3
Métodos de Resolução

Nesta seção faremos uma revisão bibliográfica dos principais métodos de resolução para o PMNC existentes na literatura. Porém, antes de apresentarmos estes métodos, iremos explicar as categorias as quais cada método se enquadra. Estas categorias são as seguintes: *algoritmos primais*, *duais* e *exatos*. A Seção 3.1 faz uma breve introdução sobre estas categorias e introduz alguns conceitos utilizados ao longo deste trabalho. A Seção 3.2 trata dos algoritmos baseados em técnicas *heurísticas* e *metaeurísticas* para o tratamento do PMNC. Esta categoria será chamada de *algoritmos primais*. Na Seção 3.3 são abordados algoritmos baseados em técnicas heurísticas, porém diferentemente dos algoritmos primais, elas trabalham sobre o problema dual. O objetivo de se obter soluções duais válidas de alta qualidade é conhecer limites válidos para o menor valor que uma solução para o PMNC pode ter, permitindo se medir a qualidade das soluções obtidas pelos *algoritmos primais*. Esta categoria será chamada de *algoritmos duais*. Finalizando, a Seção 3.4 discute métodos que garantidamente encontram a melhor solução para o PMNC, fato que motivou a classificação destes métodos como *algoritmos exatos*.

3.1
Introdução

Os algoritmos primais tem como objetivo encontrar soluções que atendam as restrições (2-1) a (2-5). Chamaremos de *soluções primais viáveis* as soluções que respeitam estas restrições. Os algoritmos primais foram subdivididos nas seguintes subcategorias: *heurísticas construtivas*, *heurísticas de refinamento* e *metaeurísticas*. Ambas as abordagens utilizam heurísticas para solucionar os problemas e estas técnicas serão detalhadas nas seções 3.2.1, 3.2.2 e 3.2.3, respectivamente. Heurísticas são técnicas inspiradas em processos intuitivos e/ou matemáticos para procurar uma boa solução, eventualmente a ótima, a um custo computacional não proibitivo. Porém, estes métodos não garantem a otimalidade e não indicam quão próximo a solução obtida está da solução ótima. O grande desafio destes métodos, bem como dos algoritmos duais citados adiante, é gerar o mais rápido possível soluções tão próximas quanto possível da solução ótima, podendo fornecer a ótima. A criação destes métodos é, em geral, baseado em características peculiares ao problema a ser tratado. Assim, a maioria das heurísticas são de caráter específico, não sendo eficientes ou aplicáveis na resolução de uma classe mais ampla de problemas. Contudo, o estudo da aplicação destes métodos a diferentes problemas poderá fornecer *insights* para o tratamento de um problema particular.

Os algoritmos duais tem com meta encontrar soluções que atendam as restrições (2-8) a (2-11). Chamaremos estas soluções de *soluções duais viáveis*. Esta abordagem também utiliza heurísticas para solucionar os problemas. Conforme já observado, a grande diferença desta abordagem quando comparada com a abordagem via algoritmos primais está nas restrições trabalhadas pelos métodos. Diferentemente dos algoritmos primais, os algoritmos duais resolvem um problema derivada do problema original e as informações obtidas com esta resolução guiam na construção de uma solução para o problema original, fornecendo, também, uma certificação para a qualidade dos algoritmos primais. Esta certificação baseia-se na diferença percentual entre o valor da solução primal e o valor da solução dual, conforme explicado mais adiante.

Os algoritmos exatos são algoritmos que encontram a solução ótima e provam a sua otimalidade para qualquer instância de um problema de otimização em tempo finito. As técnicas deste tipo mais populares para o PMNC utilizam a enumeração implícita através de uma árvore de resolução, pois todas as variáveis que constituem o problema são inteiras. Porém, existem métodos exatos que recebem como entrada um programa linear e retornam uma solução ótima para o mesmo. O *algoritmo simplex*, para maiores detalhes consultar (Chvatal, 1983), se enquadra nesta categoria e possui complexidade de pior caso exponencial.

Conforma já observado, cada solução fornecida pelos métodos expostos acima é quantificada com um valor e cada problema possui uma ou mais soluções rotulada(s) como *solução(ões) ótima(s)*. As soluções ótimas possuem o mesmo valor e atendem da melhor forma possível todas as restrições que caracterizam o problema. No caso do PMNC, a formulação primal tem como objetivo minimizar uma função, chamada *função objetivo*, que é responsável por gerar este valor. Já a formulação dual tenciona maximizar outra função objetivo. Considerando o PMNC, qualquer solução primal viável é um *limite superior* para o valor da solução ótima e qualquer solução dual viável é um *limite inferior* para o valor da solução ótima. Na Figura 3.1, o eixo vertical rotulado como *Valor* ilustra os valores possíveis para as soluções de um problema de minimização. Neste eixo, indicamos o valor para a solução ótima e para uma solução primal e dual viáveis ,respectivamente, no centro, na parte

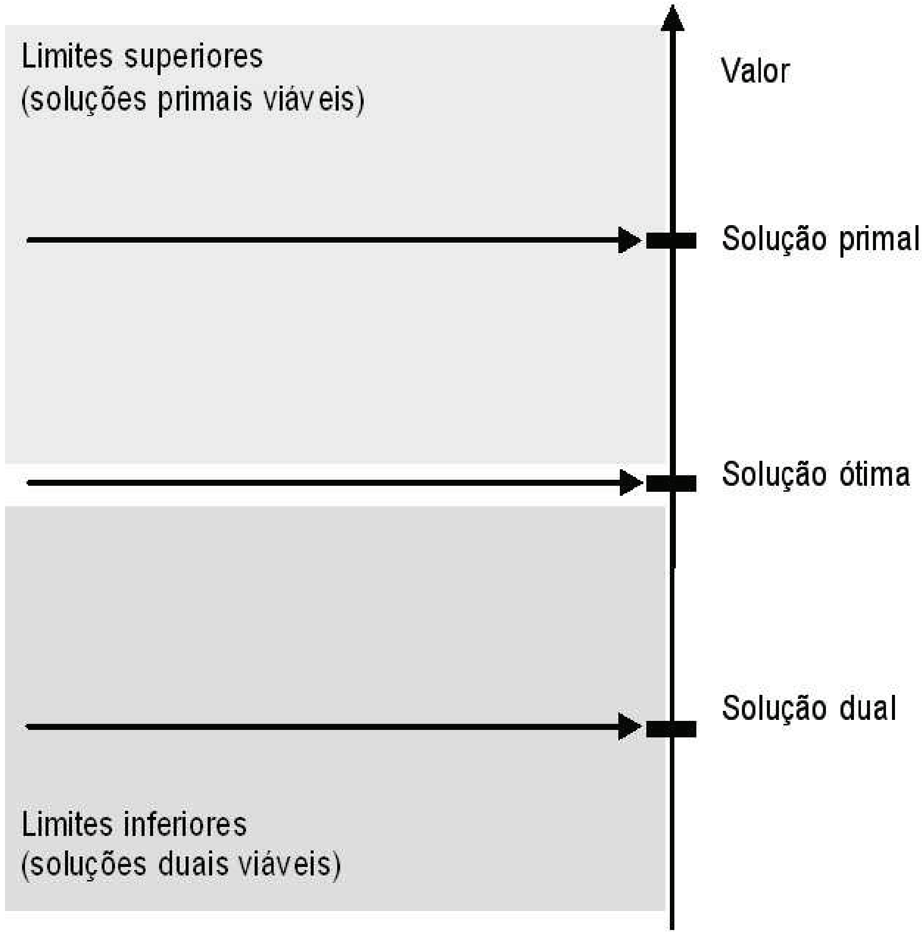

Figura 3.1: Solução ótima, limites superior e inferior para problemas de otimização

superior e na parte inferior deste eixo.

3.2 Algoritmos Primais

3.2.1 Heurísticas Construtivas

As heurísticas construtivas constroem uma solução viável para qualquer problema a partir do zero, ou seja, não utilizam nenhuma outra solução para guiar esta construção. Assim, a solução gerada por meio de uma heurística construtiva é chamada de *solução inicial*. Os demais algoritmos primais descritos nesta seção necessitam de uma solução de partida para poderem gerar soluções eventualmente melhores. Geralmente, a solução inicial é o ponto de partida para estes métodos. Porém, se tivermos acesso a uma solução já derivada da solução inicial, podemos utilizá-la como ponto de partida. Durante a construção de uma solução inicial, a cada passo é inserido um elemento nesta solução. Estes elementos podem ser inseridos pseudo-aleatoriamente ou podem ser inseridos utilizando uma certa inteligência. A forma de construção pseudo-aleatória é simples de se implementar, porém produz soluções de pior qualidade. Nas construções mais elaboradas, geralmente, é adotada uma função de avaliação para medir o benefício obtido com a inserção de um elemento durante a construção. Uma função de avaliação popularmente utilizada é a função gulosa originando o método guloso ascendente ((Whitaker, 1983) e (Kuehn e Hamburger, 1963)), que insere a cada passo o elemento que ocasiona o benefício máximo na configuração corrente da solução que está sendo construída. Outro tipo de método guloso existente é o guloso descendente que parte de uma solução contendo todas as facilidades abertas e, a cada iteração, fecha a facilidade de propicia o maior benefício com a sua retirada ((Feldman et al., 1966), (Moreno-Perez et al., 1991) e (Salhi e Atkinson, 1995)). Além destes métodos construtivos, temos o método *Híbrido* ((Moreno-Perez et al., 1991), (Captivo, 1991), (Pizzolato, 1994) e (Salhi, 1997)), que combina as idéias de dois ou mais métodos em um único método e heurísticas construtivas do tipo *Primal-Dual* ((Galvão, 1980), (Erlenkotter, 1978) e (Captivo, 1991)) que, a partir de uma solução dual, constroem uma solução primal viável. Todas estas técnicas são detalhadas na Seção 4.2.

3.2.2 Heurísticas de Refinamento

As heurísticas de *Refinamento* ou técnicas de *Busca Local* são métodos que se baseiam na noção de *vizinhança*. Mais detalhadamente, seja S o espaço de soluções viáveis de um problema de otimização e seja s uma solução tal que $s \in S$. A vizinhança de s é um conjunto de soluções $V \subseteq S$ definidas por *movimentos* realizados em s. Denomina-se movimento a modificação que transforma uma solução s em outra $s^{'}$ tal que $s^{'} \in V$. Cada solução $s^{'} \in V$ é chamada de vizinho de s. Contudo, técnicas de *Busca Local* partem de uma solução inicial qualquer s e a melhoram por meio da exploração sistemática do espaço de soluções utilizando o conceito de vizinhança. Para o PMNC, temos dois tipos de métodos de busca local citados na literatura. Estes métodos serão descritos a seguir.

Alternate

O primeiro método de busca local encontrado na literatura é chamado de *Alternate* (Maranzana, 1964). Inicialmente, este método seleciona arbitraria-

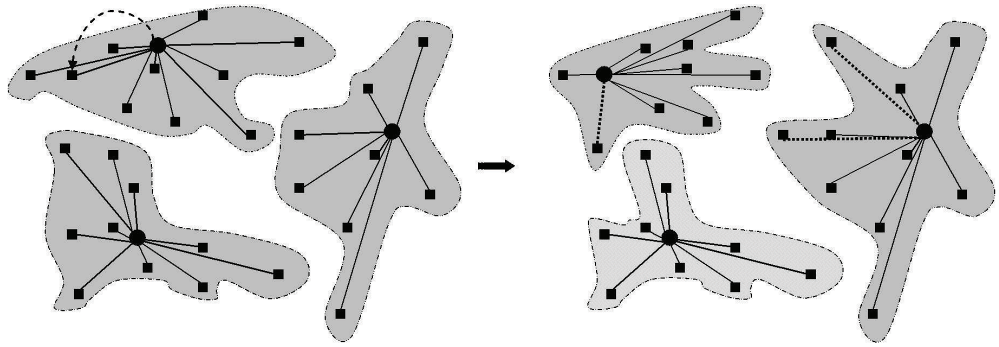

Figura 3.2: Exemplo de um movimento do tipo *Alternate*

mente p facilidades a serem abertas. Em seguida, são criadas p partições onde cada partição P_k $(k = 1, ..., p)$ possui como elemento central uma das facilidades abertas em conjunto com os clientes que são servidos por esta facilidade (os clientes contidos em cada partição são aqueles que possuem a facilidade desta partição como sendo a facilidade mais próxima dentre todas as p facilidades abertas). Tendo definido as P_k partições, recomputamos o elemento central de cada partição resolvendo o PMNC com $p = 1$ para cada uma destas partições. Após a resolução dos p subproblemas, redefinimos as partições devido ao fato de um cliente $i \in P_k$ poder ser mais próximo de uma facilidade j que define uma partição que não seja P_k. Repetimos o processo de redefinição das p partições e resolução do PMNC com $p = 1$ para cada partição até que nenhuma partição seja redefinida após a resolução dos p subproblemas. Neste momento, o método encontra-se em um ótimo local e não há nenhuma garantia que este ótimo local seja um ótimo global, característica esta inerente aos métodos de busca local.

Interchange

O segundo método de busca local encontrado na literatura é chamado de *Interchange* (ver (Teitz e Bart, 1968), (Whitaker, 1983) e (Resende e Werneck, 2004) é descrito mais detalhadamente na Seção 4.3. Porém, iremos dar uma visão geral do mesmo nesta seção. Este método é baseado na troca de papel entre um cliente e uma facilidade foi proposto por Teitz e Bart 1968 (Teitz e Bart, 1968), aprimorado por Whitaker 1983 (Whitaker, 1983) e, posteriormente, aprimorado novamente, porém por Resende e Werneck 2004 (Resende e Werneck, 2004). O procedimento de busca local básico (Teitz e Bart, 1968) analisa todas as possíveis trocas de papéis para uma determinada solução. Se existe algum melhoramento computado nestas trocas, então é feito a troca para o melhor melhoramento e o procedimento é repetido enquanto melhoramentos são descobertos. A fórmula do benefício ($profit(f_i, f_r)$) de trocar f_i por f_r é

$$\begin{aligned} profit(f_i, f_r) &= gain - netloss & (3\text{-}1) \\ gain &= \sum_{u:\phi_1(u) \neq f_r} \max\{0, [d_1(u) - d(u, f_i)]\} & (3\text{-}2) \\ netloss &= \sum_{u:\phi_1(u) = f_r} [\min\{d_2(u), d(u, f_i)\} - d_1(u)] & (3\text{-}3) \end{aligned}$$

onde u, f_i e f_r são índices para um usuário, uma facilidade candidata a inserção e uma facilidade candidata a remoção, respectivamente; $d(u, f_i)$ é o custo de servir o cliente u com a facilidade f_i; $\phi_1(u)$ é a facilidade mais próxima de u; $\phi_2(u)$ é a segunda facilidade mais próxima de u; $d(u, \phi_1(u)) = d_1(u)$ e $d(u, \phi_2(u)) = d_2(u)$. A parcela *gain* diz respeito a cada usuário u cuja facilidade mais próxima $\phi_1(u)$ não é a que será removida f_r. Neste caso, u trocará para f_i somente se esta facilidade for mais próxima. Se u é atualmente servido por f_r (segunda parcela da soma), então u deverá ser atribuído ou para $\phi_2(u)$ ou para f_i, aquele que for mais próximo. O procedimento proposto por Teitz e Bart 1968 (Teitz e Bart, 1968), possui complexidade $O(pmn)$ por iteração.

Anos depois, em (Whitaker, 1983) é descrito uma implementação eficiente do procedimento de busca local descrito anteriormente. O aspecto diferencial desta implementação é a capacidade de procurar em $O(n)$ o melhor candidato para remoção, dado um certo candidato para inserção. Com isto, ele desenvolveu um procedimento com complexidade $O(mn)$ por iteração.

Recentemente, uma nova eficiente implementação foi sugerida em (Resende e Werneck, 2004). Esta implementação executa as mesmas operações básicas daquelas descritas em (Whitaker, 1983), diferindo na ordem na qual estas operações são executadas e na utilização de estruturas auxiliares para armazenar os resultados parciais para obter acelerações evitando computar novamente informações que não alteram de uma iteração para outra do procedimento. A complexidade de pior caso deste abordagem é $O(mn)$ por iteração, similar a anteriormente. Porém, ela pode ser significativamente mais rápida na prática. A implementação utilizada nesta dissertação é aquela descrita em (Resende e Werneck, 2004) e a Seção 4.3.1 descreve detalhadamente este procedimento.

3.2.3
Metaeurísticas

As metaeurísticas são procedimentos que consistem da aplicação, em cada passo, de uma heurística subordinada. Diferenciam-se das heurísticas

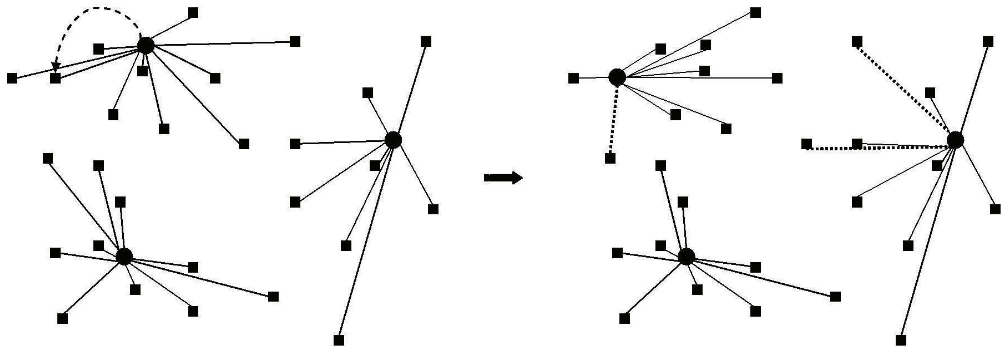

Figura 3.3: Exemplo de um movimento do tipo *Interchange*

convencionais por incluírem diversas estratégias para escapar de ótimos locais. As metaeurísticas são divididas em dois grupos, decorrente da forma como estes métodos exploram o espaço de soluções. São eles: busca local e busca populacional. Nas metaeurísticas baseadas em busca local, a exploração do espaço de soluções é feita por meio de uma seqüência de movimentos. As metaeurísticas baseadas em busca populacional, por sua vez, consistem em manter um conjunto de soluções e combiná-las de forma a tentar produzir soluções que conjuguem bons atributos das soluções existentes. Dentre as metaeurísticas baseadas em busca local utilizadas para resolver o PMNC estão: GRASP (Resende e Werneck, 2002a), Busca Tabu (Rolland et al., 1997), *Simulated Annealing* (Chiyoshi e Galvão, 2000), VNS (Hansen e Mladenovic, 1997) e uma metaeurística híbrida (Resende e Werneck, 2002b). Em (Alba e Domínguez, 2006) e (Hansen e Mladenovic, 2008) encontram-se metaeurísticas baseadas em busca populacional. A heurística de refinamento descrita na Seção 3.2.2 que se baseia na troca de papel entre um cliente e uma facilidade é uma das mais utilizadas como sub-rotina em metaeurísticas baseadas em busca local para o tratamento o PMNC. Para maiores informações sobre métodos metaeurísticos consultar (Reeves, 1993).

3.3 Algoritmos Duais

Existem dois grupos de algoritmos duais. O primeiro grupo é composto por algoritmos que trabalham com a relaxação linear (RL) de um programa linear inteiro (PLI) e o segundo grupo é composto por algoritmos que relaxam restrições do PLI na função objetivo deste mesmo PLI. Ambas as abordagens visam calcular limites inferiores para o PLI. A RL de um PLI é obtida através da relaxação das restrição de integralidade do PLI para uma ou mais variáveis. Para o PMNC, a RL é obtida substituindo as restrições 2-5 pelas restrições 2-6 e 2-7. O primeiro grupo de algoritmos duais resolvem a RL exatamente usando um algoritmo padrão (*simplex* ou *método de ponto interior*) ou heuristicamente (*Dual Ascent*). O segundo grupo utiliza técnicas de *Relaxação Lagrangeana*.

Os métodos do tipo *Dual Ascent* e *Relaxação Lagrangeana* são as abordagens duais mais referenciadas na literatura para o tratamento do PMNC. Para mensurarmos a qualidade das soluções fornecidas por estes métodos, podemos medir o quão próximo os limites fornecidos por estes métodos estão do ótimo do PLI. Porém, uma medida mais justa é comparar estes limites com as soluções ótimas fornecidas pela RL, pois estes ótimos são as melhores soluções possíveis que podemos conseguir com os métodos *Dual Ascent* e *Relaxação Lagrangeana*.

O método *Dual Ascent* é um abordagem que consiste em construir uma heurística para gerar soluções duais viáveis para a formulação dual obtida a partir da RL da PLI do PMNC. Em (Erlenkotter, 1978) é descrito o desenvolvimento de um método baseado nesta técnica para o LFNC. Os trabalhos descritos em (Galvão, 1980) e (Captivo, 1991) basearam-se no trabalho descrito em (Erlenkotter, 1978) para desenvolver um algoritmo dual de forma similar, mas especializado para resolver o PMNC. Ambos os métodos permitem obter uma solução primal para o problema a partir da solução dual de forma direta. O método apresentado em (Captivo, 1991) será a base para o estudo dos algoritmos duais descritos nesta dissertação. A escolha deste método se deve ao fato dos resultados apresentados em (Captivo, 1991) serem melhores do que os resultados apresentados em (Galvão, 1980) para um mesmo conjunto de instâncias que foram testadas em ambos os trabalhos.

Métodos do tipo *Relaxação Lagrangeana* são abordagens que consistem em anexar multiplicadores lagrangeanos para algum conjunto de restrições da PLI e relaxar estas restrições na função objetivo da PLI. Em seguida, é resolvido exatamente a nova PLI resultante destas relaxações. Em (Hanjoul e Peeters, 1985) são comparados dois métodos que utilizam diferentes relaxações lagrangeanas. As restrições 2-2 são relaxadas no primeiro método, que utiliza com base as idéias propostas em (Erlenkotter, 1978) para resolver esta relaxação. O segundo método segue basicamente as idéias descritas em (Cornuejols et al., 1977) e (Narula et al., 1977), relaxando as restrições 2-4.

3.4 Algoritmos Exatos

Alguns métodos exatos para o PMNC foram sugeridos em (Beasley, 1985), (Hanjoul e Peeters, 1985), (Avella et al., 2007), (Senne et al., 2005), entre outros. Estes trabalhos exploram tanto características do poliedro formado pelos vetores de incidência das soluções viáveis do PMNC (Avella et al., 2007), como relaxação Lagrangeana (Beasley, 1985) ou técnicas de geração de colunas (Senne et al., 2005). Nesta dissertação um algoritmo que explora os algoritmos duais aqui propostos em sua árvores de enumeração.