2
Problema das p-Medianas

2.1
Definição

O PMNC é definido da seguinte forma: determinar quais p facilidades ($p \leq m$, onde m é o número de pontos onde podem ser abertas facilidades) devem obrigatoriamente ser abertas com o intuito de minimizar a soma das distâncias de cada cliente a facilidade aberta mais próxima do mesmo. Portanto, nós queremos minimizar o custo de servir todos os clientes para um dado valor de p. Como este problema é NP-Difícil (Kariv e Hakimi, 1979), é improvável existir um algoritmo com um número de passos polinomial no tamanho da entrada para resolvê-lo, isto é, encontrar uma solução ótima e provar sua otimalidade.

A Figura 2.1 ilustra uma solução para o PMNC para uma instância contendo 50 clientes (n = 50) e 16 potenciais localização de facilidades (m = 16), sendo obrigatório a abertura de exatamente cinco facilidades (p = 5).

2.2

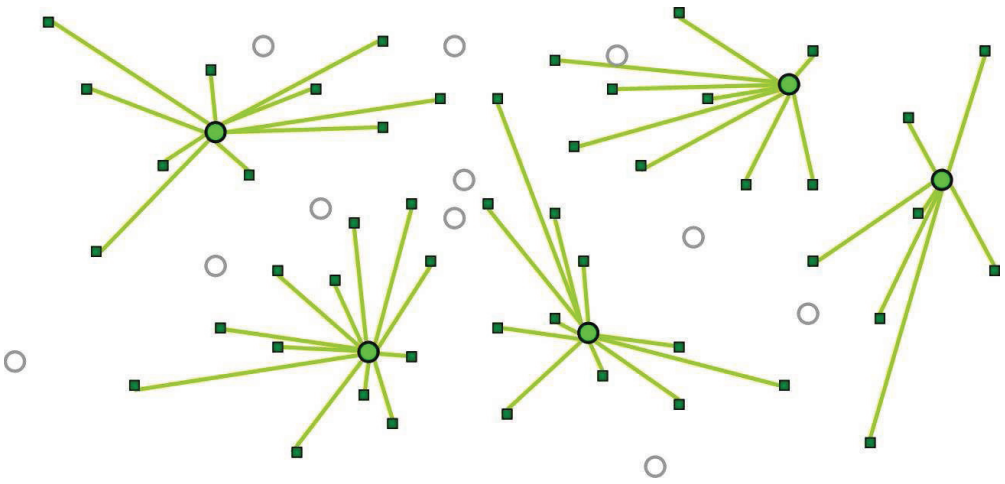

Figura 2.1: Exemplo de solução para o PMNC

Formulações

Nesta seção introduziremos as formulações como programa linear inteiro utilizadas por alguns métodos aplicados ao PMNC. Introduções à Programação Linear e Programação Linear Inteira podem ser encontradas em (Chvatal, 1983) e (Bertsimas e Tsitsiklis, 1997). Iniciaremos com uma formulação do PMNC como problema de programação linear inteira, aqui chamada de problema primal, e apresentaremos, o problema dual da sua relaxação linear. Em seguida, mostramos como o problema dual pode ter seu número de variáveis reduzido, o que é explorado nos métodos de resolução. Finalmente, descrevemos as instâncias que usualmente são utilizadas para medir a eficiência e a eficácia dos algoritmos propostos para a resolução do PMNC.

2.2.1 Formulação Primal

Existem algumas maneiras distintas de se formular o PMNC como um problema de programação inteira. Porém, a formulação mais utilizada será descrita a seguir. Esta formulação foi proposta em (ReVelle e Swain, 1970). Os métodos utilizados ao longo desta dissertação que fazem uso da programação inteira e linear empregam estas formulações. Antes de apresentarmos a formulação primal para o PMNC, definiremos as seguintes variáveis de decisão:

$$x_{ij} = \begin{cases} 1 & \text{se o cliente } i \text{ é atendido pela facilidade localizada em } j \\ 0 & \text{caso contrário} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{se a facilidade é aberta no local } j; \\ 0 & \text{caso contrário} \end{cases}$$

A seguir é apresentada a formulação primal inteira.

$$\min \sum_{i \in U} \sum_{j \in F} d_{ij} x_{ij} \tag{2-1}$$

sujeito a

$$\begin{aligned} \sum_{j \in F} x_{ij} &= 1, \forall i \in U && \text{(2-2)} \\ y_j - x_{ij} &\geq 0, \forall i \in U, \forall j \in F && \text{(2-3)} \\ \sum_{j \in F} y_j &= p && \text{(2-4)} \\ x_{ij}, y_j &\in \{0,1\}, \forall i \in U, \forall j \in F && \text{(2-5)} \end{aligned}$$

Nesta formulação, F é o conjunto contendo os possíveis pontos de abertura de uma facilidade ($|F| = m$) e U é o conjunto contendo os pontos onde estão localizados os clientes ($|U| = n$). A função objetivo (2-1) minimiza a distância total de atribuição de clientes às facilidades, ou seja, minimiza a soma dos custos de servir os clientes com a facilidade aberta mais próxima. As restrições (2-2) expressam que cada cliente deve ser designado a exatamente uma única facilidade. As restrições (2-3) previnem que um cliente seja designado a uma potencial localização de facilidade que não tenha sido aberta. As restrições (2-4) forçam a abertura de exatamente p facilidades. Por fim, as restrições (2-5) expressam a integralidade e não negatividade das variáveis de decisão.

A relaxação linear da formulação primal descrita acima é obtida substituindo-se as restrições (2-5) pelas restrições

$$x_{ij} \geq 0, \forall i \in U, \forall j \in F \tag{2-6}$$

$$y_j \geq 0, \forall j \in F \tag{2-7}$$

A solução ótima da relaxação linear de qualquer problema de minimização formulado com programação inteira fornece um limite inferior para todas as soluções viáveis da respectiva formulação inteira. Isto se justifica pelo fato de que, como se trata de um problema de minimização, a remoção de qualquer restrição diminuirá o valor da solução obtida ou, no máximo, manterá a solução inalterada se a restrição eliminada for redundante.

2.2.2 Formulação Dual

Abaixo é exposta a formulação dual para o PMNC obtida a partir da relaxação linear da formulação primal descrita anteriormente. Da teoria da *dualidade*, consultar (Chvatal, 1983) e (Bertsimas e Tsitsiklis, 1997), temos que todo problema de programação linear, problema primal, que possui uma solução ótima e finita possui um problema dual correspondente cujo valor da solução ótima de ambos os problemas são idênticos (*Teorema da Dualidade Forte*). Este teorema será ilustrado nos algoritmos duais apresentados nesta dissertação. A seguir é apresentada a formulação dual para o PMNC.

$$\max \sum_{i \in U} \lambda_i + \gamma p \tag{2-8}$$

sujeito a

$$\lambda_i - \mu_{ij} \leq d_{ij}, \forall i \in U, \forall j \in F \quad (2\text{-}9)$$

$$\sum_{i \in U} \mu_{ij} + \gamma \leq 0, \forall j \in F \quad (2\text{-}10)$$

$$\mu_{ij} \geq 0, \forall i \in U, \forall j \in F \quad (2\text{-}11)$$

Para a construção desta formulação relacionou-se as seguintes variáveis duais às seguintes restrições da formulação primal:

- Cada variável dual λ_i é associada a cada restrição (2-2)
- Cada variável dual μ_{ij} é associada a cada restrição (2-3)
- A variável dual γ é associada a restrição (2-4)

A função objetivo (2-8) maximiza a soma das variáveis duais λ_i mais um fator γp. As restrições (2-9) e (2-10) serão descritas utilizando o conceito de *custo reduzido*. O custo reduzido $\overline{c}_\pi(ij)$ de uma atribuição cliente–facilidade (i, j) em relação a uma solução dual π é definido como:

$$\overline{c}_\pi(ij) = d_{ij} - \lambda_i + \mu_{ij}, \forall i \in U, \forall j \in F \quad (2\text{-}12)$$

O custo reduzido $\overline{c}_\pi(j)$ de uma facilidade j em relação a uma solução dual π é definido como:

$$\overline{c}_\pi(j) = -\gamma - \sum_{i \in U} \mu_{ij}, \forall j \in F \quad (2\text{-}13)$$

Assim, as restrições (2-9) . A restrição (2-10) . Por fim, as restrições (2-11) expressam a não negatividade das variáveis duais μ_{ij}. As variáveis duais $\lambda_i, \forall i \in U$, e γ são irrestritas. Os custos reduzidos serão empregados em técnicas de *fixação de variáveis por custo reduzido* descrita posteriormente neste trabalho.

2.2.3 Formulação Dual Condensada

Abaixo é exposta a formulação dual condensada para o PMNC obtida a partir da formulação dual descrita na Seção 2.2.2, conforme apresentado em (Captivo, 1991).

$$\max \sum_{i \in U} \lambda_i + \gamma p \quad (2\text{-}14)$$

sujeito a

$$\sum_{i \in U} \max(0, \lambda_i - d_{ij}) \quad \leq \quad -\gamma, \forall j \in F \tag{2-15}$$

Para a construção desta formulação foi feita a seguinte simplificação:

$$\mu_{ij} \quad = \quad \max(0, \lambda_i - d_{ij}), \forall i \in U, \forall j \in F \tag{2-16}$$

A função objetivo (2-14) continua a mesma daquela da formulação dual descrita na Seção 2.2.2 e tenciona maximizar a soma das variáveis duais λ_i mais um fator γp. As restrições (2-15) serão descritas utilizando o conceito de custo reduzido idêntica aquela descrição utilizado na Seção 2.2.2, porém de forma condensada devido ao fato de estarmos considerando a formulação dual condensada. Assim, o custo reduzido $\overline{c}_\pi(ij)$ de uma atribuição cliente–facilidade (i, j) em relação a uma solução dual π é definido como:

$$\overline{c}_\pi(j) \quad = \quad -\gamma - \sum_{i \in U} \max(0, \lambda_i - d_{ij}), \forall j \in F \tag{2-17}$$

As restrições (2-17) . Contudo, as variáveis duais $\lambda_i, \forall i \in U$, e γ continuam irrestritas.

2.3
Problemas Teste

Nesta seção tratamos da descrição das instâncias de teste utilizadas para o PMNC. Em seguida, apresentamos as métricas utilizadas para a avaliação dos métodos implementados e, por fim, descrevemos o ambiente onde foram realizados os testes.

2.3.1
Instâncias de Teste

Nas subseções seguintes serão descritas as classes de instâncias de teste empregadas nos métodos utilizados neste trabalho. Estas classes são as seguintes: *OR-Library* e *TSP-Library*.

OR-Library

OR-Library é uma coleção de instâncias teste para uma variedade de problemas da área da Pesquisa Operacional. Esta classe foi introduzida em (Beasley, 1990). Para o PMNC, o repositório disponibiliza 40 instâncias teste que foram publicadas em (Beasley, 1985). A cardinalidade do conjunto de pontos destas instâncias está contida no intervalo [100,900], onde cada ponto pode assumir o papel de um cliente ou de uma facilidade (fator dependente da configuração da solução do PMNC). Cada instância possui um valor presente no intervalo [5,200] para o parâmetro p e possui, também, um conjunto de arestas com seus respectivos pesos. Soluções ótimas são conhecidas para todas as instâncias desta classe (Beasley, 1985).

Após a leitura das instâncias da classe *OR-Library*, precisamos aplicar o algoritmo de *Floyd Warshall* descrito em (Cormen et al., 2001) sobre a matriz de distâncias contida no arquivo de entrada de cada instância para obtermos a alocação completa da matriz de distâncias para cada instância. O algoritmo de *Floyd Warshall* é utilizado para encontrar caminhos mais curtos entre todos os pares de vértices em um grafo $G(V, A)$. Este algoritmo é executado em tempo $O(V^3)$. Para maiores detalhes sobre a utilização deste algoritmo sobre a matriz de distâncias, consultar (Beasley, 1990).

TSP-Library

TSP-Library (Reinelt, 1991) é uma coleção de instâncias teste para o problema do Caixeiro Viajante (*Symmetric Traveling Salesman Problem - TSP*) e para problemas relacionados. Esta classe possui diversas origens e diferentes tipos. As instâncias de grande porte para o PMNC são, em geral, obtidas no repositório da *TSP-Library* disponível na página citada em (TSPLib) e são aquelas disponibilizadas para o TSP onde os pontos estão dispostos no espaço bidimensional. A cardinalidade do conjunto de pontos destas instâncias está contido no intervalo [51,18512], onde cada ponto pode assumir o papel de um cliente ou de uma facilidade (fator dependente da configuração da solução do PMNC). O parâmetro p para cada instância é definido em função da cardinalidade do conjunto de pontos. Soluções ótimas são conhecidas para algumas instâncias desta classe.

2.3.2 Métricas

As métricas utilizadas para a avaliação dos métodos implementados foram:

– *Gap* : Refere-se a diferença percentual entre o valor de uma solução primal encontrada FO_{PE} e o valor de uma solução dual encontrada FO_{DE}. O *Gap* é calculado da seguinte forma:

$$Gap = ((FO_{PE} - FO_{DE}) \times 100) \div FO_{DE} \quad (2\text{-}18)$$

– *Desvio primal (dP)*: Refere-se a diferença percentual entre o valor de uma solução primal encontrada FO_{PE} e o valor de outra solução primal que é utilizada como referência FO_{PR}. A solução primal utilizada como referência é a solução ótima para o problema ou, caso a solução ótima não seja conhecida, é a melhor solução primal conhecida na literatura. O desvio primal dP é calculado da seguinte forma:

$$dP = ((FO_{PE} - FO_{PR}) \times 100) \div FO_{PR} \quad (2\text{-}19)$$

– *Desvio dual (dD)*: Refere-se a diferença percentual entre o valor de uma solução dual encontrada FO_{DE} e o valor de uma solução primal que é utilizada como referência FO_{PR}. A solução primal utilizada como referência é a solução ótima para o problema ou, caso a solução ótima não seja conhecida, é a melhor solução primal conhecida na literatura. O desvio dual dD é calculado da seguinte forma:

$$dD = ((FO_{PR} - FO_{DE}) \times 100) \div FO_{DE} \quad (2\text{-}20)$$

– *Desvio da relaxação linear (dR)*: Refere-se a diferença percentual entre o valor de uma solução dual encontrada FO_{DE} e o valor da solução ótima da relaxação linear FO_{RL}. O desvio da relaxação linear dP é calculado da seguinte forma:

$$dR = ((FO_{RL} - FO_{DE}) \times 100) \div FO_{DE} \quad (2\text{-}21)$$

2.3.3 Ambiente de Teste

Os algoritmos implementados foram desenvolvidos na linguagem C++, utilizando o ambiente de desenvolvimento Microsoft Visual Studio 2005. Os testes foram executados em um microcomputador com processador Intel Core 2 Duo E4500, 2.20 GHz, 2 GB de memória RAM, sob o sistema operacional Microsoft Windows XP Professional, versão 2002, Service Pack 3.