

### 3 Metaheurísticas e Resolvedores MIP

A combinação entre metaheurísticas e resolvedores MIP é uma estratégia relativamente recente, tendo seus primeiros estudos realizados ao longo da última década. O *survey* de Puchinger e Raidl [42] reúne e enumera diversos trabalhos presentes na literatura que seguem essa linha. Além desse, Fernandes e Lourenço [18] também realizam um levantamento similar, propondo uma classificação para abordagens que combinam metaheurísticas baseadas em busca local com algoritmos exatos, e efetuando uma listagem dos trabalhos existentes por tipo de problema tratado.

É possível classificar as abordagens que combinam metaheurísticas e resolvedores MIP em duas vertentes: *Combinações Colaborativas* e *Combinações Integrativas*. A primeira envolve algum tipo de troca de informação entre os algoritmos, de forma colaborativa, mas sem que um faça parte ou esteja dentro do outro. Podem, assim, serem executados de forma sequencial, intercalada ou paralela. Tipicamente, um algoritmo acaba funcionando como um pré ou pós-processamento do outro.

Já a segunda vertente contempla o caso em que um dos algoritmos consiste em um componente subordinado ao outro, podendo ser uma metaheurística incorporada dentro de um algoritmo exato, ou o contrário. A Figura 3.1 resume graficamente a classificação descrita.

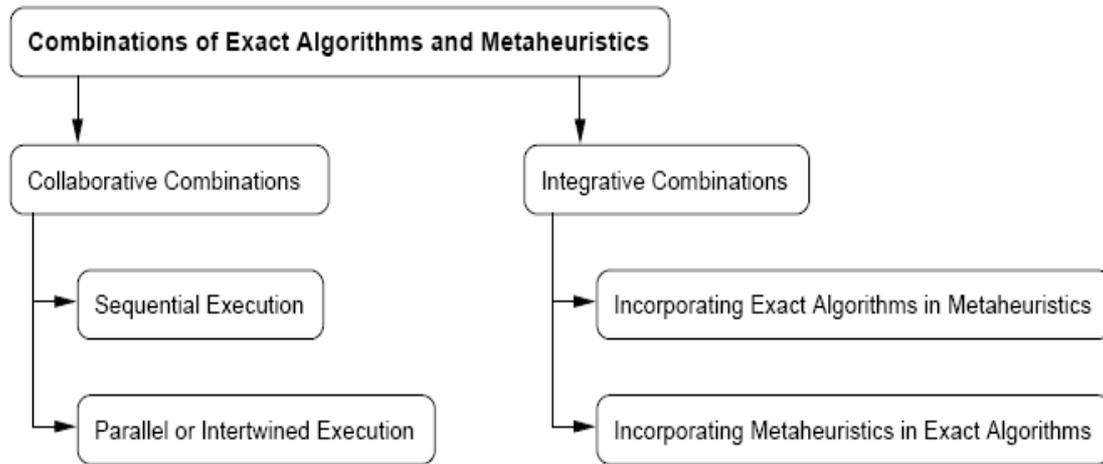


Figura 3.1: Combinações entre metaheurísticas e algoritmos exatos (Retirado de Puchinger e Raidl [42]).

### 3.1

#### *Local Branching*

O trabalho de Fischetti e Lodi [19] é considerado um exemplo de aplicação das ideias trazidas por metaheurísticas em algoritmos exatos, isto é, reside na introdução de conceitos como busca local, vizinhança, combinações de soluções e outros, dentro da estrutura do *Branch-and-Bound* [42].

Tipicamente, resolvedores MIP são desenvolvidos de forma a retornar a solução ótima do problema em um tempo de computação aceitável. No entanto, geralmente o que se deseja é somente obter uma solução razoável o mais rápido possível durante a resolução do problema. Essa motivação levou ao desenvolvimento do conceito de *Local Branching*, em que a ideia reside na resolução de um problema menor, e portanto computacionalmente mais fácil, do que o problema original, objetivando chegar a uma solução viável de boa qualidade em um curto espaço de tempo.

A fim de explicar a estrutura do *Local Branching*, seja considerado o seguinte MIP genérico:

$$(P) = \begin{cases} \min & c'x \\ & Ax \geq b \\ & x_j \in \{0, 1\} \quad \forall j \in \mathcal{B} \\ & x_j \in \mathbb{Z}^+ \quad \forall j \in \mathcal{G} \\ & x_j \geq 0 \quad \forall j \in \mathcal{C} \end{cases}$$

O conjunto  $\mathcal{B}$  contém as variáveis binárias da formulação, ao passo que o conjunto  $\mathcal{G}$  contém as variáveis inteiras gerais e  $\mathcal{C}$ , aquelas que são contínuas. Há, portanto, uma partição  $(\mathcal{B}, \mathcal{G}, \mathcal{C})$  no conjunto de variáveis da formulação.

Dada uma solução viável  $\bar{x}$  para  $(P)$ , denominada de solução-base, define-se o conjunto  $N_1(\bar{x})$ , correspondente ao suporte binário de  $\bar{x}$ , da seguinte forma:

$$N_1(\bar{x}) = \{j \in \mathcal{B} \mid \bar{x}_j = 1\} \quad (3-1)$$

isto é,  $N_1(\bar{x})$  contém os índices das variáveis binárias assumindo o valor 1. Assim, para um dado  $t$  positivo, define-se a vizinhança  $t$ -OPT  $\mathcal{N}(\bar{x}, t)$  de  $\bar{x}$ , como o conjunto de soluções viáveis de  $(P)$  satisfazendo a restrição de *local branching*:

$$\Delta(x, \bar{x}) = \sum_{j \in N_1(\bar{x})} (1 - x_j) + \sum_{j \in (\mathcal{B} \setminus N_1(\bar{x}))} x_j \leq 2t \quad (3-2)$$

Na restrição (3-2), o lado esquerdo conta o número de variáveis binárias que tem seu valor alterado, em relação a  $\bar{x}$ , de 1 para 0 e de 0 para 1, respectivamente. Dessa forma, é tido que esse valor pode corresponder a, no máximo,  $2t$ .

Quando a cardinalidade de  $N_1(\bar{x})$  for constante para qualquer solução viável do problema, a restrição pode assumir a sua chamada forma assimétrica:

$$\Delta(x, \bar{x}) = \sum_{j \in N_1(\bar{x})} (1 - x_j) \leq t \quad (3-3)$$

que permite alterar até  $t$  variáveis de 1 para 0. Como a cardinalidade de  $N_1(\bar{x})$  é constante, isso implica na mudança do valor de outras  $t$  variáveis de 0 para 1.

Supondo-se, por exemplo, que  $N_1(\bar{x})$  tenha um valor constante  $n$ , através de uma pequena manipulação algébrica chega-se à seguinte forma reduzida:

$$\Delta(x, \bar{x}) = \sum_{j \in N_1(\bar{x})} x_j \geq n - t \quad (3-4)$$

A definição de  $t$ -OPT dada acima corresponde, de fato, à definição clássica de vizinhança  $t$ -OPT utilizada em heurísticas baseadas em busca local [1]. Por exemplo, para o TSP, a restrição define que a solução  $x$  seja obtida trocando-se até  $t$  arestas da solução de referência  $\bar{x}$ . Logo, esse mesmo significado é tido na aplicação da restrição ao CVRP.

Na Figura 3.2, representa-se graficamente a vizinhança  $\mathcal{N}(\bar{x}, t)$  obtida a partir da aplicação do corte de *local branching*, também denominada de *vizinhança de bola*. Nessa, a solução-base  $\bar{x}$  corresponde ao centro da circunferência,  $p$  a um ponto qualquer nessa e  $t = \overline{\bar{x}p}$ , ao raio. Pode-se ter, dessa maneira, uma ideia do efeito do corte no espaço de soluções do problema, em que uma solução pertencente à vizinhança está contida na área definida pelo círculo.

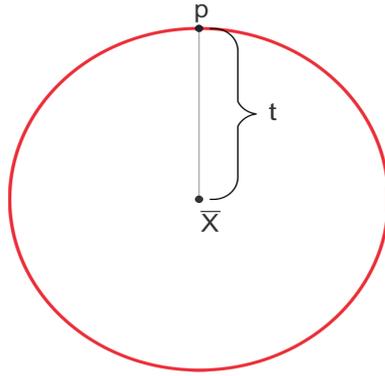


Figura 3.2: Ilustração da vizinhança de bola ( $t$ -OPT).

Fischetti e Lodi [19] ainda propõem o uso da restrição como uma estratégia de particionamento (*branching*) do espaço de soluções. Desse modo, dada uma solução viável  $\bar{x}$ , em cada nó da árvore, é feito um particionamento através das seguintes restrições: (i)  $\Delta(x, \bar{x}) \leq t$ , gerando o ramo esquerdo; e (ii) o seu complementar,  $\Delta(x, \bar{x}) \geq t + 1$ , gerando o ramo direito.

O valor escolhido para  $t$  deve ser tal que o subproblema a ser resolvido no ramo esquerdo seja mais fácil do que o associado a seu nó pai. Assim, a ideia é que a vizinhança  $\mathcal{N}(\bar{x}, t)$  seja suficientemente pequena para ser otimizada em um curto tempo de computação, mas ainda grande o suficiente para conter soluções melhores do que a solução-base  $\bar{x}$ .

Normalmente, valores de  $t$  no intervalo [10, 20] mostram-se efetivos para cumprir esse objetivo na maioria dos casos [19]. Ainda assim, esse valor pode ser aumentado ou diminuído durante a própria execução, de uma forma adaptativa. Na estrutura do algoritmo, os autores adotam a variação da vizinhança de acordo com o desempenho da busca, objetivando inserir um componente de diversificação e/ou diminuir o tempo de processamento gasto na análise da vizinhança.

### 3.2

#### Vizinhança Elipsoidal

O trabalho de Pigatti *et al.* [40] apresenta uma implementação de um *Branch-and-Cut-and-Price* para o GAP. Nesse, é feita uma estabilização na geração de colunas, a fim de contornar os problemas de convergência que podem ser trazidos por essa técnica. Todavia, a principal contribuição desse trabalho reside na proposta de uma nova vizinhança, estendendo o conceito de vizinhança de bola (ver seção 3.1).

Desse modo, em vez de centrar a busca em uma única solução-base  $\bar{x}$ , selecionam-se duas soluções  $\bar{x}_1$  e  $\bar{x}_2$ . Essas devem ser de boa qualidade e bem

diferentes uma da outra. Sejam, ainda,  $N_1(\bar{x}_1)$  e  $N_2(\bar{x}_2)$  como definidos na seção 3.1. Introduz-se, portanto, o *corte elipsoidal*:

$$\sum_{j \in N_1(\bar{x}_1) \cap N_1(\bar{x}_2)} 2x_j + \sum_{j \in (N_1(\bar{x}_1) \cup N_1(\bar{x}_2)) \setminus (N_1(\bar{x}_1) \cap N_1(\bar{x}_2))} x_j \geq \alpha(\bar{x}_1, \bar{x}_2) - t \quad (3-5)$$

em que  $\alpha(\bar{x}_1, \bar{x}_2)$  corresponde ao valor obtido fazendo-se  $x$  igual a  $\bar{x}_1$  ou  $\bar{x}_2$  no lado esquerdo. Em contrapartida,  $t$  corresponde à quantidade de folga desejada na vizinhança, especificando a distância das soluções em relação a  $\bar{x}_1$  e  $\bar{x}_2$ .

Ao se analisar a restrição, verifica-se que uma maior ponderação é dada às variáveis contidas na interseção, ou seja, presentes tanto em  $\bar{x}_1$  quanto em  $\bar{x}_2$ . A vizinhança definida dessa forma é denominada de *vizinhança elipsoidal*, em referência ao conceito de lugar geométrico elipse, que corresponde ao conjunto de pontos tal que a soma das distâncias aos dois focos é constante.

A Figura 3.3 ilustra graficamente a vizinhança, em que  $\bar{x}_1$  e  $\bar{x}_2$  correspondem aos dois focos da elipse. Assim, pode-se ter uma perspectiva da abrangência do corte no espaço de busca.

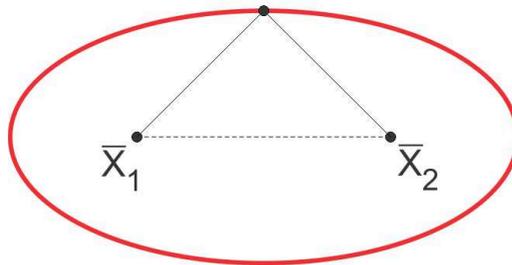


Figura 3.3: Ilustração da vizinhança elipsoidal.

Essa vizinhança possui, ainda, similaridade com os conceitos metaheurísticos de combinação entre soluções *Path-Relinking* e *Crossover*. O primeiro é usado quando se deseja criar um conjunto de soluções, vizinhas umas das outras, conectando duas soluções-base, de modo a estabelecer um caminho entre essas duas [16]. Por outro lado, um *crossover*, normalmente utilizado em abordagens seguindo o paradigma evolutivo, gera uma ou mais soluções descendentes (cromossomos) a partir da combinação de dois pais. Além desses, um outro conceito análogo é o de memória adaptativa proposto por Rochat e Taillard [47] e descrito na subseção 2.2.2.

Fazendo uso dessa vizinhança sobre o BCP implementado, Pigatti *et al.* [40] conseguiram excelentes resultados, fechando três instâncias ainda em aberto para o GAP.

Amaral [1] propõe uma extensão da vizinhança elipsoidal para o caso de três ou mais soluções, também aplicando os algoritmos propostos para

o GAP. Utilizou, portanto, a vizinhança de duas formas: (i) como um pós-processamento nas soluções fornecidas pelo resolvedor MIP usado; e (ii) mantendo um conjunto de soluções de alta qualidade e aplicando a vizinhança sobre essas (adotando a ideia do algoritmo de *Path-Relinking*).