = 0,2). .....	52
4.1 - Resultados obtidos para instâncias da <i>OR-Library</i> . .....	59
4.2 - Resultados obtidos para instâncias difíceis da <i>OR-Library</i> . .....	60
4.3 - Resultados obtidos pelo CPLEX para instâncias da <i>OR-Library</i> . .....	61
4.4 - Resultados obtidos pelo CPLEX para instâncias difíceis da <i>OR-Library</i> . .....	61
5.1 - Instâncias da cidade de São José dos Campos - SP. ....	66
5.2 - Resultados obtidos para instâncias de S. J. dos Campos com 324 vértices. ....	71
5.3 - Resultados obtidos para instâncias de S. J. dos Campos com 402 vértices. ....	71
5.4 - Resultados obtidos para instâncias de S. J. dos Campos com 500 vértices. ....	72
5.5 - Resultados obtidos para instâncias de S. J. dos Campos com 708 vértices. ....	72
5.6 - Resultados obtidos para instâncias de S. J. dos Campos com 818 vértices. ....	73
5.7 - Resultados para as instâncias de S. J. dos Campos pela formulação <i>PPM</i> . ....	74
5.8 - Resultados para as instâncias de S. J. dos Campos pela formulação <i>PMC</i> . ....	75



## LISTA DE SÍMBOLOS

$|S|$  - Cardinalidade do conjunto  $S$ .

$\text{Pot}(S)$  - Conjunto potência do conjunto  $S$ .



## LISTA DE SIGLAS E ABREVIATURAS

- FIFO - Estratégia *First In First Out*.
- GB - Gigabyte (1 GB =  $2^{30}$  bytes).
- HT - *Hyper Threading*: tecnologia presente em processadores Intel Pentium 4.
- LIFO - Estratégia *Last In First Out*.
- MB - Megabyte (1 MB =  $2^{20}$  bytes).
- PCC* - Formulação matemática do Problema de Cobertura de Conjuntos.
- $\overline{PCC}$  - Relaxação de Programação Linear do Problema de Cobertura de Conjuntos.
- PLMC - Problema de Localização de Máxima Cobertura.
- PMR - Problema Mestre Restrito.
- PPC* - Formulação matemática do Problema de Particionamento de Conjuntos.
- PPM* - Formulação matemática do Problema de  $p$ -Medianas.
- RAM - *Random Access Memory*, ou memória de acesso aleatório.
- VLSI - *Very Large Scale Integration*.



## CAPÍTULO 1

### INTRODUÇÃO

A logística de distribuição de produtos e/ou serviços tem recebido atenção crescente, ao longo dos anos, como parte do planejamento estratégico de empresas do setor público e privado. Decisões sobre a melhor configuração para instalação de facilidades destinadas ao atendimento da demanda de uma população são tratadas em uma ampla classe de problemas, conhecida como Problemas de Localização (Drezner, 1995; Daskin, 1995). Utilizando uma representação em grafo, os locais de demanda e de instalação dos centros de atendimento/distribuição são identificados como vértices em uma rede, podendo ocorrer casos em que facilidades podem ser abertas no interior das arestas. Estes problemas ocorrem tipicamente em um espaço discreto, ou seja, em um espaço em que o número de locais possíveis e de conexões entre os locais é finito.

Baseando-se no objetivo desejado, duas grandes classes de problemas de localização podem ser definidas. Uma classe trata da minimização de algum valor de distância média ou total entre os clientes e os centros de atendimento. O modelo clássico utilizado para representação dos problemas desta classe é o do problema de  $p$ -medianas, que visa selecionar  $p$  vértices em uma rede para a instalação de facilidades de forma a minimizar a soma das distâncias entre os vértices de demanda e a facilidade mais próxima. Modelos que buscam minimizar a distância total ou média são apropriados para descrever problemas do setor privado, no qual medidas de custo estão diretamente relacionadas às distâncias envolvidas no atendimento das demandas. Também se encontram aplicações deste modelo a problemas do setor público. Hillsman (1984) apresenta uma forma de manipular os coeficientes da função objetivo para adequar vários problemas de localização ao modelo do problema de  $p$ -medianas.

A segunda classe de problemas de localização enfoca a distância máxima entre qualquer cliente e a facilidade designada para atendimento. Tais problemas são conhecidos como problemas de cobertura e a distância máxima de atendimento é denominada distância de cobertura ou de serviço ou de atendimento. Toregas *et al.* (1971) apresenta um modelo de cobertura de conjuntos para determinar o número mínimo de centros necessários ao atendimento de todos os vértices de demanda, para uma dada distância de cobertura. Por sua

simplicidade, este modelo não faz distinção da demanda em cada vértice e o número de facilidades necessárias para atendimento de todos os vértices pode ser grande, incorrendo em aumento dos custos fixos de instalação das facilidades. Uma alternativa considera que o número de facilidades a serem instaladas não é suficiente para o atendimento de toda a demanda existente. Neste caso, a restrição de que toda a demanda seja atendida – para uma dada distância de cobertura – é relaxada e procura-se localizar  $p$  facilidades de forma que a configuração de cobertura atenda a maior demanda possível. Este problema é conhecido como o problema de localização de máxima cobertura (Church e ReVelle, 1974). Modelos de cobertura são freqüentemente utilizados por órgãos públicos para a localização de serviços emergenciais ou não-emergenciais.

O problema de  $p$ -medianas é combinatorial e *NP-hard* (Garey e Johnson, 1979), o que limita o tamanho das instâncias possíveis de serem resolvidas por métodos exatos com os recursos computacionais até então disponíveis. Apesar do aumento da velocidade dos processadores e da capacidade de gerenciamento da memória de acesso aleatório por sistemas operacionais modernos, o processo de solução de problemas de grande porte ainda baseia-se no aprimoramento de métodos heurísticos desenvolvidos para a resolução dos mesmos (Teitz e Bart, 1968; Neebe, 1978; Christofides e Beasley, 1982). Técnicas baseadas em relaxação lagrangeana, combinadas a procedimentos de otimização por subgradientes, de um ponto de vista primal-dual, também apresentam grande eficiência na solução deste problema (Galvão e Raggi, 1989; Beasley, 1993; Lorena e Senne, 1999). Senne e Lorena (2000) apresentam a utilização da relaxação lagrangeana/*surrogate* para resolver o problema de  $p$ -medianas. Esta relaxação generaliza a relaxação lagrangeana tradicional, utilizando informação local da restrição *surrogate*, relaxada no sentido lagrangeano, para melhorar o desempenho do método de subgradientes. Uma busca local por valores adequados do multiplicador lagrangeano/*surrogate* é realizada em iterações iniciais do método para corrigir o tamanho dos passos (Narciso e Lorena, 1999).

Atualmente, o tratamento de problemas combinatoriais de grande porte tem recebido grande impulso, graças ao desenvolvimento de pacotes comerciais de otimização mais rápidos e eficientes (ILOG, 2001). Tais ferramentas permitem que problemas inerentemente complexos também possam ser resolvidos em tempo computacional aceitável, através da utilização de técnicas combinadas como, por exemplo, o Método de Geração de Colunas

aplicado a problemas de Programação Inteira. Baseado no trabalho de Dantzig e Wolfe (1960), a primeira aplicação prática desta técnica foi na determinação de padrões de corte unidimensionais (Gilmore e Gomory, 1961; 1963) e, desde então, tem sido explorada em várias aplicações, como o corte de padrões (Vance *et al.*, 1994; Valério de Carvalho, 1999), roteamento de veículos (Desrochers e Soumis, 1989, Desrochers *et al.*, 1992), escala de tripulações (Day e Ryan, 1997; Yunes *et al.*, 2000a; Yunes *et al.*, 2000b) e projetos de circuitos VLSI (Meneses e de Souza, 1998).

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 problema mestre restrito e um subproblema gerador de colunas. A partir de um conjunto inicial de colunas resolve-se o problema mestre, obtendo-se as variáveis duais que modificarão os coeficientes de custo da função objetivo do subproblema, cuja solução corresponde às novas colunas a serem adicionadas ao problema mestre.

Sabe-se 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 (*tailing-off*). Nestes casos, observa-se que as variáveis duais oscilam em torno da solução dual ótima, sendo necessário implementar métodos de estabilização que previnam tal comportamento e possam acelerar a resolução do problema. Senne e Lorena (2001) demonstram a utilização bem sucedida do multiplicador lagrangeano/*surrogate* no processo de geração de colunas para resolver problemas de *p*-medianas. Experimentos computacionais conduzidos com dados da literatura comprovaram que o multiplicador lagrangeano/*surrogate* permite a seleção de colunas mais produtivas, acelerando a convergência dos limitantes e abreviando o processo de geração de colunas, principalmente para instâncias com grande número de vértices. Outras alternativas desenvolvidas para a melhoria da convergência do método são descritas em Neame (1999).

Neste trabalho, será utilizada uma formulação de Programação Inteira de problemas cobertura de conjuntos para representar o problema de localização de *p*-medianas. A utilização direta da técnica de geração de colunas não garante que as soluções obtidas sejam

factíveis para o problema original, sendo necessária a aplicação de um método de busca em árvore binária baseado no Método *Branch-and-Bound* (Nemhauser e Wolsey, 1999). Senne *et al.* (2005) apresentam um método de busca em árvore binária com geração de colunas, conhecido como *branch-and-price* (Barnhart *et al.*, 1998), para obtenção de soluções viáveis para problemas de  $p$ -medianas com até 900 vértices.

As principais contribuições deste trabalho são:

- A integração da relaxação lagrangeana/*surrogate* ao método de geração de colunas, como alternativa de estabilização;
- O desenvolvimento de um algoritmo *branch-and-price* para resolução de problemas de localização de  $p$ -medianas;
- A aplicação do método de geração de colunas para resolver problemas de localização de máxima cobertura;
- O desenvolvimento de estratégias para minimizar os efeitos da esparsidade nos coeficientes de custo e da matriz de restrições que dificultam a aplicação do método de geração de colunas para a obtenção de soluções viáveis ou limitantes para problemas de localização de máxima cobertura.

O trabalho está organizado da seguinte forma:

- CAPÍTULO 2: Apresenta o problema de  $p$ -medianas, suas formulações e a correspondente relaxação lagrangeana/*surrogate*. São estabelecidas algumas relações entre os limitantes inferiores relativos a cada formulação e define-se um algoritmo de busca para o multiplicador lagrangeano/*surrogate* que irá modificar o critério de seleção do método de geração de colunas.
- CAPÍTULO 3: Apresenta a formulação que permite o emprego do método de geração de colunas, obtida após a aplicação da decomposição de Dantzig-Wolfe sobre a formulação tradicional do problema de  $p$ -medianas, e o desenvolvimento da teoria que permite utilizar o multiplicador lagrangeano/*surrogate* para modificar os coeficientes de custo da função objetivo do subproblema gerador de colunas.

Também são apresentados resultados computacionais da aplicação desta técnica a instâncias de  $p$ -medianas com até 3038 vértices.

- CAPÍTULO 4: Descreve o algoritmo *branch-and-price* para obtenção de soluções viáveis para problemas de  $p$ -medianas, destacando aspectos computacionais referentes à obtenção do conjunto inicial de colunas, controle do tamanho do problema e critério de ramificação. Apresenta os resultados de testes computacionais para instâncias de problemas de  $p$ -medianas com até 900 vértices.
- CAPÍTULO 5: Apresenta o problema de máxima cobertura, sua formulação e adaptações necessárias para aplicação do método de geração de colunas discutido neste trabalho. Apresenta os resultados computacionais para dados reais de problemas com até 818 vértices.
- CAPÍTULO 6: Conclusões e perspectivas de trabalhos futuros sobre os estudos realizados.



## CAPÍTULO 2

### O PROBLEMA DE $p$ -MEDIANAS

O problema de  $p$ -medianas é um problema de localização-alocação, ou seja, visa determinar a configuração de custo mínimo de instalação de facilidades e de atendimento da demanda de cada cliente em uma rede conectada por um número finito de caminhos. Geralmente há apenas uma solução ótima para o problema, mas podem ocorrer casos em que existam mais de uma configuração de custo mínimo.

Os dados relevantes para um problema de  $p$ -medianas são:

- Um número finito de pontos, com valores conhecidos de demanda, denominados pontos de demanda;
- Um número finito de locais candidatos para a instalação de facilidades. Neste trabalho, considera-se que os pontos de demanda são candidatos para a instalação de facilidades;
- A distância entre cada ponto de demanda e os locais candidatos. Tais distâncias podem ser calculadas sobre a rede de caminhos que conectam os pontos, ou como distâncias euclidianas;
- O número  $p$  de facilidades a serem instaladas.

#### 2.1 Levantamento Bibliográfico.

Devido à sua importância prática, o problema de  $p$ -medianas apresenta um grande número de aplicações encontradas na literatura. Hakimi (1964; 1965) demonstrou as primeiras aplicações do problema de  $p$ -medianas para determinação de centros de comutação em redes de comunicações. Desde então, diversas alternativas de resolução do problema foram desenvolvidas.

Os algoritmos empregados para resolver problemas de  $p$ -medianas podem ser divididos em 4 grupos: algoritmos primais, algoritmos duais, métodos heurísticos e meta-heurísticas.

Dentre os primeiros algoritmos primais desenvolvidos estão os trabalhos de Järvinen *et al.* (1972) e El-Shaieb (1973), que apresentam algoritmos baseados no método *branch-and-bound* para obter soluções ótimas para o problema de  $p$ -medianas. Também foram desenvolvidas abordagens baseadas em programação linear (ReVelle e Swain, 1970; Swain, 1974; Garfinkel *et al.*, 1974). Entretanto, a qualidade dos limitantes obtidos e a formulação utilizada permitiam o tratamento de instâncias com poucas dezenas de vértices. Jensen (1969) sugere uma abordagem baseada em Programação Dinâmica.

A maioria dos métodos duais baseia-se no uso de técnicas baseadas em relaxação lagrangeana, iniciando-se com os trabalhos de Narula *et al.* (1977) e Cornuejols *et al.* (1977). Outras aplicações desta técnica são encontradas em Mulvey e Crowder (1979), Galvão (1980), Christofides e Beasley (1982), Mirchandani *et al.* (1985), Galvão e Raggi (1989) e Beasley (1993). Senne e Lorena (2000) demonstram a aplicação do método de otimização de subgradientes para resolver a relaxação lagrangeana/*surrogate* de problemas de  $p$ -medianas.

O estudo da aplicação de métodos heurísticos para resolver problemas de  $p$ -medianas iniciou-se com Maranzana (1964). Teitz e Bart (1968) apresentam uma heurística gulosa baseada em substituição de vértices, posteriormente melhorada por Whitaker (1983) e Resende e Werneck (2002a).

Recentemente, tem-se difundido a utilização de meta-heurísticas para o cálculo de soluções para problemas de  $p$ -medianas. Nessas abordagens destaca-se o emprego de heurísticas construtivas (Rosing e ReVelle, 1997), Busca Tabu (Rolland *et al.*, 1996; Mladenovic *et al.*, 1996; Voss, 1996), *Variable Neighborhood Search* (Hansen e Mladenovic, 1997; Hansen *et al.*, 2001), GRASP (Resende e Werneck, 2002b), *Simulated Annealing* (Chiyoshi e Galvão, 2000), Algoritmos Genéticos (Hosage e Goodchild, 1986; Erkut *et al.*, 1997; Correa *et al.*, 2001) e Algoritmo Genético Construtivo (Lorena e Furtado, 2001), dentre outras.

Galvão (2004) apresenta análises de alguns dos métodos empregados, além de outras técnicas desenvolvidas para o tratamento de vários problemas de localização.

## 2.2 Formulações Matemáticas.

Considere um grafo  $G = (N, A)$ , com  $|N| = n$ . O problema de  $p$ -medianas consiste em determinar  $p < n$  vértices (medianas) de modo a minimizar a soma das distâncias dos outros vértices do grafo à mediana mais próxima. A matriz de distâncias  $D = [d_{ij}]_{n \times n}$  entre cada par de vértices deve ser conhecida *a priori*. Uma outra forma de apresentação do problema considera o particionamento de um conjunto  $N$  em  $p$  *clusters* (agrupamentos) disjuntos, minimizando a diferença entre os elementos de cada *cluster*. Por esse motivo, problemas de  $p$ -medianas pertencem à classe de problemas de agrupamento (Vinod, 1969; Rao, 1971; Hansen e Jaumard, 1997; Fung e Mangasarian, 2000).

### 2.2.1 Problema Inteiro Binário.

O problema de  $p$ -medianas considerado neste trabalho pode ser formulado como o seguinte problema de otimização:

$$PPM \quad v(PPM) = \text{Min} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij} \quad (2.1)$$

$$\text{s. a.} \quad \sum_{j \in N} x_{ij} = 1, \quad \forall i \in N \quad (2.2)$$

$$\sum_{j \in N} x_{jj} = p \quad (2.3)$$

$$x_{ij} \leq x_{jj}, \quad \forall i, j \in N \quad (2.4)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in N \quad (2.5)$$

onde  $[x_{ij}]_{n \times n}$  é uma matriz de alocação, com  $x_{ij} = 1$  se o vértice  $i$  está alocado à mediana  $j$ , e  $x_{ij} = 0$ , caso contrário;  $x_{jj} = 1$  se o vértice  $j$  é uma mediana e  $x_{jj} = 0$ , caso contrário.

A função objetivo (2.1) informa a distância total correspondente a uma solução de localização-alocação. As restrições (2.2) e (2.4) garantem que cada vértice  $i$  seja alocado a apenas um vértice  $j$ , que deve ser uma mediana. A restrição (2.3) determina o número de medianas a serem localizadas e a restrição (2.5) impõe a condição de integralidade sobre as variáveis.

### 2.2.2 Problema de Particionamento de Conjuntos.

O algoritmo *branch-and-price* apresentado neste trabalho para resolver problemas de  $p$ -medianas considera a aplicação da decomposição de Dantzig-Wolfe sobre a formulação *PPM*. As primeiras aplicações desta técnica tinham por objetivo obter problemas lineares com menor número de restrições (Swain, 1974; Garfinkel *et al.*, 1974). Galvão (1981) destaca que a nova formulação favorece a ocorrência de degeneração, prejudicando a convergência de métodos baseados em Programação Linear empregados em sua resolução.

Sendo  $S = \text{Pot}(N) = \{S_1, S_2, \dots, S_m\}$  o conjunto de todos os subconjuntos de  $N$ , Minoux (1987) apresenta o problema de  $p$ -medianas usando a seguinte formulação de um problema de particionamento de conjuntos com restrição de cardinalidade:

$$PPC \quad v(PPC) = \text{Min} \sum_{k \in M} c_k x_k \quad (2.6)$$

$$\text{s. a.} \quad \sum_{k \in M} A^k x_k = 1 \quad (2.7)$$

$$\sum_{k \in M} x_k = p \quad (2.8)$$

$$x_k \in \{0, 1\}, \quad \forall k \in M \quad (2.9)$$

onde:

- $M = \{1, 2, \dots, m\}$ ;
- $c_k = \text{Min}_{j \in S_k} \left\{ \sum_{i \in S_k} d_{ij} \right\}, \forall k \in M$ ;
- $A^k = [a_{ik}]_{n \times 1}$ , com  $a_{ik} = 1$  se o vértice  $i \in S_k$ ;  $a_{ik} = 0$ , caso contrário;
- $x_k = 1$  se o subconjunto (*cluster*)  $S_k \in S$  for escolhido;  $x_k = 0$ , caso contrário.

Cada subconjunto  $S_k$  corresponde a uma coluna  $A^k$  do conjunto de restrições (2.7), representando um *cluster* cuja mediana  $j \in S_k$  é determinada quando o custo  $c_k$  é calculado. Dessa forma, a restrição (2.4) de *PPM* é considerada implicitamente. No cálculo de  $c_k$ ,

$\forall k \in M$ , o nó  $j \in S_k$  que corresponder ao menor valor de distância total aos outros nós  $i \in S_k$  será escolhido como mediana, e o valor da distância total calculada corresponderá ao custo do cluster  $k$ . As restrições (2.2) e (2.3) são conservadas e atualizadas para (2.7) e (2.8), respectivamente.

Exemplo: Para ilustrar a aplicação da formulação *PPC* para descrever problemas de  $p$ -medianas, considere o conjunto de vértices com índices dados por  $N = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ , e defina  $d_{ij} = |i - j|$ . Sejam  $S_{k_1} = \{2, 3, 5, 6, 8\}$ ,  $S_{k_2} = \{1, 2, 4, 7\}$  e  $S_{k_3} = \{5, 6, 9\}$  elementos de  $S$ . As colunas correspondentes aos *clusters*  $k_1, k_2$  e  $k_3 \in M$  e seus respectivos custos são:

- $A^{k_1} = [0, 1, 1, 0, 1, 1, 0, 1, 0]^T$ ,  $c_{k_1} = 9$  (vértice 5 é a mediana).
- $A^{k_2} = [1, 1, 0, 1, 0, 0, 1, 0, 0]^T$ ,  $c_{k_2} = 8$  (vértice 2 ou 4 é a mediana).
- $A^{k_3} = [0, 0, 0, 0, 1, 1, 0, 0, 1]^T$ ,  $c_{k_3} = 4$  (vértice 6 é a mediana).

Caso haja mais de um vértice que corresponda ao menor valor de custo de um dado subconjunto  $S_k$ , um critério de desempate seria definir como mediana o vértice  $j \in N$  de menor índice, dentre os possíveis candidatos. Assim, no exemplo acima, a mediana do *cluster*  $k_2$  será o vértice 2.

As soluções viáveis para o *PPC* são formadas por combinações de  $p$  colunas  $A^k$ , com cada vértice pertencendo a apenas um subconjunto  $S_k$ , segundo a restrição (2.7). No contexto de problemas de localização isto significa que cada vértice de demanda deve ser atendido por apenas uma facilidade, característica que determina a viabilidade das soluções de problemas de  $p$ -medianas, mas dificulta a obtenção das mesmas.

Entretanto, os coeficientes positivos da matriz de distâncias  $[d_{ij}]_{n \times n}$  permitem utilizar uma versão relaxada da formulação *PPC*, onde a restrição (2.7) é substituída por:

$$\sum_{k \in M} A^k x_k \geq 1 \quad (2.10)$$

mantendo a viabilidade das soluções obtidas para o problema de  $p$ -medianas.

Esta afirmação é facilmente verificada ao se considerar que, para uma dada solução viável  $x_k$ , se um vértice está presente em duas colunas simultaneamente, ele pode ser removido de uma delas e o custo resultante da nova coluna será menor que o custo da coluna original, reduzindo assim o valor da função objetivo (2.6).

### 2.3 Relaxação Lagrangeana/Surrogate.

A relaxação lagrangeana/surrogate foi proposta como uma alternativa à relaxação lagrangeana tradicional, para utilização em algoritmos de otimização de subgradientes. Senne e Lorena (2000) verificaram a eficiência desta técnica quando aplicada a problemas de  $p$ -medianas.

Para um dado vetor de multiplicadores  $\lambda \in R^n$ , propõe-se substituir as  $n$  restrições (2.2) em  $PPM$  por uma única restrição, denominada restrição *surrogate* (Glover, 1968). A formulação resultante corresponde à relaxação *surrogate* para o  $PPM$  é dada por:

$$S_\lambda PPM \quad v(S_\lambda PPM) = \text{Min} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij}$$

$$\text{s. a.} \quad \sum_{i \in N} \sum_{j \in N} \lambda_i x_{ij} = \sum_{i \in N} \lambda_i \quad (2.11)$$

e (2.3) – (2.5).

O valor ótimo  $v(S_\lambda PPM)$  é menor ou igual a  $v(PPM)$ , e resulta da resolução do problema dual *surrogate*  $\text{Max}_\lambda \{v(S_\lambda PPM)\}$ . O problema  $S_\lambda PPM$  é um problema linear inteiro sem qualquer característica que possa ser explorada. Além disso, a função *surrogate*  $s: R^n \rightarrow R$ ,  $(\lambda, v(S_\lambda PPM))$  possui algumas propriedades que dificultam a obtenção da solução dual. Métodos para encontrar soluções aproximadas para o problema dual *surrogate* foram propostos por Karwan e Rardin (1979) e Dyer (1980).

Devido a tais dificuldades com a relaxação  $S_\lambda PPM$ , propõe-se relaxar novamente o problema, agora no sentido lagrangeano. Seja  $t \in R$  o multiplicador dual associado à restrição (2.11), obtendo-se assim a relaxação lagrangeana/surrogate de  $PPM$  dada por:

$L_t S_\lambda PPM$

$$v(L_t S_\lambda PPM) = \text{Min} \sum_{i \in N} \sum_{j \in N} (d_{ij} - t\lambda_i)x_{ij} + t \sum_{i \in N} \lambda_i$$

s. a. (2.3) – (2.5).

Para  $t$  e  $\lambda$  dados, tem-se  $v(L_t S_\lambda PPM) \leq v(S_\lambda PPM) \leq v(PPM)$ . Uma característica interessante da relaxação  $L_t S_\lambda PPM$  é que para  $t = 1$  tem-se a relaxação lagrangeana tradicional, no multiplicador  $\lambda$ . Narciso (1998) demonstra que, para  $t^*$  obtido como solução ótima do dual local do problema  $L_t S_\lambda PPM$ , o valor  $v(L_{t^*} S_\lambda PPM)$  fornece um limitante melhor que o obtido pela relaxação lagrangeana usual. A próxima seção apresenta um algoritmo que fornece soluções aproximadas para  $t^*$ .

Considerando-se implicitamente a restrição (2.3), o problema  $L_t S_\lambda PPM$  é resolvido por decomposição sobre o índice  $j$ , obtendo-se  $n$  subproblemas do tipo:

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

s. a. (2.4) e (2.5).

Cada subproblema é facilmente resolvido calculando-se:

$$\beta_j = \sum_{i \in N} \min\{0, d_{ij} - t\lambda_i\}, \quad \forall j \in N$$

e definindo-se  $J$  como o conjunto dos índices  $j \in N$  que correspondem aos  $p$  menores valores de  $\beta_j$  (para satisfazer a restrição (2.3)) Assim, uma solução  $x^\lambda$  para  $L_t S_\lambda PPM$  pode ser obtida fazendo-se:

$$x_{ij}^\lambda = \begin{cases} 1, & \text{se } j \in J \\ 0, & \text{caso contrário} \end{cases} \quad (2.12)$$

e para  $i \neq j$ :

$$x_{ij}^\lambda = \begin{cases} 1, & \text{se } j \in J \text{ e } d_{ij} - t\lambda_i < 0 \\ 0, & \text{caso contrário} \end{cases} \quad (2.13)$$

e o valor de  $L_t S_\lambda PPM$  será:

$$v(L_t S_\lambda PPM) = \sum_{j \in N} \beta_j x_{jj}^\lambda + t \sum_{i \in N} \lambda_i$$

### 2.3.1 Cálculo do Multiplicador Lagrangeano/Surrogate.

Para  $\lambda \in R^n$  dado, o melhor valor de  $t$  é obtido como solução ótima do dual do problema  $L_t S_\lambda PPM$ , definido como:

$$D_{t,\lambda} \quad v(D_{t,\lambda}) = \text{Max}_t \{v(L_t S_\lambda PPM)\},$$

ou por busca dicotômica, dado que a função lagrangeana  $l: R \rightarrow R, (t, v(L_t S_\lambda PPM))$ , é côncava e linear por partes (Parker e Rardin, 1988) (FIGURA 2.1).

O algoritmo da FIGURA 2.2 foi aplicado por Senne e Lorena (2000), para obtenção de valores aproximados para o melhor valor de  $t$ .

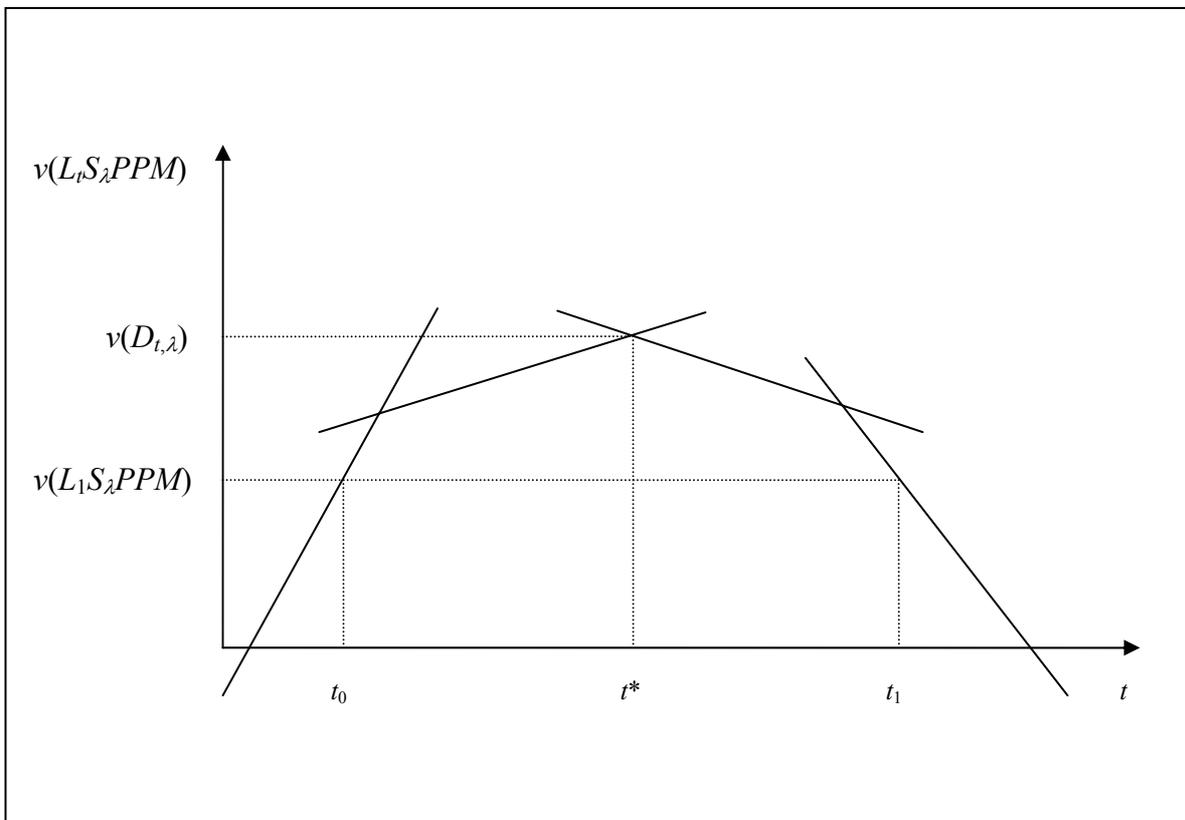


FIGURA 2.1 - Limitantes lagrangeano/surrogate.

**Algoritmo BM:**

Defina o valor inicial  $t_0$  e o tamanho do passo  $s$ ;

Faça  $t \leftarrow t_0$ ,  $a \leftarrow$  indefinido;  $b \leftarrow$  indefinido;

Enquanto ( $a =$  indefinido ou  $b =$  indefinido) repita:

Resolva  $L_t S_\lambda PPM$  e defina  $x^\lambda$  conforme (2.12) e (2.13);

$$\text{Faça } slope^\lambda \leftarrow \sum_{i \in N} \lambda_i \left( 1 - \sum_{j \in N} x_{ij}^\lambda \right);$$

Se  $slope^\lambda > 0$ , faça  $a \leftarrow t$  e  $t \leftarrow t + s$ ;

Senão, faça  $b \leftarrow t$  e  $t \leftarrow t - s$ ;

Encontre o melhor de valor de  $t \in (a, b)$  por busca dicotômica, atualizando o tamanho do passo  $s$ , se necessário.

FIGURA 2.2 - Algoritmo de busca do multiplicador lagrangeano/*surrogate*.

Testes computacionais comprovaram que a obtenção de bons limitantes para o *PPM* não depende do cálculo exato de  $t^*$ , sendo suficiente encontrar um valor  $t \in [t_0, t_1]$ . Se o valor de  $t$  permanecer inalterado por um dado número de iterações, fixo *a priori*, então este valor é mantido até o final do algoritmo principal e o procedimento de busca não é mais executado.



## CAPÍTULO 3

### APLICAÇÃO DA RELAXAÇÃO LAGRANGEANA/*SURROGATE* AO MÉTODO DE GERAÇÃO DE COLUNAS

A aplicação da decomposição de Dantzig-Wolfe à formulação *PPM* permite reescrever o conjunto de soluções factíveis do problema como função de seus raios extremos. Do ponto de vista de um problema de particionamento de conjuntos, procura-se escolher  $p$  subconjuntos de  $S = \{S_1, S_2, \dots, S_m\}$  que representem uma solução viável para o problema de  $p$ -medianas. A combinação dos  $p$  subconjuntos de menor custo é a solução ótima.

Na prática,  $|S|$  pode ser muito grande. Para um problema com apenas 10 vértices, o conjunto  $S$  correspondente terá  $2^{10}$  elementos. A enumeração completa de todos os subconjuntos  $S_k \in S$  para instâncias com um grande número de vértices torna-se uma tarefa difícil, sob o aspecto computacional. A matriz de restrições da formulação *PPC* seria composta por um número de colunas igual à  $|S|$ , e a resolução de problemas com grande número de variáveis apresenta um verdadeiro desafio, mesmo para os computadores mais potentes disponíveis.

A técnica de geração de colunas permite trabalhar com um conjunto bastante reduzido de colunas, e novas colunas são calculadas como solução de um subproblema, utilizando informação local das variáveis duais de um problema mestre. Considerando um problema de minimização (maximização), a coluna de custo reduzido mais negativo (positivo) é adicionada à matriz de restrições, definindo um novo problema mestre, cuja resolução fornecerá uma nova solução dual que será utilizada para modificar os coeficientes de custo do subproblema gerador de colunas. O processo continua até que não haja mais colunas com custo reduzido negativo (positivo) a serem incluídas na matriz de restrições. Visando acelerar o processo de resolução do problema, podem-se adicionar todas as colunas de custo reduzido negativo (positivo) obtidas da resolução do subproblema gerador em cada iteração. Esta estratégia é denominada *multi-pricing*.

Dentre os trabalhos recentes que descrevem esta técnica e algumas aplicações estão Wilhelm (2001), Lübbecke e Desrosiers (2002) e Desrosiers e Lübbecke (2004).

Lorena *et al.* (2003) demonstram a equivalência entre o método de geração de colunas e o método de Planos de Corte (Kelley, 1960), apresentando alguns resultados teóricos relativos à qualidade dos limitantes da relaxação lagrangeana/*surrogate*, juntamente com aplicações do uso combinado do método de geração de colunas e da relaxação lagrangeana/*surrogate*. Senne e Lorena (2001) utilizam a relaxação lagrangeana/*surrogate* na resolução de problemas de  $p$ -medianas pelo método de geração de colunas.

### 3.1 O Problema Mestre Restrito.

A resolução de problemas lineares de grande porte pelo método de geração de colunas é um processo iterativo onde, partindo-se de um subconjunto inicial de colunas, novas colunas são inseridas na formulação de um problema mestre, a cada iteração. Um problema mestre restrito (PMR) para resolver o problema de  $p$ -medianas pode ser definido, considerando-se um subconjunto  $K \subset M = \{1, 2, \dots, m\}$  dos índices das colunas da formulação completa do  $PPC$ , como a seguinte formulação de relaxação de programação linear de um problema de cobertura de conjuntos com restrição de cardinalidade:

$$\overline{PCC} \quad v(\overline{PCC}) = \text{Min} \sum_{k \in K} c_k x_k \quad (3.1)$$

$$\text{s. a.} \quad \sum_{k \in K} A^k x_k \geq 1 \quad (3.2)$$

$$\sum_{k \in K} x_k = p \quad (3.3)$$

$$x_k \in [0, 1] \quad \forall k \in K \quad (3.4)$$

A solução ótima de  $\overline{PCC}$  pode não ser viável para o  $PPC$ , pois as variáveis  $x_k$  podem assumir valores fracionários. Neste caso, a aplicação de métodos tipo *branch-and-bound* podem produzir soluções viáveis para o  $PPC$ , mas o conjunto final de colunas do  $\overline{PCC}$  pode não ser suficiente para obtenção da solução ótima.

Entretanto, as soluções duais ótimas  $\lambda \in R_+^n$  e  $\mu \in R$ , associadas às restrições (3.2) e (3.3) respectivamente, podem ser utilizadas para o cálculo de novas colunas, como será visto na seção 3.2.

### 3.1.1 O Conjunto Inicial de Colunas.

A FIGURA 3.1 apresenta um procedimento para o cálculo de um conjunto inicial de colunas viáveis para o PMR, visando o desenvolvimento do algoritmo de geração de colunas apresentado neste trabalho. O desenvolvimento de procedimentos para cálculo do conjunto inicial de colunas deve considerar as características da aplicação em estudo. Alguns fatores relevantes que podem determinar o sucesso ou fracasso na resolução do PMR são:

- O número inicial de colunas;
- A robustez do procedimento de geração de colunas (existência de componentes aleatórios);
- Características dos dados (por exemplo, o domínio de definição de coeficientes da matriz de distâncias).

#### Rotina CI:

Defina  $MaxC$  como o número máximo de colunas a serem geradas;

Faça  $NumC \leftarrow 0$ ;

Repita

Seja  $P = \{n_1, \dots, n_p\} \subset N$  um conjunto de vértices escolhidos aleatoriamente;

Para  $j = 1, \dots, p$  faça

$$S_j \leftarrow \{n_j\} \cup \{q \in N - P \mid d_{qn_j} = \min_{t \in P} \{d_{qt}\}\}$$

$$c_j \leftarrow \text{Min}_{i \in S_j} \left\{ \sum_{i \in S_j} d_{it} \right\}$$

Para  $i = 1, \dots, n$  faça

Se  $i \in S_j$ , faça  $a_{ij} \leftarrow 1$ ;

Se  $i \notin S_j$ , faça  $a_{ij} \leftarrow 0$ ;

Adicione a coluna  $\begin{bmatrix} A^j \\ 1 \end{bmatrix}$  ao conjunto inicial de colunas;

$NumC \leftarrow NumC + p$ ;

Enquanto ( $NumC < MaxC$ );

Fim

FIGURA 3.1 - Rotina de geração do conjunto inicial de colunas.

Exemplo: Para ilustrar uma iteração da rotina de geração de colunas iniciais, considere o conjunto  $N$  utilizado no exemplo da Seção 2.2.2, assumindo a localização de três medianas ( $p = 3$ ). Considerando as medianas obtidas anteriormente, define-se  $P = \{2, 5, 6\}$ . Para a medida de distância utilizada anteriormente, os subconjuntos  $S_j$  e as colunas  $A^j$  resultantes,  $j = 1, \dots, p$ , com seus respectivos custos, serão:

- $S_1 = \{1, 2, 3\}, c_1 = 2 \rightarrow A^1 = [1, 1, 1, 0, 0, 0, 0, 0, 0]^T$ ;
- $S_2 = \{4, 5\}, c_2 = 1 \rightarrow A^2 = [0, 0, 0, 1, 1, 0, 0, 0, 0]^T$ ;
- $S_3 = \{6, 7, 8, 9\}, c_3 = 6 \rightarrow A^3 = [0, 0, 0, 0, 0, 1, 1, 1, 1]^T$ .

Um novo conjunto de  $p$  colunas é adicionado a cada iteração, para cada conjunto  $P$ . É importante destacar que o conjunto inicial de colunas deve conter uma base viável para o problema linear inicial.

### 3.2 O Subproblema Gerador de Colunas.

A relaxação lagrangeana/*surrogate* é integrada ao processo de geração de colunas pela utilização dos multiplicadores duais ótimos  $\lambda_i, \forall i \in N$ , do problema  $\overline{PCC}$ , para resolver o problema  $D_{t,\lambda}$  e obter soluções aproximadas para o multiplicador lagrangeano/*surrogate*  $t$ . A mediana escolhida como centro do *cluster* de menor contribuição ao valor de  $v(D_{t,\lambda})$  corresponde ao vértice  $j^*$  obtido como solução ótima do subproblema:

$$SGC_t \quad v(SGC_t) = \underset{j \in N}{\text{Min}} \left\{ \underset{a_{ij} \in \{0,1\}}{\text{Min}} \sum_{i \in N} (d_{ij} - t\lambda_i) a_{ij} \right\}$$

O problema  $SGC_t$  é facilmente resolvido por inspeção, considerando-se cada vértice  $j \in N$  como mediana e fixando-se  $a_{ij}, \forall i \in N$ , da seguinte forma:

$$a_{ij} = \begin{cases} 1, & \text{se } d_{ij} - t\lambda_i \leq 0. \\ 0, & \text{se } d_{ij} - t\lambda_i > 0. \end{cases}$$

Sendo  $j^*$  o índice do vértice que resultar no menor valor para  $v(SGC_t)$ , define-se um novo subconjunto  $S_{j^*}$  como:

$$S_{j^*} = \{i \in N \mid a_{ij^*} = 1\}$$

e a coluna  $\begin{bmatrix} A^{j^*} \\ 1 \end{bmatrix}$  será incluída em  $\overline{PCC}$  se:

$$\sum_{i \in N} (d_{ij^*} - \lambda_i) a_{ij^*} < \mu \quad (3.5)$$

Na prática, todas as colunas  $\begin{bmatrix} A^j \\ 1 \end{bmatrix}$ ,  $j \in N$ , que satisfizerem a condição (3.5) podem ser adicionadas ao PMR (*multi-pricing*), permitindo acelerar o processo de resolução do problema por métodos de geração de colunas.

### 3.2.1 Gerenciamento do Tamanho do Problema.

É importante destacar que a inclusão de novas colunas à formulação do PMR aumenta o tamanho do problema, podendo exigir mais recursos computacionais que os disponíveis para sua resolução. Alguns pacotes comerciais disponibilizam versões de demonstração capazes de resolver problemas com até 300 variáveis contínuas (GAMS, 1998), um valor muito modesto para ser considerado até mesmo em PMRs correspondentes a instâncias de problemas de  $p$ -medianas de pequeno porte. Versões completas do CPLEX (ILOG, 2001) utilizam toda a memória física que o sistema operacional puder gerenciar (2 GB, no caso do Microsoft Windows XP Professional), não sendo raro casos em que a execução do aplicativo é interrompida por falta de memória.

Uma alternativa para contornar tais dificuldades consiste na inclusão de mecanismos de controle do tamanho do problema, baseados em remoção de colunas. Tal procedimento pode ser executado sempre que o problema apresentar um número de colunas maior que um valor estabelecido *a priori*, ou sempre que se desejar eliminar da formulação as colunas com custo reduzido elevado, quando comparadas com um valor médio de referência (FIGURA 3.2).

Entretanto, a utilização constante desse procedimento também pode ser prejudicial para o processo de resolução do problema, uma vez que remover colunas corresponde a retirar informação relevante da formulação.

**Rotina RC:**

Defina  $TotC$  como o número total de colunas de  $\overline{PCC}$ ;

Defina  $fator\_rc$  como um número positivo;

Defina  $cr\_médio$  como o custo reduzido médio das colunas do conjunto inicial;

Obtenha  $cr_j, j = 1, \dots, TotC$ , o custo reduzido de cada coluna  $j$  de  $\overline{PCC}$ ;

Para  $j = 1, \dots, TotC$  faça

Se  $cr_j > fator\_rc * cr\_médio$ , remova a coluna  $j$  de  $\overline{PCC}$ ;

Fim

FIGURA 3.2 - Rotina de eliminação de colunas.

O parâmetro  $fator\_rc$  permite controlar a intensidade do teste aplicado para a remoção de colunas, limitando o número de colunas consideradas úteis para o problema.

### 3.3 O Algoritmo de Geração de Colunas.

A FIGURA 3.3 apresenta o esquema geral do algoritmo de geração de colunas com a relaxação lagrangeana/*surrogate*.

A cada iteração, as variáveis duais ótimas do PMR são utilizadas para calcular novas colunas como solução de um subproblema. Tais variáveis também são utilizadas para calcular limitantes inferiores, obtidos como o valor da solução ótima do problema  $L_t S_\lambda PPM$ , onde  $t=1$  para o caso lagrangeano tradicional e para o caso lagrangeano/*surrogate*  $t$  é obtido como solução aproximada do problema  $D_{t,\lambda}$ .

Nesta implementação, o valor ótimo da função objetivo do PMR,  $v(\overline{PCC})$ , é utilizado para a atualização dos limitantes superiores. Caso os limitantes atinjam a precisão desejada, o processo é terminado. Caso contrário, se as novas colunas obtidas apresentam custos reduzidos adequados, elas são adicionadas ao PMR a ser resolvido na próxima iteração.

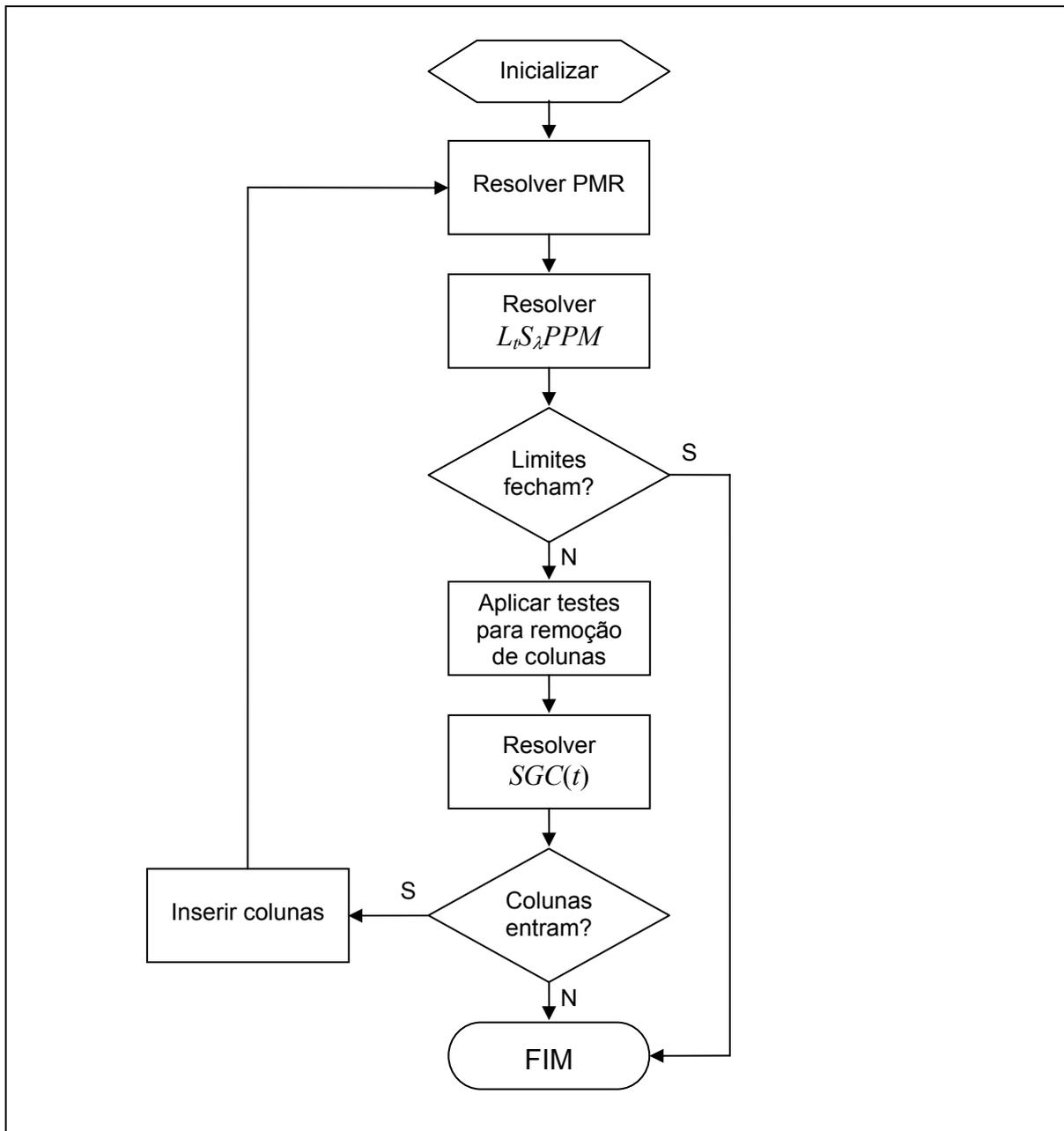


FIGURA 3.3 - Esquema geral do algoritmo de geração de colunas.

O algoritmo apresentado na FIGURA 3.4 foi implementado para obtenção de limitantes inferiores para o problema de  $p$ -medianas, utilizando o critério de custos reduzidos modificados pelo multiplicador lagrangeano/*surrogate* proposto na seção 3.2. O PMR inicial é definido pela formulação correspondente ao  $\overline{PCC}$  composta pelas colunas obtidas pela execução da rotina apresentada na FIGURA 3.1. Para o caso lagrangeano tradicional não é necessário resolver  $D_{t,\lambda}$ , pois neste caso  $t$  é fixado em 1.

**Algoritmo GC(t):**

Faça *condição* ← TRUE;

Enquanto (*condição* = TRUE) faça

    Resolva  $\overline{PCC}$  utilizando CPLEX e obtenha os valores ótimos das variáveis duais  $\lambda$  e  $\mu$ ;

    Obtenha uma solução aproximada para  $D_{t,\lambda}$ ;

    Se  $|v(\overline{PCC}) - v(L_{t,S,\lambda}PPM)| < \varepsilon$ , faça *condição* ← FALSE;

    Senão

        Execute testes para remoção de colunas e aplique a rotina **RC**, se necessário;

        Resolva o problema  $SGC_t$  e adicione à  $\overline{PCC}$  as colunas  $\begin{bmatrix} A^j \\ 1 \end{bmatrix}$  que satisfizerem (3.5);

        Se não houver tais colunas faça *condição* ← FALSE;

    Fim;

Fim

FIGURA 3.4 - Algoritmo de geração de colunas.

Convém destacar que o mesmo conjunto de colunas foi utilizado para definir o PMR inicial, tanto para o caso lagrangeano tradicional quanto para o caso lagrangeano/*surrogate*.

### 3.4 Resultados Computacionais.

Foram realizados testes computacionais para avaliar a aplicação da relaxação lagrangeana/*surrogate* para resolver instâncias de problemas de *p*-medianas obtidos da *OR-Library* (Beasley, 1990), pelo método de geração de colunas.

O algoritmo e as rotinas apresentadas neste capítulo foram implementados em linguagem C e foram executados em uma estação Sun Ultra 30, com processador UltraSPARC-II de 296 MHz e 384 MB de RAM, executando o sistema operacional SunOS 5.6. Foi usado o aplicativo comercial ILOG CPLEX versão 6.5.

A TABELA 3.1 apresenta os valores obtidos com a aplicação do algoritmo  $GC_t$  e do algoritmo  $GC_1$  (valores mostrados entre parênteses). O número máximo de colunas a serem geradas para o PMR inicial (*MaxC*) é 2000. O parâmetro *fator\_rc* foi fixado em 1 e o número máximo de iterações é 1000.

A legenda utilizada na TABELA 3.1 é dada a seguir:

- $n$ : número de vértices da rede;
- $p$ : número de medianas;
- **solução ótima**: valor da solução ótima da instância;
- **iterações**: número de iterações;
- **colunas geradas**: número total de colunas geradas;
- **colunas usadas**: número efetivo de colunas aproveitadas;
- **gap primal**:  $100 \times |\nu(\overline{PCC}) - \text{solução ótima}| / \text{solução ótima}$ ;
- **gap dual**:  $100 \times |\text{solução ótima} - \nu(L_\nu S_\lambda PPM)| / \text{solução ótima}$ ;
- **tempo**: tempo computacional total (em segundos).

O símbolo “–” representa *gap* (primal ou dual) nulo e os tempos apresentados não consideram o tempo gasto na preparação das instâncias, como por exemplo, a geração do conjunto de colunas do respectivo PMR inicial.

TABELA 3.1 - Resultados obtidos para instâncias da *OR-Library*.

$n$	$p$	solução ótima	iterações	colunas geradas	colunas usadas	gap primal	gap dual	tempo
100	5	5819	184 (155)	5458 (5969)	3861 (3775)	– (–)	– (–)	36,35 (36,31)
200	5	7824	399 (381)	16929 (23630)	11763 (12533)	– (–)	– (–)	902,77 (1625,63)
200	10	5631	936 (757)	24375 (24483)	20584 (18701)	– (–)	– (–)	996,00 (864,83)
300	5	7696	1000 (919)	39299 (48431)	38173 (42704)	0,246 (–)	1,796 (–)	17889,12 (23337,79)
300	10	6634	731 (1000)	33342 (55200)	26638 (36864)	– (0,108)	– (0,215)	10749,91 (13214,36)
300	30	4374	198 (1000)	12040 (40166)	8016 (30381)	– (–)	– (0,118)	831,22 (1057,43)
400	5	8162	1000 (1000)	60624 (85762)	53181 (64266)	0,686 (0,832)	1,662 (1,022)	52807,93 (83877,77)
400	10	6999	675 (627)	41156 (66680)	26561 (26070)	– (–)	– (–)	36829,25 (41202,98)
400	40	4809	195 (191)	18160 (24213)	13130 (13101)	– (–)	– (–)	1055,20 (1078,27)

Os resultados obtidos demonstram que a integração da relaxação lagrangeana/*surrogate* no algoritmo de geração de colunas permitiu encontrar os mesmos resultados que os obtidos com a relaxação lagrangeana tradicional, porém gerando menos colunas. Também se verificou que para um dado número de vértices, quanto menor for o número de medianas da instância, mais difícil torna-se a obtenção de soluções pelo método de geração de colunas.

Christofides e Beasley (1982) discutem a influência da relação entre o número de vértices e o número de medianas a serem localizadas ( $n/p$ ) no desempenho de métodos baseados em otimização de subgradientes aplicados a abordagens lagrangeanas para resolução de problemas de  $p$ -medianas. Instâncias onde essa relação é igual a 3 geralmente apresentam mais dificuldades de resolução. Galvão e Raggi (1989) propõem que outros fatores – como valores de distância gerados aleatoriamente – contribuem para o aumento da complexidade de métodos empregados para a resolução de problemas de  $p$ -medianas.

A TABELA 3.2 apresenta uma comparação entre os resultados obtidos com a aplicação do algoritmo de subgradientes (*SG*) proposto por Senne e Lorena (2000) e do algoritmo de geração de colunas apresentado neste trabalho, em instâncias da *OR-Library* consideradas difíceis para abordagens de subgradientes.

O número máximo de iterações foi fixado em 50 e *fator\_rc* foi fixado em 1. Para o algoritmo de otimização de subgradientes, uma busca local é executada em cada cluster para encontrar medianas que forneçam soluções de menor valor, permitindo calcular:

- **gap primal:**  $100 \times |v(\text{melhor solução}) - \text{ótima}| / \text{ótima}$ ;
- **gap dual:**  $100 \times |\text{ótima} - v(L_r S_\lambda PPM)| / \text{ótima}$ ;

Os valores apresentados entre parênteses foram obtidos para  $t = 1$ . O símbolo “–” representa *gap* (primal ou dual) nulo.

O algoritmo  $GC(t)$  foi capaz de produzir resultados de qualidade comparável aos obtidos pelo algoritmo *SG*, em tempo computacional consideravelmente menor. Para estas instâncias, os resultados confirmam a superioridade da utilização da relaxação lagrangeana/*surrogate* em métodos de geração de colunas comparada com a utilização da mesma relaxação em métodos de otimização de subgradientes.

TABELA 3.2 - Comparação entre  $SG$  e  $GC(t)$ .

$n$	$p$	solução ótima	gap primal		gap dual		tempo	
			SG	GC	SG	GC	SG	GC
100	33	1355	-	- (-)	-	- (-)	0,58	0,37 (0,35)
200	67	1255	-	- (-)	-	- (0,667)	4,00	1,29 (1,89)
300	100	1729	-	0,116 (-)	-	0,058 (-)	16,78	4,55 (4,90)
400	133	1789	-	0,112 (-)	-	0,950 (0,783)	51,80	6,21 (6,04)
500	167	1828	-	0,055 (0,036)	-	0,310 (0,210)	127,60	11,00 (12,91)
600	200	1989	-	0,302 (0,101)	-	0,285 (0,235)	257,02	15,81 (17,59)
700	233	1847	-	0,081 (0,325)	-	0,379 (0,785)	482,97	21,50 (21,41)
800	267	2026	-	0,518 (0,222)	-	0,346 (0,271)	1374,74	26,14 (27,95)
900	300	2106	0,047	0,518 (0,607)	0,004	0,827 (0,443)	3058,65	33,37 (49,99)

As colunas obtidas como solução do algoritmo  $GC(t)$  contém informação de melhor qualidade, proporcionando a redução do tempo computacional envolvido. Isto pode ser comprovado ao se utilizar um critério de remoção de colunas mais forte. A TABELA 3.3 apresenta os resultados obtidos com a variação de  $fator\_rc$  para a instância  $n = 200$  e  $p = 5$ .

TABELA 3.3 - Influência do parâmetro  $fator\_rc$ .

$fator\_rc$	iterações	colunas geradas	colunas usadas	gap primal	gap dual	tempo
0,5	403 (487)	18493 (47634)	7543 (7364)	- (-)	- (-)	619,63 (971,59)
0,4	414 (1000)	20395 (167247)	6627 (3270)	- (0,631)	- (4,635)	613,79 (1370,99)
0,3	400 (1000)	23521 (186267)	3886 (421)	0,276 (11,171)	2,010 (65,181)	532,27 (905,67)

As FIGURAS 3.5 a 3.7 apresentam a evolução do valor ótimo dos PMRs para a instância  $n = 200$  e  $p = 5$ . A curva mais escura corresponde aos valores obtidos para o caso lagrangeano e a curva mais clara apresenta os valores obtidos para o caso lagrangeano/surrogate.

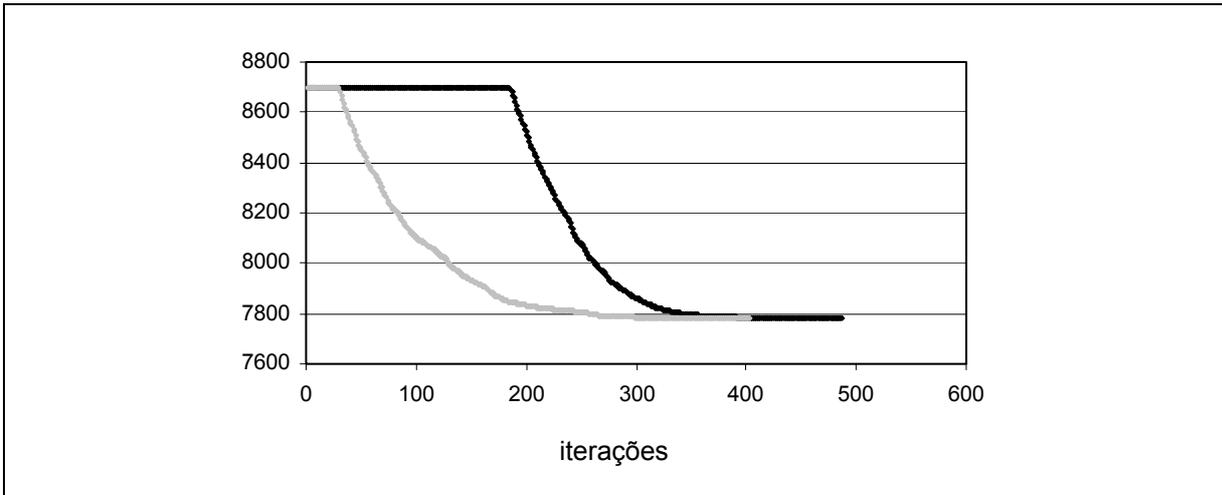


FIGURA 3.5 - Evolução de  $\nu(\overline{PCC})$  para  $fator\_rc = 0,5$ .

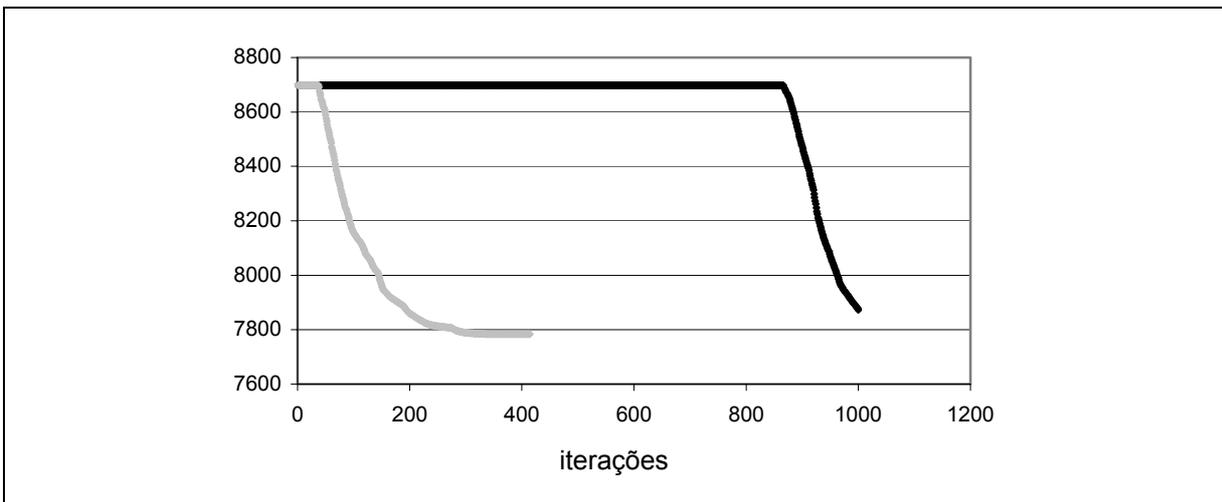


FIGURA 3.6 - Evolução de  $\nu(\overline{PCC})$  para  $fator\_rc = 0,4$ .

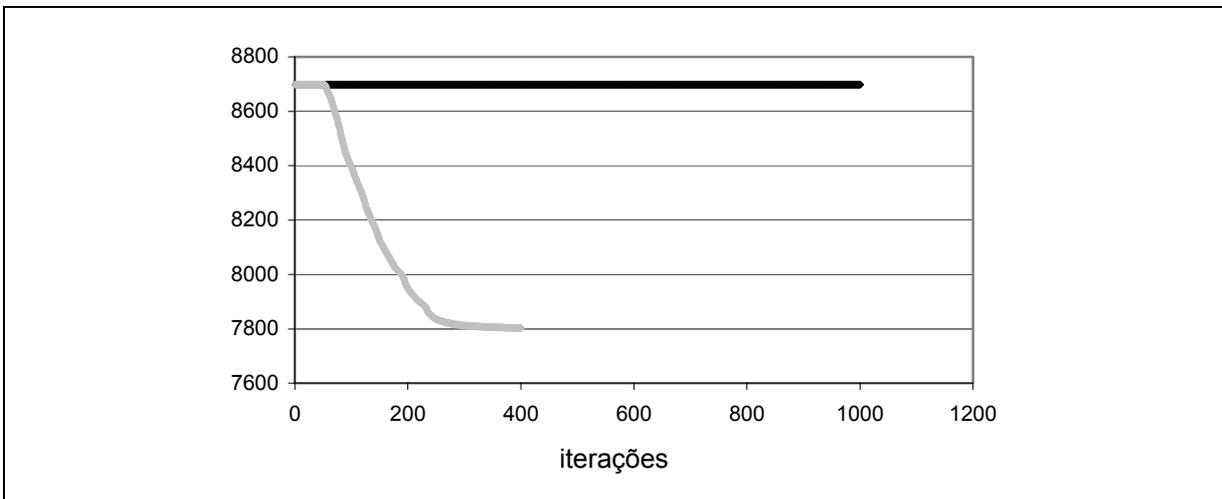


FIGURA 3.7 - Evolução de  $\nu(\overline{PCC})$  para  $fator\_rc = 0,3$ .

Nos testes computacionais realizados foi constatado que o valor do multiplicador lagrangeano/*surrogate t* converge para 1. A FIGURA 3.8 apresenta o comportamento crescente de *t* observado com os dados da instância  $n = 300$  e  $p = 5$ .

Também foram consideradas instâncias maiores, obtidas da *TSPLIB* (Reinelt, 1994), com até 3038 vértices. As TABELAS 3.4 a 3.6 apresentam os resultados obtidos para este conjunto de instâncias. Nessas tabelas, os valores *gap primal* e *gap dual* são calculados como:

- $gap\ primal = 100 \times |v(\overline{PCC}) - \text{melhor solução}| / \text{melhor solução};$
- $gap\ dual = 100 \times |\text{melhor solução} - v(L_r S_\lambda PPM)| / \text{melhor solução}.$

Os valores apresentados entre parênteses correspondem aos obtidos para o caso lagrangeano ( $t = 1$ ).

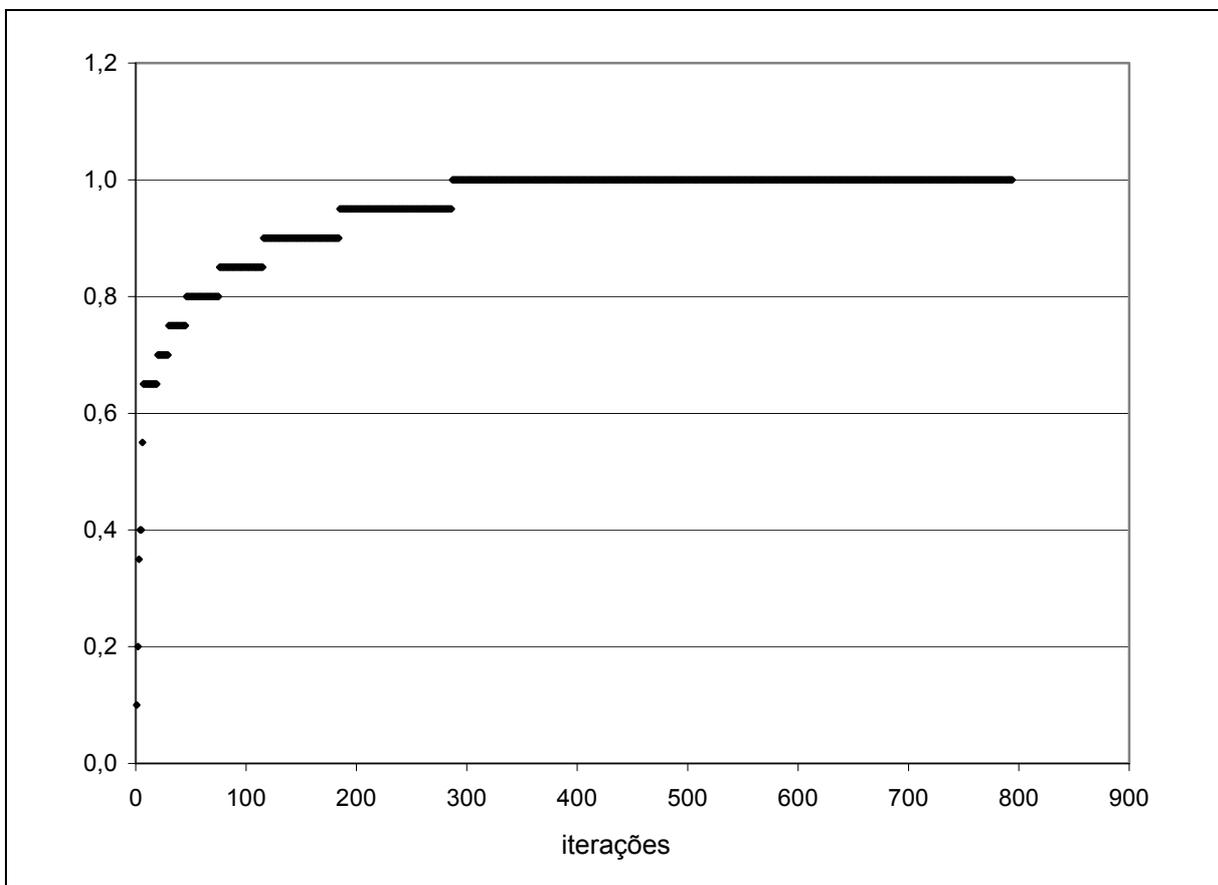


FIGURA 3.8 - Convergência do multiplicador lagrangeano/*surrogate*.

TABELA 3.4 - Resultados computacionais para instâncias PCB3038 ( $fator\_rc = 1,0$ ).

$p$	melhor solução	iterações	colunas geradas	colunas usadas	gap primal	gap dual	tempo
300	187723,46	42 (48)	58339 (65007)	44599 (44081)	0,043 (0,043)	0,044 (0,043)	22235,02 (35132,76)
350	170973,34	47 (37)	58758 (65545)	45576 (43956)	0,044 (0,044)	0,045 (0,045)	10505,93 (20457,59)
400	157030,46	33 (35)	50807 (60287)	37318 (39563)	0,008 (0,008)	0,008 (0,008)	4686,27 (8962,82)
450	145422,94	32 (30)	45338 (52515)	32637 (33544)	0,052 (0,052)	0,053 (0,052)	1915,84 (3241,71)
500	135467,85	22 (21)	31778 (36386)	22854 (22839)	0,036 (0,035)	0,036 (0,036)	597,86 (787,46)

TABELA 3.5 - Resultados computacionais para instâncias PCB3038 ( $fator\_rc = 0,5$ )

$p$	melhor solução	iterações	colunas geradas	colunas usadas	gap primal	gap dual	tempo
300	187723,46	79 (67)	96798 (111597)	40053 (39448)	0,043 (0,043)	0,044 (0,043)	19371,01 (36029,23)
350	170973,34	65 (53)	86113 (90651)	29179 (31664)	0,044 (0,044)	0,045 (0,044)	7077,99 (12905,94)
400	157030,46	53 (49)	77174 (94716)	22857 (30101)	0,008 (0,008)	0,008 (0,008)	2872,48 (5682,90)
450	145422,94	40 (41)	55870 (80631)	18662 (23767)	0,052 (0,052)	0,052 (0,053)	1288,56 (2568,56)
500	135467,85	34 (53)	45092 (79338)	16750 (22956)	0,036 (0,036)	0,036 (0,044)	716,78 (1425,33)

TABELA 3.6 - Resultados computacionais para instâncias PCB3038 ( $fator\_rc = 0,2$ )

$p$	melhor solução	iterações	colunas geradas	colunas usadas	gap primal	gap dual	tempo
300	187723,46	617 (834)	958984 (1655221)	28718 (93535)	0,043 (0,043)	0,044 (0,043)	36333,01 (117707,31)
350	170973,34	393 (719)	576789 (1232357)	24475 (74005)	0,044 (0,044)	0,044 (0,044)	10823,10 (49874,03)
400	157030,46	235 (586)	330475 (1232440)	15973 (54724)	0,008 (0,008)	0,008 (0,008)	4529,20 (39883,02)
450	145422,94	155 (363)	176348 (843026)	13489 (20517)	0,052 (0,052)	0,052 (0,052)	2356,97 (12990,88)
500	135467,85	121 (210)	119884 (420737)	12997 (24254)	0,035 (0,036)	0,035 (0,036)	1682,15 (4340,33)

Estes resultados confirmam que instâncias com um número relativamente pequeno de medianas (para um dado  $n$ ) são mais difíceis de serem resolvidas por técnicas de geração de

colunas. À medida que o número de medianas aumenta, mais *clusters* podem ser gerados e mais colunas podem ser incluídas na formulação do PMR.

É importante destacar que o critério de parada por limites permite encerrar a execução do algoritmo se o valor absoluto da diferença entre os limitantes for menor que um valor  $\epsilon$  estabelecido *a priori*, o que pode diminuir o tempo computacional necessário para resolver o  $\overline{PCC}$ . Por outro lado, se o algoritmo é encerrado pelo critério dos limitantes, é possível que conjunto final de colunas não seja favorável no início de métodos de busca em árvore binária, visando a obtenção de soluções viáveis para o *PPC*.

A possibilidade de ocorrência de soluções degeneradas foi apontada em Galvão (1981) como um importante fator que afeta a convergência de métodos que utilizam a formulação  $\overline{PCC}$  para obtenção de soluções para problemas de  $p$ -medianas. Neste caso, o valor do limitante inferior se mantém inalterado por várias iterações consecutivas em função da não alteração das variáveis duais do  $\overline{PCC}$ . Para o problema de localização de máxima cobertura, a possibilidade de existência de *clusters* com custo nulo favorece também a ciclagem, fenômeno em que múltiplas soluções viáveis apresentam o mesmo valor de função objetivo (Bazaraa *et al.*, 1990). Para garantir a terminação do algoritmo em tempo computacional finito também foram implementados dois critérios de parada baseados em número de iterações:

- Parada por número máximo de repetições consecutivas do valor do limitante inferior;
- Parada por número máximo de iterações do algoritmo  $CG(t)$ .

A avaliação do compromisso entre os critérios de parada se apresenta como um dos temas a serem abordados em pesquisa futura.



## CAPÍTULO 4

### IMPLEMENTAÇÃO DO ALGORITMO *BRANCH-AND-PRICE* E RESULTADOS COMPUTACIONAIS

O método *branch-and-price* (Barnhart *et al.*, 1998) utiliza a técnica de geração de colunas em cada nó de uma árvore de busca *branch-and-bound* para obtenção de novas variáveis não-básicas para um PMR. Senne *et al.* (2005) demonstram a aplicação bem sucedida desta metodologia para resolver problemas de  $p$ -medianas com até 900 vértices. Ceselli e Righini (2002) implementaram o *branch-and-price* para resolver instâncias de problema de  $p$ -medianas capacitado com até 200 vértices. A FIGURA 4.1 apresenta a estrutura geral do algoritmo *branch-and-price* implementado neste trabalho para resolução do problema de  $p$ -medianas com formulação dada por *PPC*.

No algoritmo da FIGURA 4.1, os problemas resultantes da aplicação da regra de ramificação são armazenados em uma lista. Dependendo da estratégia de exploração da árvore, esta lista pode ser implementada como uma pilha (lista LIFO), como uma fila simples (lista FIFO) ou como uma fila de prioridades (lista FIFO onde os problemas são inseridos em ordem crescente do valor dos limitantes inferiores). Listas LIFO proporcionam explorar a árvore em profundidade, permitindo obter soluções viáveis mais rapidamente. Listas FIFO percorrem a árvore em largura, visando obter melhores limitantes e soluções viáveis. Filas de prioridades buscam explorar os nós mais promissores de fornecerem uma solução ótima. O algoritmo mostra que a lista de problemas foi implementada como uma pilha.

#### 4.1 Condições de Poda.

As soluções dos PMRs resolvidos em cada nó da árvore podem ser viáveis para *PPC* ou não.

Em caso afirmativo, e se  $v(\overline{PCC})$  é maior que o menor valor de solução viável para o *PCC* disponível, este nó será descartado, definindo o processo de poda. A poda também ocorre se  $v(L_r S_\lambda PPM)$  é maior que o menor valor de solução viável para o *PCC*.

**Algoritmo BP:**

Seja  $P$  o problema mestre inicial. Faça  $lista \leftarrow \{P\}$ ;

Enquanto ( $lista \neq \emptyset$ )

    Faça  $P \leftarrow$  último problema inserido em  $lista$  e remova  $P$  de  $lista$ ;

    Faça  $z \leftarrow$  valor ótimo de  $P$ , resolvido pelo algoritmo **GC(t)**;

    Se (houver condições de poda)

        Podar o nó do problema  $P$ ;

    Senão

        Identifique o par de vértices  $q$  e  $r$  que definirão uma ramificação;

        Defina  $PE$  como o PMR formado pelas colunas de  $P$  onde os vértices  $q$  e  $r$  aparecem simultaneamente;

        Defina  $PD$  como o PMR formado pelas colunas de  $P$  onde os vértices  $q$  e  $r$  não aparecem simultaneamente;

        Faça  $lista \leftarrow lista \cup \{PE, PD\}$ ;

Fim.

FIGURA 4.1 - Algoritmo *Branch-And-Price*.

## 4.2 Regra de Ramificação.

A regra de ramificação considera o problema de particionamento de conjuntos idênticos descrita em Wolsey (1998), sugerida por Ryan e Foster (1981). Na ramificação da árvore, considera-se que os ramos à esquerda correspondem aos problemas nos quais os coeficientes das linhas  $q$  e  $r$  de uma dada coluna com valor fracionário na solução final do PMR são idênticos, e que nos ramos à direita apenas um dos coeficientes pode assumir o valor 1 (FIGURA 4.2). Esta regra permite identificar os pares de vértices que pertencem ao mesmo *cluster* na solução ótima de  $\overline{PCC}$ . O par de vértices  $(q, r)$  que definirá a ramificação é determinado da forma descrita a seguir.

### 4.2.1 Identificação de $q$ .

Seja  $X = \{x_1, \dots, x_m\}$  o conjunto de variáveis de decisão não-nulas correspondentes ao conjunto  $S = \{S_1, \dots, S_m\}$  de colunas de  $\overline{PCC}$ . Seja  $QS(i) = \{S_j \mid i \in S_j, j = 1, \dots, m\}$  para cada índice de linha  $i \in N$ . Então,  $q$  é escolhido como o índice de linha tal que  $|QS(q)| > |QS(i)|$ ,  $\forall i \in N, i \neq q$ . Note que, se  $|QS(q)| = 1$ , então  $X$  é uma solução viável para o  $PCC$ .

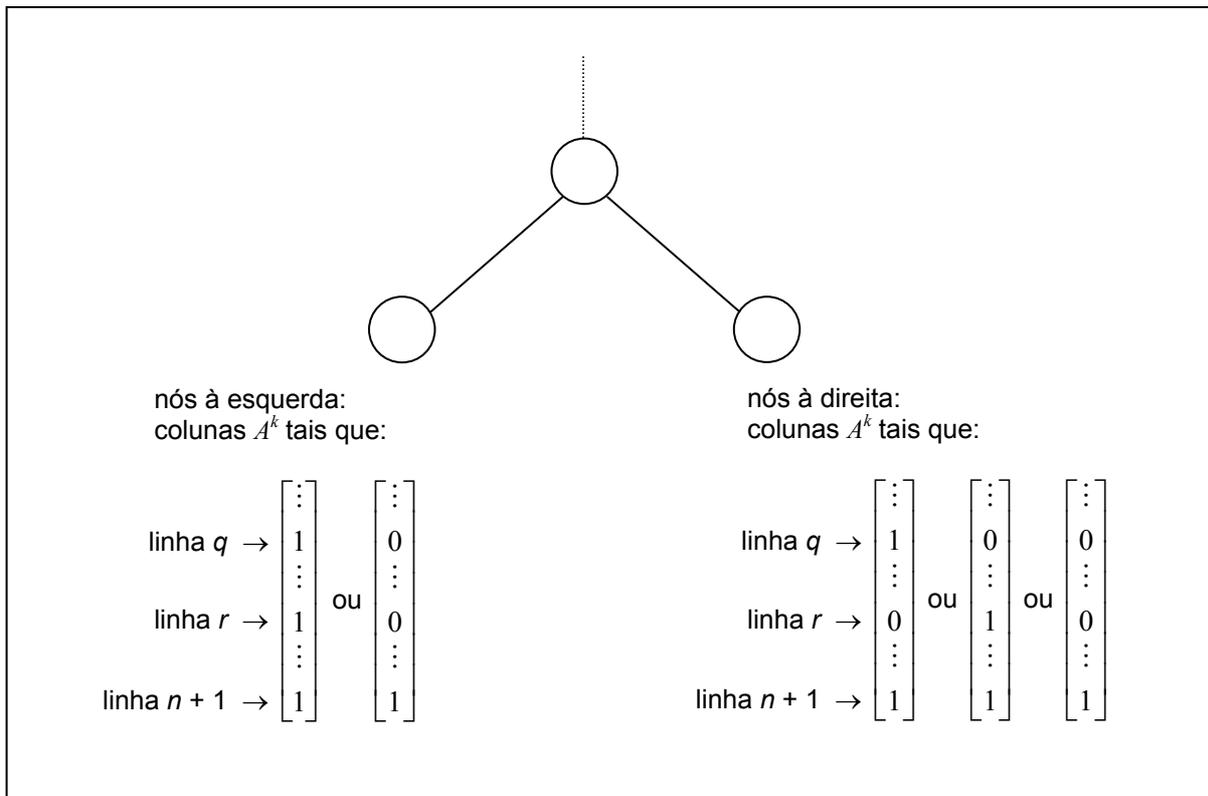


FIGURA 4.2 - Estrutura das colunas dos problemas segundo a regra de ramificação.

#### 4.2.2 Identificação de $r$ .

Seja  $RS(i) = \{S_j \in QS(q) \mid i \in S_j, j = 1, \dots, m\}$  para cada índice de linha  $i \in N$ . Seja  $R$  o conjunto de índices de linha para as quais o conjunto  $RS(i)$  é não-vazio, ou seja,  $R = \{i \mid RS(i) \neq \emptyset, i \in N\}$ . Então,  $r$  é escolhido como o índice de linha tal que  $|RS(r)| < |RS(i)|, \forall i \in R, i \neq r$ .

#### 4.3 Definição dos Subproblemas.

Uma vez determinado o par  $(q, r)$  de índices de linha, o seguinte problema inteiro binário deve ser resolvido nos ramos esquerdos da árvore de busca:

$$v(SGC_l) = \underset{j \in N}{\text{Min}} \left[ \underset{a_{ij} \in \{0,1\}}{\text{Min}} \sum_{i \in N} (d_{ij} - t\lambda_i) a_{ij} \right]$$

s. a.  $a_{qj} = a_{rj}, \quad \forall j \in N.$

Nos ramos direitos da árvore, deve ser resolvido o seguinte problema inteiro binário:

$$v(SGC_t) = \underset{j \in N}{\text{Min}} \left[ \underset{a_{ij} \in \{0,1\}}{\text{Min}} \sum_{i \in N} (d_{ij} - t\lambda_i) a_{ij} \right]$$

s. a.  $a_{qj} + a_{rj} \leq 1, \quad \forall j \in N.$

Convém destacar que ambos são problemas inteiros binários e que a complexidade desses subproblemas aumenta à medida que novos nós são adicionados, pois em um nó qualquer da árvore de busca o subproblema a ser resolvido deverá conter as restrições referentes às regras de ramificação adotadas em todos ramos existentes no caminho deste nó até o nó raiz.

A resolução de  $D_{t,\lambda}$  para obtenção do multiplicador lagrangeano/*surrogate*  $t$  considera implicitamente a regra de ramificação descrita na seção anterior, ou seja, as variáveis duais  $\lambda$  utilizadas nos subproblemas são obtidas a partir de PMRs com colunas apropriadas para o nó da árvore em questão.

#### 4.4 Resultados Computacionais.

Os algoritmos e rotinas apresentados neste trabalho foram implementados em linguagem C, utilizando o compilador Borland C++ Builder 5 com opções padrão para obtenção de um programa executável em linha de comando. Os testes, considerando instâncias de problemas de  $p$ -medianas da *OR-Library* (Beasley, 1990), foram conduzidos em um computador equipado com processador Intel Pentium 4 (2.6 GHz HT) e 1 GB de RAM, executando o sistema operacional Windows XP (com *Service Pack 2*).

Os subproblemas de geração de colunas foram resolvidos considerando os casos lagrangeano tradicional ( $t = 1$ ) e lagrangeano/*surrogate* ( $t$  obtido como solução aproximada de  $D_{t,\lambda}$ ). A resolução dos problemas lineares e dos subproblemas inteiros foi feita pelo aplicativo comercial ILOG CPLEX versão 7.5.

A legenda dos símbolos utilizados nas TABELAS 4.1 e 4.2 é descrita a seguir:

- $n$ : número de vértices da rede;
- $p$ : número de medianas;

- **solução ótima:** valor da solução ótima da instância;
- **colunas:** tamanho máximo do problema, em termos do número de colunas;
- **nós:** número de nós gerados na árvore de busca;
- **tempo:** tempo computacional total (em segundos).

A TABELA 4.1 apresenta os resultados obtidos para algumas das instâncias disponíveis na *OR-Library*.

TABELA 4.1 - Resultados obtidos para instâncias da *OR-Library*.

<i>n</i>	<i>p</i>	solução ótima	Lagrangeano			Lagrangeano/ <i>surrogate</i>		
			colunas	nós	tempo	colunas	nós	tempo
100	20	3034	2096	1	0,09	2131	1	0,09
100	33	1355	1613	1	0,03	1605	1	0,05
200	40	2734	3047	3	5,36	3157	3	7,73
200	67	1255	2663	1	0,20	2597	1	0,17
300	60	2968	3500	11	23,64	3512	9	32,56
300	100	1729	2404	1	0,39	2337	1	0,30
400	80	2845	3491	1	9,08	3494	1	2,70
400	133	1789	2776	1	0,42	2721	1	0,41
500	100	2961	27530	29	1606,49	14840	7	230,71
500	167	1828	3042	1	2,05	3377	3	16,87
600	120	3033	43563	165	11842,88	6453	11	140,22
600	200	1989	3784	7	55,38	3430	7	43,09
700	140	3013	38101	183	17447,67	9679	23	512,46
700	233	1847	4089	3	12,55	3836	1	1,44
800	267	2026	3942	7	35,70	3763	3	16,36
900	300	2106	4260	17	201,08	5635	19	332,84
<b>Total:</b>			<b>149901</b>	<b>432</b>	<b>31243,01</b>	<b>72567</b>	<b>92</b>	<b>1338,00</b>

Este conjunto de instâncias demonstra que a utilização do multiplicador lagrangeano/*surrogate* para modificar os coeficientes da função objetivo dos subproblemas de geração de colunas permite resolver os problemas explorando árvores menores. A relação entre o número total de colunas obtidas para o caso lagrangeano e para o caso lagrangeano/*surrogate* fornece um indicativo da qualidade superior das colunas deste último. Isso pode ser comprovado pelo tempo computacional total necessário para execução deste conjunto de instâncias.

Conforme observado nos resultados computacionais do capítulo anterior, métodos baseados em geração de colunas tornam-se computacionalmente mais custosos à medida que a relação  $n/p$  aumenta. A TABELA 4.2 apresenta os resultados obtidos para instâncias consideradas difíceis para a abordagem de geração de colunas, onde a relação  $n/p \geq 10$ .

TABELA 4.2 - Resultados obtidos para instâncias difíceis da *OR-Library*.

$n$	$p$	solução ótima	Lagrangeano			Lagrangeano/surrogate		
			colunas	nós	tempo	colunas	nós	tempo
100	5	5819	3372	1	3,52	3323	1	3,73
100	10	4093	5193	9	18,14	7534	35	106,57
100	10	4250	3415	15	11,61	4258	25	32,99
200	20	4445	8982	1	34,05	10498	3	61,94
300	30	4374	9225	5	181,53	8259	1	93,62
400	40	4809	49998	103	18890,35	17348	83	1255,62
500	50	4619	37242	3	3008,53	14459	3	354,24
<b>Total:</b>			<b>117427</b>	<b>137</b>	<b>22147,63</b>	<b>65679</b>	<b>151</b>	<b>1908,71</b>

Os resultados comprovam a superioridade da abordagem de geração de colunas modificada pelo multiplicador da relaxação lagrangeana/surrogate.

Visando testar a eficiência e a capacidade computacional das rotinas de otimização do CPLEX, as instâncias da *OR-Library* foram utilizadas para definir problema de  $p$ -medianas com formulação dada por *PPM*. Sendo  $N$  o conjunto de vértices do problema, é importante destacar que a matriz de restrições do *PPM* apresenta  $|N|^2 + 1$  linhas e  $|N|^2$  colunas, enquanto que na formulação *PCC* existem  $|N| + 1$  linhas e  $2^{|N|}$  colunas. A principal dificuldade apresentada pelo *PCC* é exatamente o grande número de colunas.

A legenda dos símbolos utilizados nas TABELAS 4.3 e 4.4 é descrita a seguir:

- $n$ : número de vértices da rede;
- $p$ : número de medianas;
- **variáveis**: número de colunas na matriz de restrições;
- **restrições**: número de linhas na matriz de restrições;
- **não-nulos**: número de elementos diferentes de zero presentes na matriz de restrições;

- **iterações**: número de iterações do Método Simplex;
- **nós**: número de nós gerados na árvore de busca;
- **solução**: valor da solução ótima da instância;
- **tempo**: tempo computacional total (em segundos).

TABELA 4.3 - Resultados obtidos pelo CPLEX para instâncias da *OR-Library*.

<i>n</i>	<i>p</i>	variáveis	restrições	não-nulos	iterações	nós	solução	tempo
100	20	10000	10001	29900	411	0	3034	0,27
100	33	10000	10001	29900	333	0	1355	0,23
200	40	40000	40001	119800	992	0	2734	1,66
200	67	40000	40001	119800	620	0	1255	1,41
300	60	90000	90001	269700	1660	0	2968	6,81
300	100	90000	90001	269700	1063	0	1729	4,72
400	80	160000	160001	479600	2051	0	2845	11,91
400	133	160000	160001	479600	1293	0	1789	10,44
500	100	250000	250001	749500	2713	0	2961	21,77
500	167	250000	250001	749500	1855	0	1828	21,58
600	120	360000	360001	1079400	3383	0	3033	37,08
600	200	360000	360001	1079400	1805	0	1989	24,22
700	140	490000	490001	1469300	3722	0	3013	76,83
700	233	490000	490000	1469300	2500	0	1847	49,25
800	267	640000	640001	1919200	2318	0	2026	50,06
900	300	810000	810001	2429100	2762	0	–	92,59

TABELA 4.4 - Resultados obtidos pelo CPLEX para instâncias difíceis da *OR-Library*.

<i>n</i>	<i>p</i>	variáveis	restrições	não-nulos	iterações	nós	solução	tempo
100	5	10000	10001	29900	1508	0	5819	0,63
100	10	10000	10001	29900	961	0	4093	0,84
100	10	10000	10001	29900	1010	2	4250	2,91
200	20	40000	40001	119800	1729	0	4445	2,17
300	30	90000	90001	269700	2559	0	4374	5,98
400	40	160000	160001	479600	3940	0	4809	18,70
500	50	250000	250001	749500	4572	0	4619	24,78

O símbolo “–” significa que o processo foi interrompido por falta de memória, sem que uma solução viável tenha sido encontrada.

O CPLEX conseguiu encontrar as soluções ótimas para todos os problemas dos dois conjuntos de instâncias, exceto para a instância com 900 vértices, e em apenas um desses casos foi necessário descer um nível na árvore de busca.

Os limites físicos impostos pelos recursos computacionais existentes justificam a pesquisa por métodos alternativos para tratamento de instâncias de problemas de  $p$ -medianas com um número relativamente grande de vértices.

## CAPÍTULO 5

### INSTÂNCIAS DIFÍCEIS: O PROBLEMA DE LOCALIZAÇÃO DE MÁXIMA

#### COBERTURA

O problema de localização de máxima cobertura (PLMC) busca obter uma configuração de localização de  $p$  facilidades para o atendimento do maior número de clientes – representados pela demanda dos vértices em uma rede – para uma dada distância ou tempo de serviço. Aplicações deste problema são encontradas principalmente para definir locais de instalação de serviços de utilidade pública emergenciais (bombeiros, ambulâncias, postos de policiamento) ou não emergenciais (escolas, postos de correios). Também é crescente o número de aplicações deste problema para a determinação de reservas de proteção ambiental (Church *et al.*, 1996) e áreas de vigilância (Scaparra e Church, 2005). No setor privado encontram-se aplicações do PLMC em projetos de distribuição de antenas de telefonia celular (Tutschku, 1997), dentre outros.

#### 5.1 Formulação Matemática.

Considere um grafo  $G = (N, A)$ , com  $|N| = n$  e matriz de distâncias entre cada par de vértices dada por  $D = [d_{ij}]_{n \times n}$ . Sendo  $b_i$  a informação de demanda para cada vértice  $i \in N$ ,  $p$  o número de facilidades a serem localizadas e  $U$  a distância de serviço, Church e ReVelle (1974) apresentam a seguinte formulação para o PLMC:

$$PMC \quad v(PMC) = \text{Max} \sum_{i \in N} b_i v_i \quad (5.1)$$

$$\text{s. a.} \quad \sum_{j \in N} e_{ij} y_j \geq v_i, \quad \forall i \in N \quad (5.2)$$

$$\sum_{j \in N} y_j = p \quad (5.3)$$

$$v_i \in \{0, 1\}, \quad \forall i \in N \quad (5.4)$$

$$y_j \in \{0, 1\}, \quad \forall j \in N \quad (5.5)$$

onde  $e_{ij} = 1$  se a demanda do vértice  $i$  pode ser coberta pela facilidade instalada no vértice  $j$  ( $d_{ij} \leq U$ ), e  $e_{ij} = 0$ , caso contrário;  $v_i = 1$  se o vértice  $i$  for atendido por alguma facilidade a menos de  $U$  unidades de distância, e  $v_i = 0$ , caso contrário;  $y_j = 1$  se o vértice  $j$  for escolhido para instalação de uma facilidade, e  $y_j = 0$ , caso contrário.

A matriz de restrições da formulação  $PMC$  possui  $|N| + 1$  linhas e  $2|N|$  colunas. A função objetivo (5.1) tenta maximizar a demanda total atendida. A restrição (5.2) especifica que cada vértice de demanda será considerado coberto apenas se uma facilidade for instalada a menos de  $U$  unidades de distância. A restrição (5.3) determina o número de facilidades a serem instaladas e as restrições (5.4) e (5.5) estabelecem o domínio de validade das variáveis do problema. Para o caso particular em que  $b_i = 1, \forall i \in N$ ,  $v(PMC)$  informa o maior número de vértices possível de serem cobertos, para uma dada distância de atendimento  $U$ .

As primeiras técnicas de resolução do PLMC visavam a obtenção de soluções a partir da formulação de relaxação de programação linear de  $PMC$ . Em seu trabalho de apresentação do PLMC, Church e ReVelle (1974) propõem uma heurística gulosa baseada em troca de vértices. Galvão *et al.* (2000) apresentam uma comparação entre as relaxações lagrangeana e *surrogate* para resolver problemas de máxima cobertura com até 900 vértices. Arakaki e Lorena (2001) apresentam uma implementação do algoritmo genético construtivo para instâncias reais do PLMC com até 500 vértices. Resende (1998) apresenta resultados com a aplicação do GRASP a instâncias reais de grande porte e geradas aleatoriamente.

Outras abordagens baseadas na estrutura da formulação  $PMC$  para resolução do PLMC podem ser encontradas na literatura.

## 5.2 Formulação de $p$ -Medianas para o PLMC.

Hillsman (1984) propõe definir novos coeficientes para a função objetivo (2.1) da formulação  $PPM$ , baseando-se na informação de demanda nos vértices do PLMC, como a seguir:

$$c_{ij} = \begin{cases} 0, & \text{se } d_{ij} \leq U \\ b_i, & \text{se } d_{ij} > U \end{cases} \quad (5.6)$$

permitindo a aplicação de métodos desenvolvidos para o problema de  $p$ -medianas na resolução de problemas de máxima cobertura.

O valor ótimo da função objetivo  $v(PPM)$  com coeficientes dados por (5.6) reflete a demanda não atendida. Para obter o valor da demanda coberta na solução ótima faz-se:

$$v(PMC) = \sum_{i \in N} b_i - v(PPM)$$

Lorena e Pereira (2002) apresentam os resultados obtidos com a aplicação de um algoritmo de otimização de subgradientes para resolver a relaxação lagrangeana/*surrogate* da formulação  $PPM$  com os coeficientes da função objetivos substituídos pelos coeficientes calculados em (5.6). Foram utilizadas instâncias de dados reais da cidade de São José dos Campos para simular a instalação de antenas de provedores de serviços de Internet com alcances de 800, 1.200 e 1.600 m, com o número de facilidades a serem instaladas variando de uma até o mínimo necessário para a cobertura de todos os vértices. A FIGURA 5.1 mostra a solução de cobertura para uma instância com 708 vértices, onde foram localizadas três antenas com alcance  $U = 800$  m.



FIGURA 5.1 - Localização de antenas em São José dos Campos - SP.

Para conduzir os estudos realizados neste trabalho, foram geradas novas instâncias considerando  $U = 150$  m, permitindo avaliar o desempenho do algoritmo de geração de colunas para valores onde a relação  $n/p \in [3, 10]$ . Os dados relativos à este novo conjunto de instâncias estão disponíveis em <http://www.lac.inpe.br/~lorena/instancias.html>.

O algoritmo de subgradientes de Lorena e Pereira (2002) foi aplicado às instâncias de São José dos Campos (TABELA 5.1), visando obter soluções ótimas ou limites inferiores para posterior comparação com as obtidas pelo emprego do algoritmo da FIGURA 3.4.

TABELA 5.1 - Instâncias da cidade de São José dos Campos - SP.

Número de vértices ( $n$ )	Número de facilidades a serem instaladas ( $p$ )
324	20, 30, 40, 50, 60, 80, 108
402	30, 40, 50, 60, 70, 80, 100, 134
500	40, 50, 60, 70, 80, 100, 130, 167
708	70, 80, 90, 100, 120, 140, 180, 236
818	80, 90, 100, 120, 140, 160, 200, 273

Esta linha de pesquisa foi abandonada após verificar, baseados em estudos computacionais, a dificuldade em se estabelecer um conjunto de parâmetros – tamanho e frequência de atualização do passo, valor inicial do vetor de multiplicadores *surrogate*, valor inicial e frequência de atualização do parâmetro  $\pi$  (Held e Karp, 1971), critérios de parada – adequado para todas as instâncias, que possibilitasse a aplicação do algoritmo de subgradientes para a resolução dos problemas  $L_1S_\lambda PPM$  e  $L_tS_\lambda PPM$  (Narciso, 1998).

A descontinuidade dos coeficientes de  $L_tS_\lambda PPM$  com coeficientes da função objetivo dados por (5.6) causou dificuldades também para o algoritmo de busca do multiplicador lagrangeano/*surrogate* (Schilling *et al.*, 2000). A determinação de novos valores iniciais para  $t$  e a implementação de algoritmos eficientes que executassem uma busca abreviada também não forneceu bons resultados para o novo conjunto de instâncias.

### 5.3 Geração de Colunas com Ampliação e Perturbação.

Conforme observado anteriormente, métodos de Programação Linear desenvolvidos para resolver problemas com formulações dadas por  $PPC$  e  $\overline{PCC}$  apresentam dificuldades de convergência, devido à ocorrência de soluções básicas com valor nulo. Este fenômeno é

conhecido na literatura como degeneração. Esta é uma característica presente em vários problemas resolvidos por métodos de geração de colunas, pois à medida que o método progride, colunas com custo reduzido nulo podem ser geradas para substituir outras colunas com custo nulo. No caso do PLMC resolvido pela formulação de  $PPM$ , colunas de custo nulo são desejáveis, pois identificam *clusters* onde toda a demanda foi atendida. Dessa forma, o cálculo de soluções para o PLMC pelo algoritmo de geração de colunas apresentado neste trabalho apresenta-se como um grande desafio.

Em testes computacionais preliminares com instâncias do PLMC foi possível verificar que o processo de obtenção de soluções pelo algoritmo de geração de colunas originalmente implementado para resolver instâncias de problemas de  $p$ -medianas não apresentava convergência satisfatória. Constatou-se que, na maioria dos casos, os limites inferiores fornecidos pela resolução do problema  $L_t S_\lambda PPM$  permaneciam inalterados por muitas iterações, indicando que o conjunto de colunas do  $\overline{PCC}$  produzia sempre a mesma solução dual  $\lambda$ .

Numa tentativa de provocar alterações significativas nos valores obtidos para  $\lambda$ , permitiu-se a inclusão em  $\overline{PCC}$  de todas as colunas obtidas da resolução do subproblema gerador de colunas, mesmo as que apresentavam custos reduzidos positivos, após um número pré-determinado de iterações consecutivas sem melhoria em  $v(L_t S_\lambda PPM)$ . Com isso, os valores duais se alteravam e os limites inferiores voltavam a aumentar. Esse procedimento foi denominado perturbação, podendo ser aplicado no método de geração de colunas para os casos lagrangeano e lagrangeano/*surrogate*.

A idéia de incluir mais colunas no  $\overline{PCC}$  numa única iteração serviu de inspiração para outro procedimento, para inserir ainda mais colunas no PMR. Iniciando com valores próximos de 0, o multiplicador lagrangeano/*surrogate*  $t$  converge para 1 à medida que o algoritmo  $GC(t)$  prossegue. Visando aumentar o desempenho do algoritmo nos passos iniciais, em todas iterações do algoritmo  $CG(t)$  seriam acrescentadas à formulação do PMR as colunas com custo reduzido calculadas com o valor de  $t$  obtido do procedimento de busca (FIGURA 2.2) e também as calculadas com valores do conjunto  $T = \{0,50; 0,55; 0,60; 0,65; 0,70; 0,75; 0,80; 0,85; 0,90; 1,0\}$  maiores que  $t$  corrente, tentando antecipar informação que só estaria

disponível em iterações avançadas do algoritmo. Este procedimento foi denominado ampliação, e é válido apenas para o caso lagrangeano/*surrogate*.

Em testes computacionais, foi observado que o emprego da perturbação em algumas instâncias resolvidas pelo algoritmo de geração de colunas, caso lagrangeano/*surrogate*, provocou fortes alterações no comportamento crescente do valor do limitante inferior, quando  $t$  já havia convergido para 1. A seqüência foi interrompida, com  $v(L_t S_\lambda PPM)$  assumindo valores muito inferiores aos anteriormente obtidos. Por esse motivo, um mecanismo de controle foi implementado, evitando que se faça a perturbação quando  $t = 1$  no caso lagrangeano/*surrogate*. Observou-se também que o emprego constante da perturbação tende a produzir valores de soluções primais  $v(\overline{PCC})$  de valor reduzido, e o processo de geração de colunas converge para valores de limitantes inferiores mais baixos. Como o tamanho da árvore de busca depende dos valores de tais limitantes, deseja-se que os mesmos convirjam para valores mais elevados, fazendo com que o processo de poda da árvore seja mais eficaz.

Com tantas colunas sendo incluídas por iteração, a rotina de remoção de colunas era executada com mais freqüência, sempre que o número total de colunas fosse maior que 30.000. Este valor foi estabelecido, baseado em experimentos computacionais, como o que fornecia o melhor compromisso entre quantidade de informação e rapidez na obtenção de soluções.

O desempenho do algoritmo de geração de colunas com os procedimentos de perturbação, ampliação e controle mencionados pode ser comprovado nas figuras a seguir. Os gráficos mostram a evolução dos valores das soluções duais  $v(L_t S_\lambda PPM)$  (curva inferior) e das soluções primais (curva superior) para a instância  $n = 324$ ,  $p = 20$  e  $U = 150$ . Na FIGURA 5.2 a execução de  $CG(t)$  foi interrompida após 80 iterações, por falta de colunas de custo reduzido negativo. Na FIGURA 5.3, o processo se estendeu até a iteração 1.972, sendo interrompido pela falta de melhoria em  $v(L_t S_\lambda PPM)$  por 1.000 iterações consecutivas. Na FIGURA 5.4 foram executadas 374 iterações do método até a convergência dos valores dos limitantes.

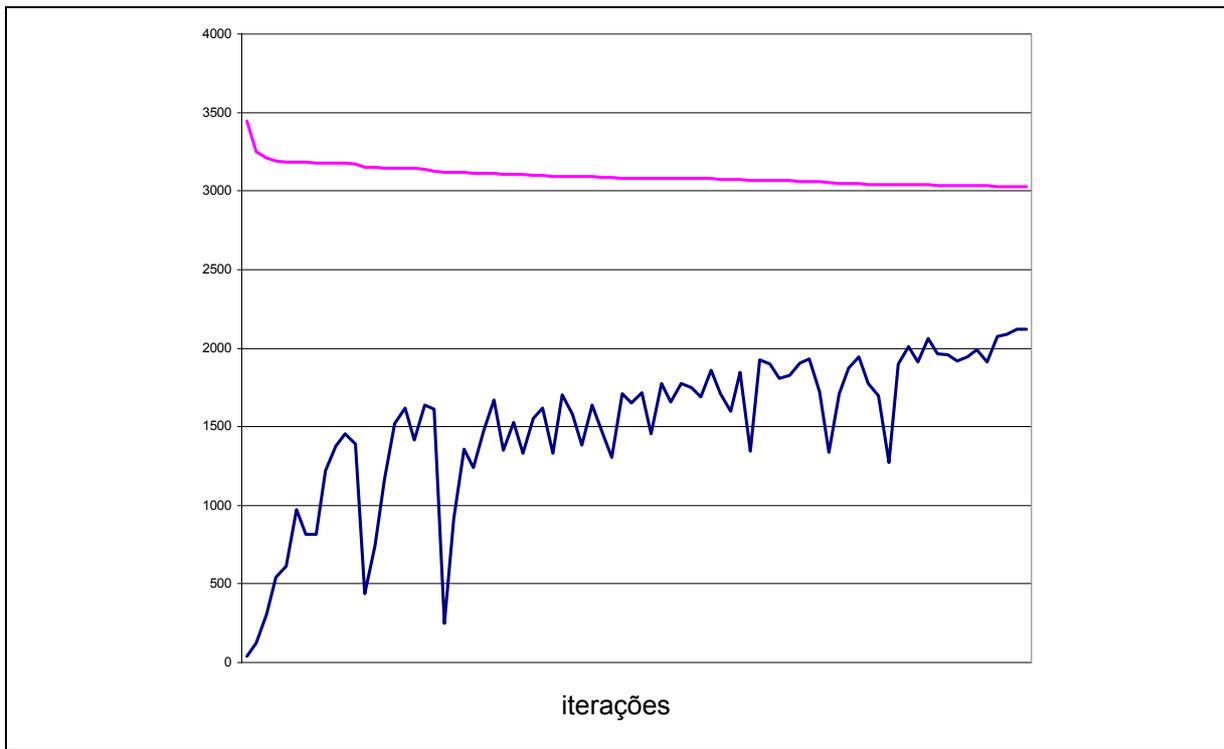


FIGURA 5.2 -  $GC(t)$  sem ampliação ou perturbação:  $v(L_t S_\lambda PPM) = 2121,40$ .

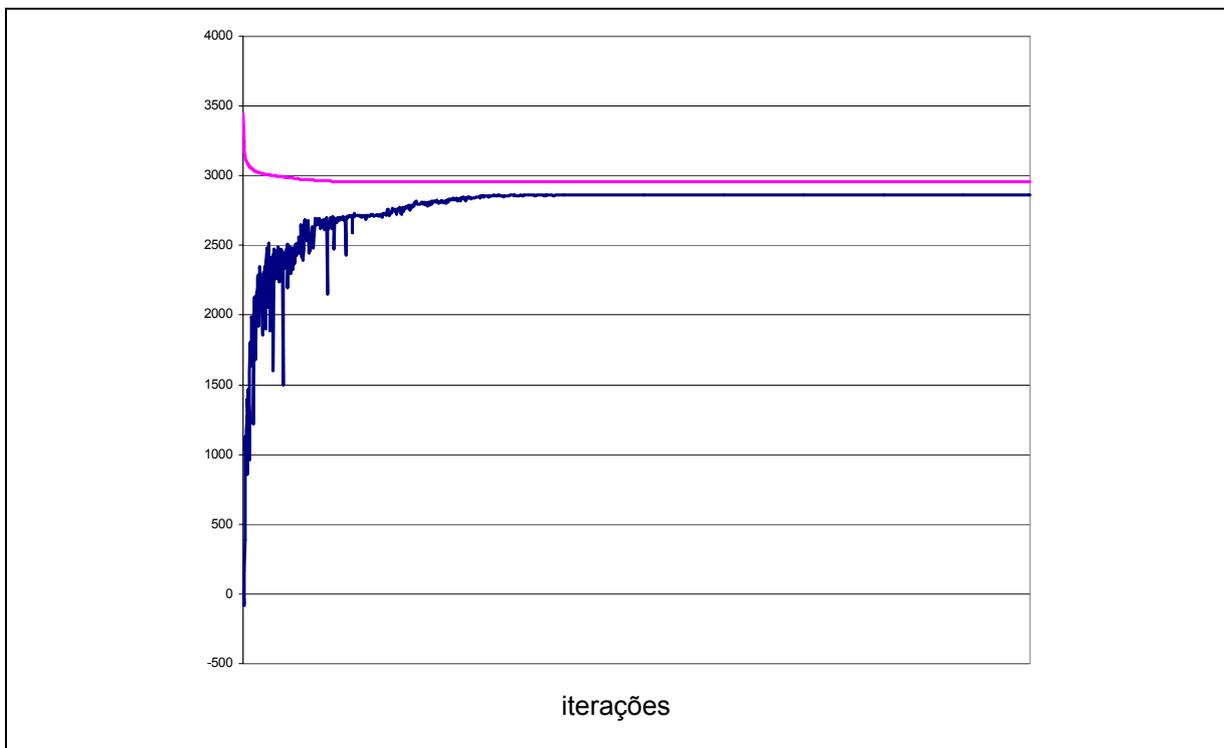


FIGURA 5.3 -  $GC(t)$  com ampliação:  $v(L_t S_\lambda PPM) = 2862,21$ .

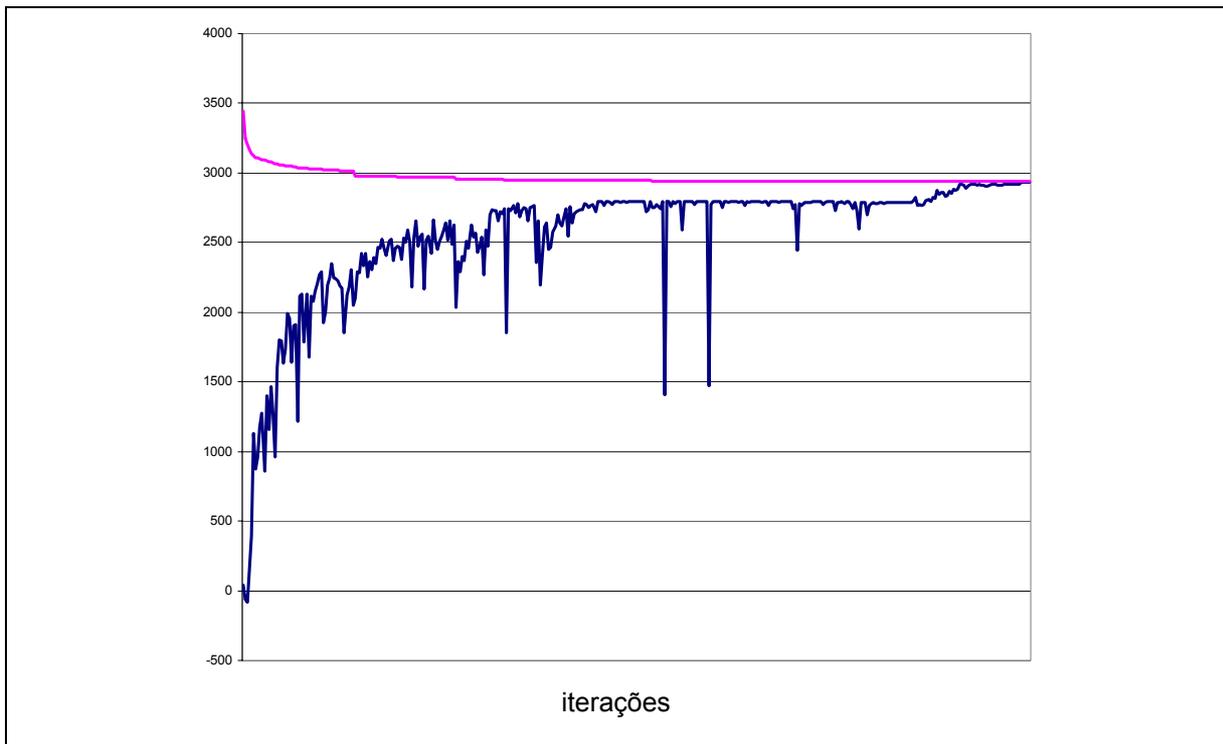


FIGURA 5.4 -  $GC(t)$  com ampliação e controle de perturbação:  $v(L_r S_\lambda PPM) = 2936,49$ .

#### 5.4 Resultados Computacionais

Foram então conduzidos testes computacionais com o algoritmo  $CG(t)$  com procedimentos de ampliação e perturbação controlada, considerando as instâncias definidas na TABELA 5.1. A plataforma computacional foi a mesma que a utilizada para execução do algoritmo *branch-and-price* apresentado no capítulo anterior.

A legenda usada nas TABELAS 5.2 a 5.6 é a seguinte:

- $n$ : número de vértices da rede;
- $p$ : número de facilidades;
- **iterações**: número de iterações executadas pelo algoritmo  $CG(t)$ ;
- **colunas geradas**: número total de colunas geradas;
- **colunas usadas**: número total de colunas aproveitadas no PMR;
- **limite inferior**: melhor valor  $v(L_r S_\lambda PPM)$  encontrado;

- **gap**: diferença percentual entre  $v(\overline{PCC})$  e o limitante inferior;
- **tempo**: tempo computacional total (em segundos).

Os números entre parênteses são os valores obtidos para o caso lagrangeano tradicional. O símbolo “-” na coluna **gap** indica que os limitantes convergiram para o mesmo valor, ou que  $v(\overline{PCC}) < v(L_r S_\lambda PPM)$

TABELA 5.2 - Resultados obtidos para instâncias de S. J. dos Campos com 324 vértices.

<i>n</i>	<i>p</i>	iterações	colunas geradas	colunas usadas	limite inferior	gap	tempo
324	20	881 (100000)	510866 (11702106)	28675 (1915)	4763,06 (4522,45)	- (3,244)	523,23 (40944,96)
	30	2257 (59937)	1376214 (13421937)	23069 (18688)	2911,73 (2160,09)	- (29,675)	530,38 (16451,18)
	40	1313 (7478)	1114176 (1913483)	6783 (30082)	1537,18 (1075,44)	- (40,152)	292,24 (1219,45)
	50	438 (2423)	326096 (700628)	5344 (9876)	689,11 (386,53)	- (58,024)	83,92 (299,70)
	60	1043 (952)	176719 (156259)	85681 (12580)	108,81 (124,31)	18,692 (-)	74,66 (222,30)
	80	9 (33)	27627 (11290)	27627 (11290)	0,00 (0,00)	- (-)	1,81 (2,14)
	108	5 (16)	16238 (6429)	16238 (6429)	0,00 (0,00)	- (-)	0,53 (0,58)

TABELA 5.3 - Resultados obtidos para instâncias de S. J. dos Campos com 402 vértices.

<i>n</i>	<i>p</i>	iterações	colunas geradas	colunas usadas	limite inferior	gap	tempo
402	30	10441 (100000)	7862281 (34654329)	96104 (20847)	4499,20 (2826,46)	3,512 (42,107)	3393,48 (55877,57)
	40	3146 (100000)	3885112 (30366135)	3095 (7562)	2816,95 (1977,76)	- (41,166)	1285,91 (28147,48)
	50	3453 (26872)	3512811 (9101114)	4245 (25563)	1789,38 (683,55)	- (69,954)	1133,55 (6364,61)
	60	1837 (6189)	1918008 (2368197)	22955 (30318)	962,86 (- 725,18)	- (> 100)	372,21 (1122,58)
	70	1222 (2170)	1897366 (830359)	6109 (19081)	255,35 (- 610,41)	20,792 (> 100)	223,74 (312,17)
	80	1039 (292)	87151 (56664)	23895 (27280)	32,05 (41,06)	30,336 (-)	31,64 (28,05)
	100	8 (32)	27883 (12892)	27883 (12892)	0,00 (0,00)	- (-)	1,62 (2,38)
	134	5 (15)	21216 (7120)	21216 (7120)	0,00 (0,00)	- (-)	0,59 (0,62)

TABELA 5.4 - Resultados obtidos para instâncias de S. J. dos Campos com 500 vértices.

<i>n</i>	<i>p</i>	iterações	colunas geradas	colunas usadas	limite inferior	gap	tempo
500	40	10797 (100000)	21422705 (47947348)	3146 (2334)	5382,43 (2942,82)	– (55,219)	11403,01 (87822,27)
	50	29107 (100000)	29005863 (49810865)	178519 (18372)	4168,67 (779,87)	9,048 (85,513)	8032,36 (63239,42)
	60	24108 (100000)	28507334 (48017859)	205444 (6621)	3379,67 (640,88)	3,692 (85,428)	6826,83 (40509,12)
	70	668 (100000)	2027369 (46476364)	570 (25249)	1854,00 (649,12)	– (81,682)	597,89 (29953,01)
	80	8526 (52658)	10943885 (25102301)	17265 (18739)	1792,27 (– 260,48)	– (> 100)	1824,89 (12270,24)
	100	108 (9618)	403308 (4703855)	1182 (1217)	433,99 (– 1077,10)	– (> 100)	74,89 (2420,87)
	130	20 (336)	95257 (130206)	870 (11262)	6,83 (0,36)	– (–)	7,22 (46,19)
	167	7 (43)	35617 (22010)	11276 (22010)	0,00 (0,00)	– (–)	1,84 (4,40)

TABELA 5.5 - Resultados obtidos para instâncias de S. J. dos Campos com 708 vértices.

<i>n</i>	<i>p</i>	iterações	colunas geradas	colunas usadas	limite inferior	gap	tempo
708	70	9467 (100000)	37654385 (70801933)	1438 (30268)	3262,17 (– 8569,14)	– (> 100)	23566,19 (182079,69)
	80	7353 (100000)	28081728 (70801933)	1024 (30236)	2651,88 (– 6273,93)	– (> 100)	13826,58 (127565,43)
	90	9949 (100000)	30628953 (70796775)	14076 (28100)	2305,90 (– 4943,89)	– (> 100)	10563,38 (84151,88)
	100	2358 (100000)	10326618 (70719646)	920 (1138)	1446,75 (– 3148,53)	– (> 100)	3222,66 (52267,47)
	120	52 (16382)	383893 (11582316)	783 (24534)	272,47 (– 3,00)	– (> 100)	155,55 (38174,01)
	140	164 (3198)	1053089 (2199458)	739 (12958)	139,61 (– 877,93)	– (> 100)	185,02 (3856,92)
	180	243 (95)	97573 (57986)	25538 (28363)	0,30 (3,01)	– (–)	21,18 (18,62)
	236	7 (25)	45613 (18105)	15731 (18105)	0,00 (0,00)	– (–)	2,43 (2,50)

TABELA 5.6 - Resultados obtidos para instâncias de S. J. dos Campos com 818 vértices.

<i>n</i>	<i>p</i>	iterações	colunas geradas	colunas usadas	limite inferior	gap	tempo
818	80	29947 (84679)	130865821 (69268588)	25159 (2070)	4173,99 (- 3,00)	25,059 (> 100)	100206,88 (206442,80)
	90	8084 (100000)	40231415 (81801981)	827 (14514)	2901,62 (- 15780,4)	- (> 100)	27721,61 (178731,16)
	100	4948 (100000)	24244186 (81801981)	20287 (14414)	2411,37 (- 11254,1)	32,192 (> 100)	16630,18 (131980,80)
	120	334 (100000)	2797745 (81535844)	9558 (28897)	820,88 (- 3117,13)	- (> 100)	1597,38 (46365,92)
	140	397 (21744)	2665881 (17627162)	660 (19117)	509,47 (- 2602,10)	- (> 100)	716,29 (6055,78)
	160	197 (6845)	1352662 (5599300)	852 (1208)	176,26 (- 9,00)	- (> 100)	384,90 (2289,91)
	200	1070 (418)	115052 (295776)	18156 (28315)	8,02 (6,13)	65,228 (-)	103,55 (101,070)
	273	8 (29)	53002 (23892)	33296 (23892)	0,00 (23892)	0,00 (-)	3,26 (4,48)

O número máximo de iterações permitido foi fixado em 100000. O procedimento de perturbação era aplicado sempre que não houvesse melhora em  $v(L_r S_\lambda PPM)$  por 10 iterações consecutivas. O pequeno valor para a relação entre o número de colunas aproveitadas e o número de colunas geradas reflete a utilização intensa do procedimento de perturbação.

O número de colunas geradas na relaxação lagrangeana/*surrogate* é um indicativo da qualidade superior das mesmas, quando comparadas com as obtidas pela relaxação lagrangeana tradicional. Isto também pode ser verificado pelo fato que para um número significativo de instâncias o algoritmo de geração de colunas considerando a relaxação lagrangeana tradicional atingiu o número máximo de iterações permitido sem obter limitantes razoáveis. A relaxação lagrangeana/*surrogate* conseguiu encontrar os melhores valores de limitantes inferiores na maioria dos casos.

Os tempos obtidos para as instâncias em questão correspondem apenas ao tempo necessário para a resolução do PMR no nó raiz do algoritmo *branch-and-price*. Outros desenvolvimentos devem ser realizados visando permitir a aplicação do mesmo para obtenção de soluções viáveis para problemas de máxima cobertura com formulação dada pelo PPC em tempos computacionais reduzidos. Este fato pode ser confirmado em vista dos resultados obtidos para as mesmas instâncias, considerando-se as formulações PPM e PMC, resolvidas de forma inteira pelo CPLEX, e mostradas nas TABELAS 5.7 e 5.8 (a legenda

usada é a mesma das TABELAS 4.3 e 4.4). Os símbolos em parênteses utilizados ao lado dos valores de solução obtidos têm o seguinte significado:

- $\epsilon$ : solução inteira ótima para a tolerância especificada (gap relativo  $< 10^{-4}$ );
- s: solução inteira sub-ótima para a tolerância especificada (gap relativo  $> 10^{-4}$ );

TABELA 5.7 - Resultados para as instâncias de S. J. dos Campos pela formulação PPM.

<i>n</i>	<i>p</i>	variáveis	restrições	não-nulos	iterações	nós	solução	tempo
324	20	104976	104977	314604	4409	0	4850	5,344
	30	104976	104977	314604	3832	0	3025	7,890
	40	104976	104977	314604	3697	5	1709	76,703
	50	104976	104977	314604	3012	8	755	84,594
	60	104976	104977	314604	2339	1	161	22,312
	80	104976	104977	314604	1631	0	0	11,110
	108	104976	104977	314604	981	0	0	8,406
402	30	161604	161605	484410	5272	0	4911	8,890
	40	161604	161605	484410	5092	20	3309	100,203
	50	161604	161605	484410	4353	4	2013	100,453
	60	161604	161605	484410	3271	0	1020	20,781
	70	161604	161605	484410	2823	46	362	71,391
	80	161604	161605	484410	5394	118	69	271,844
	100	161604	161605	484410	2115	0	0	20,375
134	161604	161605	484410	1127	0	0	12,203	
500	40	250000	250001	749500	5604	0	6367	14,735
	50	250000	250001	749500	4835	0	4934	19,313
	60	250000	250001	749500	4032	0	3788	23,140
	70	250000	250001	749500	3458	0	2799	14,140
	80	250000	250001	749500	2895	0	1958	14,125
	100	250000	250001	749500	2310	4	795	57,203
	130	250000	250001	749500	1967	0	43	32,562
	167	250000	250001	749500	1184	0	0	20,078
708	70	501264	501265	1503084	7582	0	4711	50,390
	80	501264	501265	1503084	6806	0	3659	59,860
	90	501264	501265	1503084	6325	0	2743 ( $\epsilon$ )	67,969
	100	501264	501265	1503084	5630	0	2019	63,078
	120	501264	501265	1503084	5001	16	992	547,000
	140	501264	501265	1503084	4583	13	331	359,407
	180	501264	501265	1503084	3289	0	7	78,062
	236	501264	501265	1503084	1768	0	0	50,218
818	80	669124	669125	2006554	9729	0	5843	117,531
	90	669124	669125	2006554	8387	1	4713	269,766
	100	669124	669125	2006554	116366	5220	3733 (s)	7304,172
	120	669124	669125	2006554	12695	1730	2186 (s)	2408,469
	140	669124	669125	2006554	24436	1680	1165 (s)	2480,531
	160	669124	669125	2006554	24502	1400	470 (s)	2436,859
	200	669124	669125	2006554	3438	0	-	82,797
	273	669124	669125	2006554	2184	0	0 ( $\epsilon$ )	105,641

Para o restante dos casos, a solução encontrada é comprovadamente inteira ótima.

O CPLEX não conseguiu encontrar uma solução inteira para apenas uma das instâncias estudadas com a formulação *PPM*.

TABELA 5.8 - Resultados para as instâncias de S. J. dos Campos pela formulação *PMC*.

n	p	variáveis	restrições	não-nulos	iterações	nós	solução	tempo
324	20	648	325	2504	435	0	4850	0,078
	30	648	325	2504	476	0	3025	0,046
	40	648	325	2504	540	0	1709	0,047
	50	648	325	2504	650	3	755	0,265
	60	648	325	2504	757	27	161	0,375
	80	648	325	2504	475	20	1 ( $\epsilon$ )	0,141
	108	648	325	2504	134	0	0	0,016
402	30	804	403	3038	577	0	4911	0,031
	40	804	403	3038	696	10	3309	0,141
	50	804	403	3038	671	6	2013	0,156
	60	804	403	3038	722	2	1020	0,172
	70	804	403	3038	558	8	362	0,109
	80	804	403	3038	994	26	69 ( $\epsilon$ )	0,313
	100	804	403	3038	657	30	1 ( $\epsilon$ )	0,296
	134	804	403	3038	140	0	0	0,016
500	40	1000	501	3072	664	0	6367	0,031
	50	1000	501	3072	655	0	4934	0,032
	60	1000	501	3072	667	0	3788	0,031
	70	1000	501	3072	672	0	2799	0,031
	80	1000	501	3072	695	0	1958	0,031
	100	1000	501	3072	693	4	795	0,094
	130	1000	501	3072	622	4	43	0,141
	167	1000	501	3072	385	0	0	0,046
708	70	1416	709	4958	1093	0	4711	0,094
	80	1416	709	4958	1114	2	3659 ( $\epsilon$ )	0,235
	90	1416	709	4958	1099	0	2743	0,094
	100	1416	709	4958	1007	0	2019	0,078
	120	1416	709	4958	1206	18	994 ( $\epsilon$ )	0,343
	140	1416	709	4958	1087	8	331	0,547
	180	1416	709	4958	961	23	7	0,516
	236	1416	709	4958	374	0	0	0,047
818	80	1636	819	5756	1196	0	5843	0,110
	90	1636	819	5756	1257	5	4713 ( $\epsilon$ )	0,485
	100	1636	819	5756	1198	2	3733 ( $\epsilon$ )	0,187
	120	1636	819	5756	1142	1	2186	0,203
	140	1636	819	5756	1347	18	1165 ( $\epsilon$ )	0,719
	160	1636	819	5756	1644	26	469 ( $\epsilon$ )	0,703
	200	1636	819	5756	887	0	13	0,141
	273	1636	819	5756	365	0	0	0,031

Os resultados apresentados confirmam a importância da formulação no processo de resolução do problema.

## CAPÍTULO 6

### CONCLUSÃO

Este trabalho apresentou o desenvolvimento de um algoritmo baseado no método *branch-and-price* para resolver problemas de localização de  $p$ -medianas de grande porte, formulados como um problema de cobertura de conjuntos. A aplicação da decomposição de Dantzig-Wolfe permitiu definir um problema mestre restrito e um subproblema gerador de colunas. O multiplicador lagrangeano/*surrogate* foi utilizado para modificar o critério de seleção das colunas a comporem a base nas iterações seguintes, permitindo escolher um número maior de colunas mais produtivas. O controle de tamanho do problema permitiu trabalhar com uma base com menos colunas, tornando mais rápida a resolução dos problemas em cada nó da árvore.

Os resultados obtidos com dados de instâncias de problemas de  $p$ -medianas da *OR-Library* permitem concluir que os algoritmos de geração de colunas e *branch-and-price* desenvolvidos neste trabalho são mais eficientes que abordagens de geração de colunas baseadas em relaxação lagrangeana tradicional. Também é possível concluir, com base nos resultados apresentados, que o uso combinado de geração de colunas com a relaxação lagrangeana/*surrogate* produz resultados superiores aos obtidos pelas heurísticas lagrangeanas baseadas em otimização de subgradientes mencionadas neste trabalho, mesmo para instâncias onde a relação entre o número de vértices e o número de facilidades a serem localizadas não sejam favoráveis às abordagens baseadas em geração de colunas.

Para o caso de problemas de máxima cobertura, a vantagem da relaxação lagrangeana/*surrogate* sobre a relaxação lagrangeana tradicional ficou ainda mais evidente, mas os elevados tempos computacionais obtidos em ambos casos indicam que melhorias devem ser executadas no procedimento de geração de colunas para que a metodologia aqui desenvolvida seja uma alternativa competitiva.

As características da implementação do algoritmo de geração de colunas favorecem a obtenção de colunas com custo nulo, o que aumenta a ocorrência de degeneração, dificultando a convergência do mesmo. Abordagens baseadas no método *branch-and-bound* aplicadas às formulações *PPM* e *PMC* permitem a obtenção da solução ótima – e em menos

tempo computacional – na quase totalidade das instâncias utilizadas. Entretanto, é justamente nas outras instâncias, quando os recursos computacionais disponíveis não são suficientes sequer para a obtenção de uma solução intermediária, que uma formulação que trabalhe com um subconjunto das variáveis do problema pode, ao menos, fornecer um limitante inferior que pode ser utilizado para a poda da árvore de busca em métodos baseados em *branch-and-bound*.

## 6.1 Trabalhos Futuros

A pesquisa iniciada neste trabalho pode ser complementada. A seguir, relacionam-se alguns aspectos que demandam uma investigação mais detalhada, podendo gerar novos trabalhos e novas linhas de pesquisa:

- As estratégias de controle de ampliação e perturbação desenvolvidas para cálculo de limitantes das instâncias de problemas de máxima cobertura podem ser incluídas em implementações futuras do algoritmo *branch-and-price*, podendo ser ativadas ou não, dependendo da escolha do usuário.
- Estudar a influência do conjunto inicial de colunas.
- Investigar novas formas de cálculo do conjunto inicial de colunas, visando obter soluções duais iniciais de melhor qualidade, diminuindo o esforço computacional necessário à resolução do problema.
- Avaliar formas de ponderar diferentes critérios de parada do algoritmo de geração de colunas, visando melhorar a qualidade das colunas finais e aumentar o desempenho do algoritmo *branch-and-price*.
- Pesquisar novas estratégias de ramificação.
- Modificar a implementação do código, proporcionando ao usuário definir estratégia de exploração da árvore binária em profundidade (implementação atual), largura, ou a combinação dos dois (primeiro em profundidade, visando obter soluções viáveis, depois em largura, para melhoria da solução). Também pode ser explorada a implementação da estratégia de fila de prioridades.

- Desenvolver novos mecanismos de geração de colunas iniciais e intermediárias para problemas de máxima cobertura, visando minimizar a degeneração e acelerar a convergência do algoritmo na obtenção de soluções viáveis ou limitantes de melhor qualidade.
- Investigar a aplicação do algoritmo desenvolvido a outros problemas de localização que também possam ser formulados como um *PPM*.
- Estudar a possibilidade de utilizar a relaxação lagrangeana/*surrogate* em conjunto com outros métodos de estabilização.
- Desenvolvimento de um algoritmo *branch-and-cut-and-price* baseado na formulação do *PPM* para resolver problemas de máxima cobertura e relacionados.
- Desenvolvimento de ferramentas de apoio à decisão baseadas em Sistemas de Informação Geográfica para tratamento de problemas reais de *p*-medianas e de máxima cobertura, através da agregação ou combinação de várias técnicas de resolução (geração de colunas, *branch-and-price*, *branch-and-cut-and-price*, otimização de subgradientes, algoritmos populacionais, meta-heurísticas).



## REFERÊNCIAS BIBLIOGRÁFICAS

Arakaki, R.G.I.; Lorena, L.A.N. A constructive genetic algorithm for the maximal covering location problem. In: Metaheuristics International Conference (MIC'2001), 4., 16-20 jul., Porto, Portugal. **Proceedings...** Porto: Kluwer, 2001. p. 13-17.

Barnhart, C.; Johnson, E.L.; Nemhauser, G.L.; Savelsbergh, M.W.P.; Vance, P.H. Branch-and-price: column generation for solving huge integer programs. **Operations Research**, v. 46, p. 316-329, 1998.

Bazaraa, M.S.; Jarvis, J.J.; Sherali, H.D. **Linear programming and network flows**. 2<sup>a</sup> edição, New York: John Wiley & Sons, 1990. 684p.

Beasley, J.E. OR-Library: distributing test problems by electronic mail. **Journal of the Operational Research Society**, v. 41, p. 1069-1072, 1990.

Beasley, J.E. Lagrangean heuristics for location problems. **European Journal of Operational Research**, v. 65, p. 383-399, 1993.

Ceselli, A.; Righini G. A branch-and-price algorithm for the capacitated  $p$ -median problem. University of Milano, 2002. Disponível em:  
<http://sansone.crema.unimi.it/~righini/Papers/CPMP.pdf>

Chiyoshi, F.Y.; Galvão, R.D. A statistical analysis of simulated annealing applied to the  $p$ -median problem. **Annals of Operations Research**, v. 96, p. 61-74, 2000.

Christofides, N.; Beasley, J.E. A tree search algorithm for the  $p$ -median problem. **European Journal of Operational Research**, v. 10, p. 196-204, 1982.

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

Church, R.L.; Stoms, D.M.; Davis, F.W. Reserve selection as a maximal covering location problem. **Biological Conservation**, v. 76, p. 105-112, 1996.

Cornuejols, G.; Fisher, M.L.; Nemhauser, G.L. Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. **Management Science**, v.23, p. 789-810, 1977.

Correa, E.S.; Steiner, M.T.A.; Freitas A.A.; Carnieri, C. A genetic algorithm for the  $p$ -median problem. In: Genetic and Evolutionary Computation Conference (GECCO 2001), 7-11 jul. San Francisco, United States. **Proceedings...**, San Francisco: Morgan Kauffman, p. 1268-1275, 2001.

Dantzig, G.B.; Wolfe, P. Decomposition principle for linear programs. **Operations Research**, v. 8, p. 101-111, 1960.

Daskin, M. **Network and discrete location: models, algorithms and applications**. New York: Wiley Interscience, 1995. 500p.

Day, P.R.; Ryan, D.M. Flight attendant rostering for short-haul airline operations. **Operations Research**, v. 45, p. 649-661, 1997.

Desrochers, M.; Soumis, F. A column generation approach to the urban transit crew scheduling problem. **Transportation Science**, v. 23, p. 1-13, 1989.

Desrochers, M.; Desrosiers, J.; Solomon, M. A new optimization algorithm for the vehicle routing problem with time windows. **Operations Research**, v. 40, p. 342-354, 1992.

Desrosiers, J.; Lübbecke, M.E. **A primer in column generation**. Montreal: HEC, 2004. 26 p. (Les Cahiers du GERAD G-2004-02).

Drezner, Z. (Editor) **Facility location: a survey of applications and methods**. New York: Springer-Verlag, 1995. 571p.

Dyer, M.E. Calculating surrogate constraints. **Mathematical Programming**, v. 19, p. 255-278, 1980.

El-Shaieb, A.M. (1973). A new algorithm for locating sources among destinations. **Management Science**, v. 20, p. 221-231, 1973.

- Erkut E.; Bozkaya, B.; Zhang, J. **An effective genetic algorithm for the  $p$ -median problem**. Alberta: University of Alberta, 1997, 23 p. (Working Paper 97-2).
- Fung, G.; Mangasarian, O.L. Semi-supervised support vector machines for unlabeled data classification. **Optimization Methods and Software**, v. 1, nº 15, p. 29-44, 2000.
- Galvão, R.D. A dual-bounded algorithm for the  $p$ -median problem. **Operations Research**, v. 28, p. 1112-1121, 1980.
- Galvão, R.D. A note on Garfinkel, Neebe and Rao's decomposition for the  $p$ -median problem. **Transportation Science**, v. 15, nº 3, p. 175-182, 1981.
- Galvão, R.D. Uncapacitated facility location problems: contributions. **Pesquisa Operacional**. v. 24, nº 1, 2004.
- Galvão, R.D.; Espejo, L.G.A.; Boffey, B. A comparison of Lagrangean and surrogate relaxations for the maximal covering location problem. **European Journal of Operational Research**, v. 124, p. 377-389, 2000.
- Galvão, R.D.; Raggi, L.A. A method for solving to optimality uncapacitated location problems. **Annals of Operations Research**, v. 18, p. 225-244, 1989.
- GAMS Development Corporation, **GAMS: a user's guide**. Washington, 1998, 262p.
- Garey, M.R.; Johnson, D.S. **Computers and intractability: a guide to the theory of NP-completeness**. San Francisco: W. H. Freeman and Co., 1979. 340p.
- Garfinkel, R.S.; Neebe, W.; Rao, M.R. An algorithm for the  $m$ -median location problem. **Transportation Science**, v. 8, p. 217-236, 1974.
- Gilmore, P.C.; Gomory, R.E. A linear programming approach to the cutting stock problem. **Operations Research**, v. 9, p. 849-859, 1961.
- Gilmore, P.C.; Gomory, R.E. A linear programming approach to the cutting stock problem: Part II. **Operations Research**, v. 11, p. 863-888, 1963.
- Glover, F. Surrogate constraints. **Operations Research**, v. 16, p. 741-749, 1968.

Hakimi, S. L. Optimum location of switching centers and the absolute centers and medians of a graph. **Operations Research**, v. 12, p. 450-459, 1964.

Hakimi, S. L. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. **Operations Research**, v. 13, p. 462-475, 1965.

Hansen, P.; Jaumard, B. Cluster analysis and mathematical programming. **Mathematical Programming**, v. 79, p. 191-215, 1997.

Hansen, P.; Mladenovic, N. Variable neighborhood search for the  $p$ -median. **Location Science**, v. 5, p. 207-226, 1997.

Hansen, P.; Mladenovic, N.; Perez-Brito, D. Variable neighborhood decomposition search. **Journal of Heuristics**, v. 7, p. 335-350, 2001.

Held, M.; Karp, R.M. The traveling-salesman problem and minimum spanning trees: part II. **Mathematical Programming**, v. 1, p. 6-25, 1971.

Hillsman, E.L. The  $p$ -median structure as a unified linear model for location-allocation analysis. **Environmental and Planning A**, v. 16, p. 305-318, 1984.

Hosage, C.M.; Goodchild, M.F. Discrete space location-allocation solutions from genetic algorithms. **Annals of Operations Research**, v. 6, p. 657-682, 1986.

ILOG CPLEX 7.1 **User's Manual**. ILOG Inc., CPLEX Division, 2001.

Järvinen, P.J.; Rajala, J.; Sinervo, H. A branch and bound algorithm for seeking the  $p$ -median. **Operations Research**, v. 20, p. 173-178, 1972.

Jensen, F. A dynamic programming algorithm for cluster analysis. **Operations Research**, v. 17, p. 1034-1057, 1969.

Karwan, M.L.; Rardin, R.L. Some relationships between lagrangean and surrogate duality in integer programming. **Mathematical Programming**, v. 17, p. 320-334, 1979.

Kelley, J.E. The cutting plane method for solving convex programs. **Journal of the SIAM**, v. 8, p. 703-712, 1960.

- Lorena, L.A.N.; Furtado, J.C. Constructive genetic algorithm for clustering problems. **Evolutionary Computation**, v. 9, nº 3, p. 309-327, 2001.
- Lorena, L. A. N.; Pereira, M. A. A Lagrangean/surrogate heuristic for the maximal covering location problem using Hillsman's edition. **International Journal of Industrial Engineering**, v. 9, nº 1, p. 57-67, 2002.
- Lorena, L.A.N.; Pereira, M.A.; Salomão, S.N.A. A relaxação lagrangeana/*surrogate* e o método de geração de colunas: novos limitantes e novas colunas. **Pesquisa Operacional**, v.23, nº 1, p. 29-47, 2003.
- Lorena, L.A.N.; Senne, E.L.F. Improving traditional subgradient scheme for lagrangean relaxation: an application to location problems. **International Journal of Mathematical Algorithms**, v. 1, p. 133-151, 1999.
- Lübbecke, M.E.; Desrosiers, J. **Selected topics in column generation**. Montreal: HEC, 2002. 32 p. (Les Cahiers du GERAD G-2002-64).
- Maranzana, F.E. On the location of supply points to minimize transport costs. **Operational Research Quarterly**, v. 15, p. 261-270, 1964.
- Meneses, C.N.; de Souza, C.C. **Exact solutions of rectangular partitions via integer programming**. Campinas: UNICAMP, 1998. 48p. (IC-98-35).
- Minoux, M. A class of combinatorial problems with polynomially solvable large scale set covering/partitioning relaxations. **R.A.I.R.O. Recherche Opérationnelle**, v. 21, nº 2, p. 105-136. 1987.
- Mirchandani, P.B.; Oudjit, A.; Wong, R.T. Multidimensional extensions and a nested dual approach for the  $m$ -median problem. **European Journal of Operational Research**, v. 21, p. 121-137, 1985.
- Mladenovic, N.; Moreno, J.P.; Moreno-Vega, J. A chain-interchange heuristic method. **Yugoslav Journal of Operations Research**, v. 6, p. 41-54, 1996.
- Mulvey, J.M.; Crowder, H.P. Cluster analysis: an application of lagrangian relaxation. **Management Science**, v.25, p. 329-340, 1979.

Narciso, M.G. **A relaxação lagrangeana/surrogate e algumas aplicações em otimização combinatória**. 1998. 134p. Tese (Doutorado em Computação Aplicada) Instituto Nacional de Pesquisas Espaciais, São José dos Campos. 1998.

Narciso, M.G.; Lorena, L.A.N. Lagrangean/surrogate relaxation for generalized assignment problem. **European Journal of Operational Research**, v. 114, p. 165-177, 1999.

Narula, S.C.; Ogbu, U.I.; Samuelsson, H.M. An algorithm for the  $p$ -median problem. **Operations Research**, v. 25, p. 709-713, 1977.

Neame, P.J. **Nonsmooth dual methods in integer programming**. 1999. 172p. Tese de Doutorado. University of Melbourne, Melbourne, 1999.

Neebe, A.W. A branch and bound algorithm for the  $p$ -median transportation problem. **Journal of the Operational Research Society**, v. 29, p. 989-995, 1978.

Nemhauser, G.L.; Wolsey L.A. **Integer and combinatorial optimization**. New York: Wiley-Interscience series in discrete mathematics and optimization, 1999, 764p.

Parker, R.G.; Hardin, R.L. **Discrete Optimization**. New York: Academic Press, 1988, 472p.

Rao, M.R. Cluster analysis and mathematical programming. **Journal of the American Statistical Association**, v. 66, p. 622-626, 1971.

Reinelt, G. The traveling salesman problem: computational solutions for TSP applications. **Lecture Notes in Computer Science**, v. 840, Berlin: Springer-Verlag, 1994.

Resende, M.G.C. Computing approximate solutions of the maximum covering problem using GRASP. **Journal of Heuristics**, vol. 4, p. 161-171, 1998.

Resende, M.G.C.; Werneck, R.F. **On the implementation of a swap based local search procedure for the  $p$ -median problem**. AT&T Labs Research, 2002a. (TD-5E4QKA).

Resende, M.G.C.; Werneck, R.F. **A GRASP with path-relinking for the  $p$ -median problem**. AT&T Labs Research, 2002b. 22p. (TD-5E53XL).

ReVelle, C.S.; Swain, R.W. Central facilities location. **Geographical Analysis**, v. 2, p. 30-42, 1970.

Rolland, E.; Schilling, D.A.; Current, J.R. An efficient tabu search procedure for the  $p$ -median problem. **European Journal of Operational Research**, v. 96, p. 329-342, 1996.

Rosing, K.E.; ReVelle, C.S. Heuristic concentration: two stage solution construction. **European Journal of Operational Research**, v. 97, p. 75-86, 1997.

Ryan, D.M; Foster, B.A. An integer programming approach to scheduling. In: **Computer Scheduling of Public Transport, Urban Passenger Vehicle and Crew Scheduling** (A. Wren,ed.), Amsterdam: North Holland, p. 269-280, 1981.

Scaparra, M.P; Church, R.L. **An optimal approach for the interdiction median problem with fortification**. Canterbury: Kent Business School, 2005. 34p. (Working Paper 78).

Schilling, D.A.; Rosing, K.E.; ReVelle, C.S. Network distance characteristics that affect computational effort in  $p$ -median location problems. **European Journal of Operational Research**, v. 127, p. 525-536, 2000.

Senne, E.L.F.; Lorena, L.A.N. 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, pp. 115-130, 2000.

Senne, E.L.F.; Lorena, L.A.N. Stabilizing column generation using lagrangean/surrogate relaxation: an application to  $p$ -median location problems. In: European Operations Research Conference – EURO XVIII, 9-11 julho 2001, Erasmus University Rotterdam, The Netherlands. Disponível em: <http://www.lac.inpe.br/~lorena/ejor/EURO2001.pdf>

Senne, E.L.F.; Lorena, L.A.N.; Pereira, M.A. A branch-and-price approach to  $p$ -median location problems. **Computers & Operations Research**, v. 32, n<sup>o</sup>. 6, p. 1655-1664, 2005.

Swain, R.W. A parametric decomposition approach for the solution of uncapacitated location problems. **Management Science**, v. 21, p. 955-961, 1974.

Teitz, M. B.; Bart, P. Heuristic methods for estimating the generalized vertex median of a weighted graph. **Operations Research**, v. 16, p. 955-961, 1968.

Toregas, C.; Swain, R.; ReVelle, C.; Bergman, L. The location of emergency service facilities. **Operations Research**, v. 19, p. 1363-1373, 1971.

Tutschku, K. **Demand-based radio network planning of cellular mobile communication systems**. Würzburg: University of Würzburg, 1997. 23p. (Report nº 177).

Valério de Carvalho, J.M. Exact solution of bin-packing problems using column generation and branch-and-bound. **Annals of Operations Research**, v. 86, p. 629-659, 1999.

Vance, P.H.; Barnhart, C.; Johnson, E.L.; Nemhauser, G.L. Solving binary cutting stock problems by column generation and branch-and-bound. **Computational Optimization and Applications**, v. 3, p. 111-130, 1994.

Vinod, H.D. Integer programming and the theory of groups. **Journal of the American Statistical Association**, v. 64, p. 506-519, 1969.

Voss, S. A reverse elimination approach for the  $p$ -median problem. **Studies in Locational Analysis**, v. 8, p. 49-58, 1996.

Whitaker, R.A. A fast algorithm for the greedy interchange for large-scale clustering and median location problems. **INFOR**, v. 21, p. 95-108, 1983.

Wilhelm, W.E. A technical review of column generation in integer programming. **Optimization and Engineering**, v. 2, p. 159-200, 2001.

Wolsey, L.A. **Integer Programming**. New York: Wiley-Interscience, 1998, 260p.

Yunes, T.H.; Moura, A.V.; de Souza, C.C. **Hybrid column generation approaches for solving real world crew management problems**. Campinas: UNICAMP, 2000a. 38p. (IC-00-18).

Yunes, T.H.; Moura, A.V.; de Souza, C.C. Solving very large crew scheduling problems to optimality. In: Symposium on Applied Computing, 19-21 mar. Como, Italy. **Proceedings...**, Como: ACM, 2000b, p. 446-451 (ISBN:1-58113-240-9).