

3 Algoritmo de Busca Tabu

3.1 Introdução

A forma básica do algoritmo de Busca Tabu está fundamentada nas idéias propostas em [Glover & Laguna, 1997] e é baseado em procedimentos heurísticos que permitem explorar o espaço de busca e encontrar novas soluções além daquelas encontradas em uma busca local. Um algoritmo é heurístico quando utiliza a intuição a respeito do problema e de sua estrutura para resolvê-lo de forma rápida. Uma busca local é caracterizada pela operação chamada movimento, que consiste na transição entre duas configurações distintas. No sistema de distribuição de energia elétrica, essa operação é caracterizada pela mudança na topologia da rede.

3.2 Histórico

Os trabalhos nas áreas de inteligência artificial e pesquisa operacional iniciaram com o desenvolvimento de métodos para solucionar problemas de competição (*challenging problems*) com o artigo [Simon & Newell, 1958]. Em pouco tempo as áreas se separaram e a Pesquisa Operacional concentrou-se em resultados dos problemas matemáticos relativos à convergência das soluções, enquanto que a Inteligência Artificial deu mais atenção à análise simbólica e qualitativa.

Nessas áreas desenvolveram-se estratégias que geram regras e reforçam o aprendizado, criam novas regras e permitem que a busca por novas soluções explore além do mínimo local. Além disso, restrições que impedem a geração de soluções não-aprimorantes para a função objetivo, através do registro de algumas características dos atributos que levam a essas soluções. Estruturas de memória permitem que as características desses atributos sejam armazenadas, de forma a controlar o número de soluções geradas. Procedimentos que combinam características semelhantes em atributos de boas soluções podem ser utilizados na geração de outras novas. Esses fundamentos na metade dos anos 80 tornaram-se os subsídios básicos da criação do método de Busca Tabu.

Busca Tabu também é utilizado na otimização do problema de reconfiguração de sistemas de distribuição de energia elétrica. [Mori & Ogita, 2002] utilizaram o método de Busca Tabu Paralelo, que consiste decompor o espaço de busca em subdivisões de modo a reduzir o esforço computacional, para redução de perdas técnicas. [Nara et al, 2002] abordaram o mesmo problema levando em consideração a presença de geração distribuída. [Jeon & Chin, 2006] utilizaram um método híbrido com Busca Tabu e *Simulated Annealing* e [Zhang et al, 2006] com Busca Tabu e Algoritmos Genéticos no problema de redução de perdas técnicas. [Guimarães, 2005] utilizou o algoritmo tendo como objetivo a maximização da margem de segurança com relação à estabilidade de tensão.

3.3 Definição

O algoritmo inicia da mesma forma que em um algoritmo de busca local, progredindo iterativamente de um ponto (solução) a outro até o critério de parada estabelecido seja satisfeito. Cada solução x dentro do espaço de busca \hat{X} , i.e. $x \in \hat{X}$, tem uma vizinhança de soluções $N(x)$, tal que $N(x) \subset \hat{X}$. Uma função objetivo $f(x)$ é utilizada como mecanismo de seleção entre soluções distintas. Este mecanismo consiste na aceitação ou rejeição de uma nova solução $x' \in N(x)$, e a transição entre a solução anterior x e a nova solução x' é feita através da operação de movimento. O método consiste em usar movimentos que aprimorem o valor da função objetivo e termina quando não existirem soluções que melhorem o valor da função objetivo.

Busca Tabu é semelhante ao método de descida simples, onde o objetivo é a minimização da função $f(x)$ ou um método correspondente ascendente onde objetivo seria a maximização da função $f(x)$. A Figura 3.1 mostra o pseudo-código de um método de descida simples genérico.

- 1 – Escolher $x \in \widehat{X}$ para iniciar o processo.
- 2 – Encontrar $x' \in N(x)$ tal que $f(x') < f(x)$.
- 3 – Se nenhuma solução x' for encontrada, x é uma solução ótima local e o método para.
- 4 – Caso contrário atribua x' como nova solução inicial e volte para o passo 2.

Figura 3.1 – Pseudo-código genérico do método de descida simples

Como pode ser visto no pseudo-código da figura 3.1, o método de descida simples busca ao redor de toda a vizinhança $N(x)$ da solução x , a solução vizinha x' que apresente o menor valor da função $f(x')$. A desvantagem do método é que muitas vezes devido ao tamanho de $N(x)$, torna-se computacionalmente impraticável verificar todas as soluções vizinhas e o valor encontrado não é um ótimo global, conseqüentemente $f(x)$ não será minimizada para todo $x \in \widehat{X}$. A estratégia da Busca Tabu é de encontrar soluções que aprimoram o valor da função objetivo, evitando a busca em toda a vizinhança.

De forma geral, os algoritmos de Busca Tabu resolvem problemas do tipo:

$$\begin{aligned} \min f(x) \\ \text{s.a } x \in \widehat{X} \end{aligned} \tag{3.1}$$

onde:

x - solução

$f(x)$ - Função Objetivo

\widehat{X} - Espaço de Busca

Usualmente os sistemas de distribuição operam na forma radial por serem mais econômicos e mais simples de serem projetados [Gönen, T., 1986], ou seja, existe um único caminho de transmissão de fluxos de potência ativa e reativa entre a fonte (subestação) e cada consumidor. Com isso, $x \in \widehat{X}$ inclui a restrição de radialidade na resolução do problema de reconfiguração em sistemas de distribuição. A restrição de radialidade é um problema de difícil representação matemática, e é tratado no algoritmo

de Busca Tabu de maneira eficiente através de uma codificação adequada das configurações.

O Algoritmo de Busca Tabu é diferente de um algoritmo de busca local em dois aspectos fundamentais:

1. Na transição entre a configuração corrente e a melhor configuração vizinha ou à menos pior, é permitida a degradação do valor da função objetivo.
2. O conjunto de vizinhos de x não se caracteriza de maneira estática. Assim, o algoritmo define uma nova vizinhança, $N^*(x)$ que varia dinamicamente em estrutura e tamanho durante o processo de otimização. Esta estratégia permite a Busca Tabu realizar uma busca eficiente e inteligente. A figura 3.2 mostra uma relação entre $N(x)$ e $N^*(x)$. Os elementos de $N^*(x)$ são determinados de várias formas, algumas apresentadas a seguir:

- Usando uma lista tabu que armazena atributos de configurações consideradas tabu (proibidas). Neste caso $N^*(x) \subset N(x)$ pois alguns vizinhos definidos pela estrutura de vizinhança e cujos atributos fazem parte da lista tabu estão proibidos. Esta estratégia impede o retorno a configurações anteriormente visitadas evitando ciclagem.
- Utilizam-se estratégias para diminuir o tamanho da vizinhança ou a lista de configurações candidatas. Em alguns casos o número de configurações vizinhas $x' \in N(x)$ pode ser muito grande e avaliar a função objetivo de cada uma dessas configurações, para encontrar aquela que apresenta melhor desempenho, pode demandar um elevado esforço computacional. Estas estratégias visam diminuir este esforço necessário em cada transição.
- Usando configurações de elite e *path relinking* para caracterizar e encontrar novas configurações candidatas. Esta estratégia visa encontrar novas configurações de alta qualidade que dificilmente seriam encontradas a partir da definição de $N^*(x)$. Verifica-se que neste caso a relação $N^*(x) \subset N(x)$ não necessariamente é verdadeira.
- Redefinição do conjunto $N^*(x)$ durante o processo de otimização.

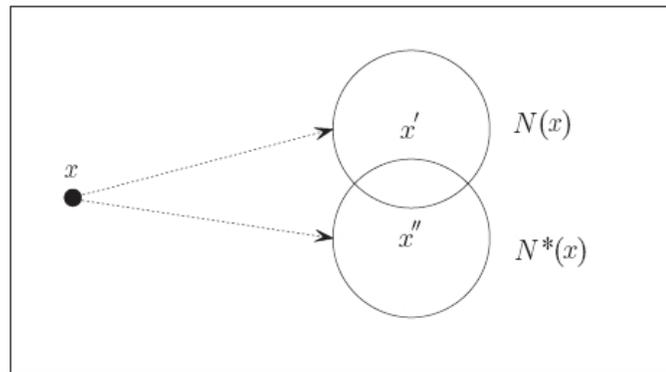


Figura 3.2 - Transição de configurações no algoritmo de Busca Tabu

A representação e codificação do problema permitem identificar claramente a forma e estrutura de uma configuração e a possibilidade de aparecimento de configurações factíveis e infactíveis, e também a forma da função objetivo. A definição e caracterização eficiente da vizinhança de uma configuração proporcionam que a busca continue além do mínimo local, permitindo movimentos que não aprimoram o valor da função objetivo e modificações na estrutura da vizinhança de soluções subseqüentes, entretanto isso depende do problema. A Figura 3.3 mostra a comparação entre a exploração do espaço de busca entre os métodos de busca local e Busca Tabu com o objetivo de encontrar o valor mínimo da função.

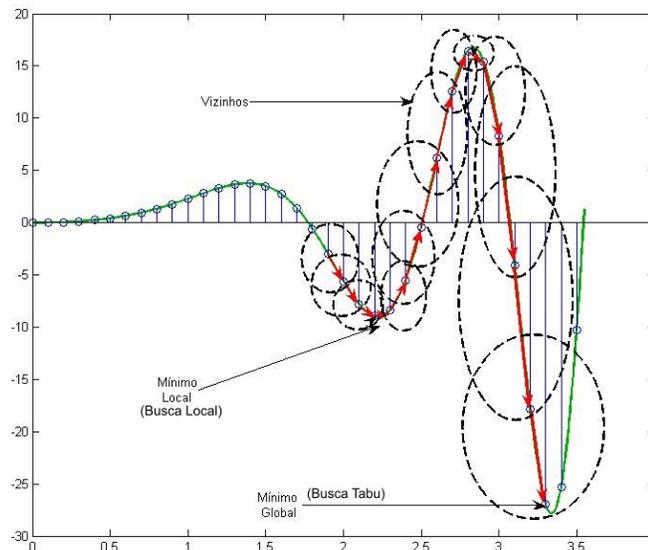


Figura 3.3 - Espaço de busca entre busca local e Busca Tabu.

3.4 Escolha da Configuração Inicial

A definição da configuração inicial adotada tem influência na direção da busca e no desempenho do algoritmo. *Simulated Annealing* e Algoritmos Genéticos são exemplos de técnicas de otimização na reconfiguração de sistemas de distribuição que utilizam processos estocásticos de busca que tendem a alcançar um mínimo (máximo) local [Mori & Ogita, 2002]. Busca Tabu não dá nenhuma ênfase à aleatoriedade, de forma que o caminho de busca pode ser direcionado com uma escolha inteligente, dependendo do tipo de problema e da experiência do usuário, da configuração inicial.

3.5 Representação e Codificação do Problema

Nesta seção será apresentada a representação do sistema de distribuição para a resolução do problema reconfiguração com o objetivo de maximizar a margem de potência. O sistema de distribuição será representado por fase e as cargas pelo modelo de potência constante. Supõe-se que haja uma chave em cada ramo da rede, exceto quando indicado o contrário. Além das chaves existentes nos ramos, existem chaves de interconexão normalmente abertas. O comando para a abertura das chaves é realizado através de um vetor de variáveis inteiras (*ch*), que indica sempre as chaves que deverão ser abertas e, portanto, todas as outras estarão fechadas.

3.5.1 Fluxo de carga em sistemas radiais de distribuição

Existem várias configurações nos sistemas de distribuição, mas a maioria delas são radiais. Circuitos radiais tem muitas vantagens em relação aos outros tipos, tais como [Short, 2004]:

- Melhor proteção contra correntes de curto-circuito
- Menor corrente de curto-circuito na maioria dos circuitos
- Melhor controle de tensão
- Melhor controle e previsão dos fluxos de potência
- Menor custo

O problema de fluxo de carga (*load flow*) consiste na obtenção do estado de operação do sistema (ângulos e magnitudes dos fasores de tensões nodais). Uma vez

obtido o estado de operação da rede, as outras grandezas, tais como, fluxos de potência ativa e reativa, correntes e perdas técnicas nos ramos podem ser facilmente determinados. O método de Newton-Raphson é robusto no estudo em sistemas de transmissão, no entanto devido à característica predominantemente radial e a problemas de convergência, este método não é o mais indicado. Os métodos baseados em varredura (*back-forward sweep*) são os mais indicados, pois não necessitam da inversão da matriz Jacobiana.

3.5.1.1 Método de soma de correntes

Seja o sistema radial de distribuição genérico de cinco barras mostrado na figura 3.4, onde a barra que está conectada o gerador representa a subestação. Inicialmente define-se o valor da tensão na barra de subestação e assume-se um perfil de tensão *flat* em todas as barras, com valor igual a $1\angle 0^0$ pu.

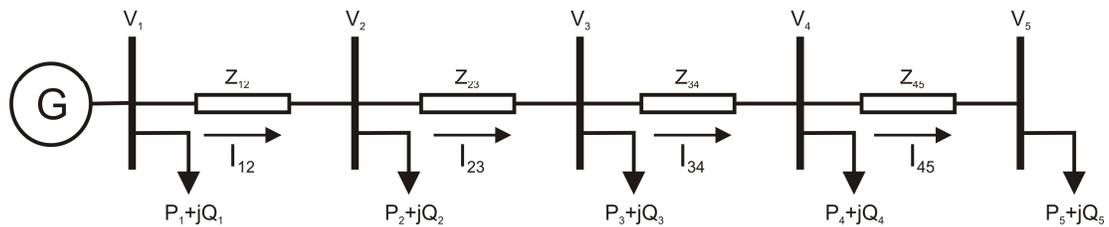


Figura 3.4 – Sistema radial de distribuição de cinco barras

A solução do fluxo de carga pelo método de varredura por soma de correntes é obtida com a aplicação de 4 passos [Shirmohammadi et al, 1988]:

3.5.1.1.1 Cálculo das injeções de corrente nodais

Na iteração k a injeção de corrente $\dot{i}_i^{(k)}$ na barra i é dada por:

$$\dot{i}_i^{(k)} = \left(\frac{\dot{S}_i}{\dot{V}_i^{(k-1)}} \right)^* - \dot{Y}_i \cdot \dot{V}_i^{(k-1)} \quad i = 1, \dots, 5, \dots, n \quad (3.2)$$

onde:

$\dot{V}_i^{(k-1)}$ é a tensão na barra i calculada na $(k-1)$ -ésima iteração.

\dot{S}_i é a injeção de potência complexa especificada na barra i .

\dot{Y}_i é o somatório dos elementos shunt conectados na barra i .

n é o número total de barras do sistema.

3.5.1.1.2 Atualização para trás (*backward sweep*)

Na iteração k saindo dos ramos conectados à barras terminais em direção aos ramos conectados à barra de subestação, calculam-se as correntes nos ramos. No exemplo de 5 barras tem-se:

$$\dot{i}_{45}^{(k)} = -\dot{i}_5^{(k)} \quad (3.3)$$

$$\dot{i}_{34}^{(k)} = -\dot{i}_4^{(k)} + \dot{i}_{45}^{(k)} \quad (3.4)$$

$$\dot{i}_{23}^{(k)} = -\dot{i}_2^{(k)} + \dot{i}_{34}^{(k)} \quad (3.5)$$

$$\dot{i}_{12}^{(k)} = -\dot{i}_1^{(k)} + \dot{i}_{23}^{(k)} \quad (3.6)$$

Em forma matricial fica:

$$\begin{bmatrix} \dot{i}_{12} \\ \dot{i}_{23} \\ \dot{i}_{34} \\ \dot{i}_{45} \end{bmatrix}^{(k)} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ & -1 & -1 & -1 \\ & & -1 & -1 \\ & & & -1 \end{bmatrix} \cdot \begin{bmatrix} \dot{i}_1 \\ \dot{i}_2 \\ \dot{i}_3 \\ \dot{i}_4 \end{bmatrix}^{(k)} \quad (3.7)$$

Para cada corrente de ramo existem 2 parcelas: A primeira igual ao negativo da injeção de corrente da barra terminal e a segunda igual ao somatório das corrente nos ramos conectados à barra terminal. A equação (3.7) é a aplicação da lei das correntes de Kirchhoff.

3.5.1.1.3 Atualização para frente (*forward sweep*)

As tensões nodais são atualizadas calculando-se as quedas de tensões a partir dos ramos conectados na barra de subestação em direção aos ramos das barras terminais. No circuito da figura 3.4, partindo-se da barra de subestação 1 em direção à barra terminal 5, temos:

$$\dot{V}_2^{(k)} = \dot{V}_1^{(k)} - \dot{Z}_{12} \cdot \dot{i}_{12}^{(k)} \quad (3.8)$$

$$\dot{V}_3^{(k)} = \dot{V}_2^{(k)} - \dot{Z}_{23} \cdot \dot{i}_{23}^{(k)} \quad (3.9)$$

$$\dot{V}_4^{(k)} = \dot{V}_3^{(k)} - \dot{Z}_{34} \cdot \dot{i}_{34}^{(k)} \quad (3.10)$$

$$\dot{V}_5^{(k)} = \dot{V}_4^{(k)} - \dot{Z}_{45} \cdot \dot{i}_{45}^{(k)} \quad (3.11)$$

Em forma matricial fica:

$$\begin{bmatrix} \dot{V}_1 \\ \dot{V}_2 \\ \dot{V}_3 \\ \dot{V}_4 \end{bmatrix}^{(k)} = \dot{V}_1 \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} \dot{Z}_{12} & & & \\ \dot{Z}_{12} & \dot{Z}_{23} & & \\ \dot{Z}_{12} & \dot{Z}_{23} & \dot{Z}_{34} & \\ \dot{Z}_{12} & \dot{Z}_{23} & \dot{Z}_{34} & \dot{Z}_{45} \end{bmatrix} \cdot \begin{bmatrix} \dot{i}_{12} \\ \dot{i}_{23} \\ \dot{i}_{34} \\ \dot{i}_{45} \end{bmatrix}^{(k)} \quad (3.12)$$

A equação (3.12) é aplicação direta da lei de Kirchhoff das tensões.

3.5.1.1.4 Critério de convergência

Na iteração k , calcula-se a injeção de potência aparente na barra i por:

$$\dot{S}_i^{(k)} = \dot{V}_i^{(k)} \cdot [\dot{i}_i^{(k)}]^* - \dot{Y}_i \cdot |V_i^{(k)}|^2 \quad (3.13)$$

Usando (3.13) obtém-se os erros de potência ativa e reativa em:

$$\Delta P_i^{(k)} = \text{Re}[S_i^{(k)} - S_i] \quad i = 1, \dots, n \quad (3.14)$$

$$\Delta Q_i^{(k)} = \text{Im}[S_i^{(k)} - S_i] \quad i = 1, \dots, n \quad (3.15)$$

A convergência é alcançada quando os erros máximos de potência ativa e reativa na k-ésima iteração são menores que a tolerância especificada, ou seja:

$$\max[\Delta P_i^{(k)}, \Delta Q_i^{(k)}] < \varepsilon \quad (3.16)$$

3.5.1.2 Método de soma de potências

As equações de fluxo de potência serão demonstradas utilizando o sistema de 5 barras da Figura 3.4. A solução do método de varredura por soma de potências é obtida admitindo-se que se conheça a tensão e os fluxos de potência ativa e reativa da barra de referência do sistema [Baran, Wu, 1989]. Aplicando-se o critério de atualizações, tem-se um sistema de equações recursivas.

3.5.1.2.1 Atualização para frente (*forward update*)

Conhecendo-se a tensão V_1 e os fluxos de potência P_{12} e Q_{12} , calculam-se os fluxos nos ramos e as tensões nas barras em direção à barra terminal. No sistema de 5 barras tem-se:

$$P_{23} = P_{12} - r_{12} \cdot \left(\frac{P_{12}^2 + Q_{12}^2}{|V_1|^2} \right) - P_1 \quad (3.17)$$

$$Q_{23} = Q_{12} - x_{12} \cdot \left(\frac{P_{12}^2 + Q_{12}^2}{|V_1|^2} \right) - Q_1 \quad (3.18)$$

$$P_{34} = P_{23} - r_{23} \cdot \left(\frac{P_{23}^2 + Q_{23}^2}{|V_2|^2} \right) - P_2 \quad (3.19)$$

$$Q_{34} = Q_{23} - x_{23} \cdot \left(\frac{P_{23}^2 + Q_{23}^2}{|V_2|^2} \right) - Q_2 \quad (3.20)$$

$$P_{45} = P_{34} - r_{34} \cdot \left(\frac{P_{34}^2 + Q_{34}^2}{|V_3|^2} \right) - P_3 \quad (3.21)$$

$$Q_{45} = Q_{34} - x_{34} \cdot \left(\frac{P_{34}^2 + Q_{34}^2}{|V_3|^2} \right) - Q_3 \quad (3.22)$$

$$V_2^2 = V_1^2 - 2 \cdot (r_{12} \cdot P_{12} + x_{12} \cdot Q_{12}) + \left[(r_{12}^2 + x_{12}^2) \cdot \left(\frac{P_{12}^2 + Q_{12}^2}{V_1^2} \right) \right] \quad (3.23)$$

$$V_3^2 = V_2^2 - 2 \cdot (r_{23} \cdot P_{23} + x_{23} \cdot Q_{23}) + \left[(r_{23}^2 + x_{23}^2) \cdot \left(\frac{P_{23}^2 + Q_{23}^2}{V_2^2} \right) \right] \quad (3.24)$$

$$V_4^2 = V_3^2 - 2 \cdot (r_{34} \cdot P_{34} + x_{34} \cdot Q_{34}) + \left[(r_{34}^2 + x_{34}^2) \cdot \left(\frac{P_{34}^2 + Q_{34}^2}{V_3^2} \right) \right] \quad (3.25)$$

$$V_5^2 = V_4^2 - 2 \cdot (r_{45} \cdot P_{45} + x_{45} \cdot Q_{45}) + \left[(r_{45}^2 + x_{45}^2) \cdot \left(\frac{P_{45}^2 + Q_{45}^2}{V_4^2} \right) \right] \quad (3.26)$$

3.5.1.2.2 Atualização para trás (*backward update*)

O processo de atualização para trás inicia-se da barra terminal em direção à barra de subestação. Assumindo que os fluxos de potência e a tensão da barra terminal são conhecidos, tem-se no sistema de 5 barras

$$P_{34} = P_{45} + r_{45} \cdot \left(\frac{P_{45}'^2 + Q_{45}'^2}{V_4^2} \right) + P_4 \quad (3.27)$$

$$Q_{34} = Q_{45} + x_{45} \cdot \left(\frac{P_{45}'^2 + Q_{45}'^2}{V_4^2} \right) + Q_4 \quad (3.28)$$

$$P_{23} = P_{34} + r_{34} \cdot \left(\frac{P_{34}'^2 + Q_{34}'^2}{V_3^2} \right) + P_3 \quad (3.29)$$

$$Q_{23} = Q_{34} + x_{34} \cdot \left(\frac{P_{34}'^2 + Q_{34}'^2}{V_3^2} \right) + Q_3 \quad (3.30)$$

$$P_{12} = P_{23} + r_{23} \cdot \left(\frac{P_{23}'^2 + Q_{23}'^2}{V_2^2} \right) + P_2 \quad (3.31)$$

$$Q_{12} = Q_{23} + x_{23} \cdot \left(\frac{P_{23}'^2 + Q_{23}'^2}{V_2^2} \right) + Q_2 \quad (3.32)$$

onde,

$$P_{45}' = P_{45} + P_4 \quad (3.33)$$

$$Q_{45}' = Q_{45} + Q_4 \quad (3.34)$$

$$P_{34}' = P_{34} + P_3 \quad (3.35)$$

$$Q'_{34} = Q_{34} + Q_3 \quad (3.36)$$

$$P'_{23} = P_{23} + P_2 \quad (3.37)$$

$$Q'_{23} = Q_{23} + Q_2 \quad (3.38)$$

3.5.1.2.3 Critério de convergência

A convergência é atingida quando o erro entre os valores de fluxos de potência e tensões estimados e calculados é menor que a tolerância estabelecida.

3.5.2 Geração das configurações vizinhas

O algoritmo de Busca Tabu explora a vizinhança da configuração corrente para encontrar a configuração que aprimore o valor da função objetivo. Portanto, o desempenho do algoritmo depende da qualidade dessas configurações. O método de seleção das configurações vizinhas utilizados nesta dissertação é baseado no algoritmo heurístico de [Goswami & Basu, 1992] e divide-se em 2 partes: A primeira avalia se a configuração vizinha encontrada é factível e na segunda calculam-se as margens de potência dos vizinhos.

A primeira parte consiste em fechar uma chave aberta e assim forma-se um laço, devido ao fato que o algoritmo foi desenvolvido para redes radiais. As chaves presentes no laço, excluindo a chave aberta que foi fechada anteriormente, serão abertas sequencialmente, partindo-se de um dos lados, verificando a desconexão da rede resultante (barras ilhadas, ramos sem alimentação e malhas) através da rotina “caminhos” em MATLAB [Yamamoto, 2004]. Na Figura 3.4 pode-se observar que um laço formado pelo fechamento da chave i-j. Para manter a topologia da rede radial é necessária a abertura de uma das chaves pertencente ao laço.

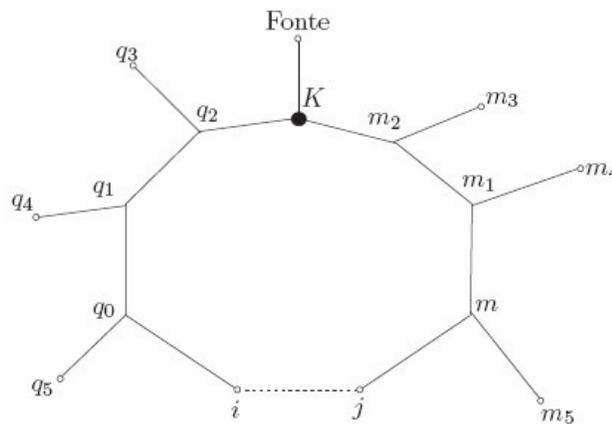


Figura 3.5 – Laço formado ao se fechar a chave i-j

Na segunda parte calculam-se as margens de potência das configurações factíveis, através de (2.68) e (2.70) em ordem decrescente da margem da barra crítica por configuração, armazenando somente as que apresentam margens da barra crítica maior que a margem da barra crítica da configuração corrente. Com isso evita-se que a vizinhança apresente configurações que não levem à maximização da margem de carregamento do sistema, reduz a dimensão do conjunto de configurações vizinhas em sistemas de distribuição maiores e evita-se a escolha aleatória das opções de chaveamento, levando à escolha de vizinhos em direção a soluções de boa qualidade.

3.5.3 Estruturas de memória utilizadas em Busca Tabu

Estruturas de memória permitem armazenar informações sobre a mudança de atributos do problema durante a transição de soluções. Informações de qualidade permitem avaliar o mérito das configurações visitadas durante a busca e identificar elementos que são comuns em soluções de boa qualidade ou em direções que levam a tais soluções. Isto se traduz em aprendizado por incentivo, em que incentivos a ações que direcionam às boas soluções são reforçados e penalidades são aplicadas àquelas que levam a soluções de qualidade inferior. Existem dois tipos de memória utilizados na Busca Tabu, que são descritas a seguir.

3.5.3.1 Memória de curto prazo

A memória de curto prazo armazena registros dos atributos que foram alterados durante os mais recentes movimentos feitos pelo algoritmo. As informações contidas nessa memória permitem rotular como proibidos (tabu), os atributos selecionados de configurações que foram visitadas em um passado recente. Isso evita que o algoritmo volte a elas, dirigindo a busca para novas regiões. O armazenamento dos atributos, porém traz um problema adicional, porque pode impedir o algoritmo de visitar configurações de boa qualidade se estas configurações tiverem um atributo proibido. Isso é contornado criando-se um mecanismo que permite a eliminação do rótulo “tabu” dos elementos de uma configuração, desde que um critério pré-estabelecido seja satisfeito. Este mecanismo chama-se Critério de Aspiração, que pode ser definido e aplicado de maneiras diferentes, dependendo do objetivo do algoritmo, sendo muito utilizado aquele em que é permitido um movimento “tabu” desde que o valor da função objetivo seja melhor do que o encontrado até aquele momento.

Em resumo, o algoritmo de Busca Tabu com memória de curto prazo apresenta as seguintes características:

- É um processo de K transições entre configurações factíveis e/ou infactíveis em um problema de otimização. O valor de K pode ser predefinido ou adaptativamente determinado.
- Dentre os vizinhos da configuração corrente, deve ser localizado aquele que menos deteriore o valor da função objetivo.
- Uma configuração é válida se não contiver atributos proibidos, ou caso tiver, que atenda ao critério de aspiração estabelecido.
- A lista tabu é atualizada a cada iteração do algoritmo.

3.5.3.2 Lista Tabu com memória de curto prazo

Como dito anteriormente, a memória de curto prazo guarda as informações sobre atributos que sofreram alteração no passado recente. Podem-se armazenar esses atributos em uma matriz ou em um vetor, sendo que este tipo de armazenamento é vantajoso no problema de reconfiguração em sistemas de distribuição porque se armazenam a informação referente à chave selecionada, dado pela posição do vetor, e o

tempo em que a chave deverá permanecer na lista tabu, pelo valor da posição do vetor. A Figura 3.5 ilustra este tipo de armazenamento para o sistema radial de distribuição de [Baran, Wu, 1989] com 32 barras.

$$t| = \begin{array}{cccccccccccccccccccccccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 \\ \hline 0 & 0 & 0 & 4 & 0 & 0 & 0 & 3 & 5 & 0 & 0 & 0 & 8 & 2 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \end{array}$$

Figura 3.6 - Exemplo de armazenamento da Lista Tabu em vetor

Na Figura 3.6, por exemplo, o valor da posição 19 do vetor $t|$ é igual a 5, indicando que o estado da chave 19 não poderá ser alterado durante as próximas 5 iterações.

3.5.3.3 Memória de longo prazo

A memória de longo prazo é um aperfeiçoamento adicionado ao algoritmo básico, e permite encontrar novas configurações de alta qualidade, utilizando informações armazenadas durante o processo de memória de curto prazo. Os componentes principais da memória de longo prazo são: Memória baseada em frequência, intensificação e diversificação.

A memória baseada em frequência fornece informações a respeito de movimentos preferenciais, levando em conta a influência do movimento na qualidade da solução. O armazenamento de atributos que produzam soluções de alta qualidade, baixa qualidade ou ambas em uma determinada trajetória da busca é chamado de frequência de residência. A informação obtida por esse processo pode ser utilizada pelo algoritmo para a pesquisa de regiões mais atrativas no processo de intensificação. O armazenamento por frequência de transição indica o número de vezes que um determinado atributo foi adicionado ou modificado durante o processo de busca. Esta informação pode ser utilizada para a mudança de estratégia de busca no processo de diversificação

No mecanismo de intensificação são armazenadas configurações de alta qualidade, chamadas de configurações de elite. Uma busca mais profunda ao redor dessas configurações pode levar a uma solução ótima global. Portanto intensificar a busca consiste em fazer uma exploração cuidadosa em torno das configurações elite armazenadas, tentando encontrar configurações melhores que as já encontradas e, eventualmente, a solução ótima global. A Figura 3.7 ilustra o processo de intensificação que inicia na região A e nas regiões de configurações de elite (1, 2, 3 e 4) são feitas

buscas locais de modo a aprimorar o valor da função objetivo, encontrando ótimos locais. O processo termina na região B onde é encontrado o valor ótimo global.

A Diversificação consiste em levar o algoritmo a regiões do espaço de busca ainda não exploradas, eliminando atributos anteriormente visitados em busca de novas configurações. Outra maneira de realizar a diversificação é aumentar o tempo de permanência de um atributo na lista tabu, forçando-se a pesquisa de novas regiões. A Figura 3.7 também ilustra o processo de diversificação que tem início na região C e após a transição para a região D a busca toma direções divergentes. São 3 processos de diversificação diferentes: (1) transição D-E termina em uma configuração ótima local, (2) movimento D-F termina em outra configuração ótima local e (3) D-B encontra a configuração ótima global.

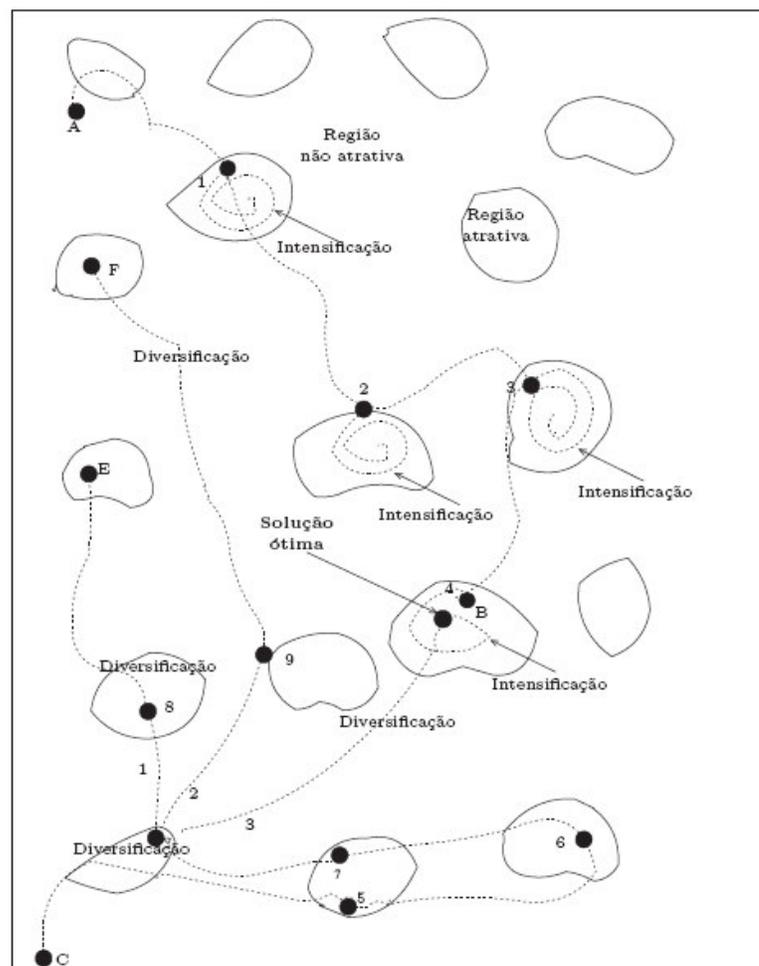


Figura 3.7 - Intensificação e diversificação em Busca Tabu

3.5.4 Estratégias adicionais utilizadas em Busca Tabu

3.5.4.1 Configurações de elite

Trata-se de um conjunto reduzido das melhores configurações que foram visitadas durante o processo. Uma configuração de elite pode ser utilizada em um processo de diversificação em busca de novas regiões atrativas. Uma configuração é considerada de elite e substitui a pior configuração encontrada no processo de Busca Tabu se satisfizer os seguintes critérios:

1. Possui o valor da função objetivo de melhor qualidade em relação aos valores das configurações armazenadas
2. Possui algum atributo diferente em relação aos das configurações armazenadas.

3.5.4.2 Path relinking

Consiste em usar duas ou mais configurações de elite, chamadas configurações de referência, para gerar uma nova configuração. A Figura 3.8 mostra a configuração base A e as configurações de referência 5, 9 e 12. A partir delas são geradas novas configurações em regiões factíveis, encontrando a solução ótima global na configuração 10.

Outra forma de implementar *path relinking*, é chamada de vizinhança construtiva, que consiste em encontrar uma única configuração e iniciar um processo de intensificação ou diversificação a partir dela. A configuração é formada por atributos de alta qualidade presentes na configuração base e nas configurações de referência. Quando o número de atributos da configuração base é grande, pode-se iniciar um processo de intensificação, caso contrário utiliza-se o processo de diversificação.

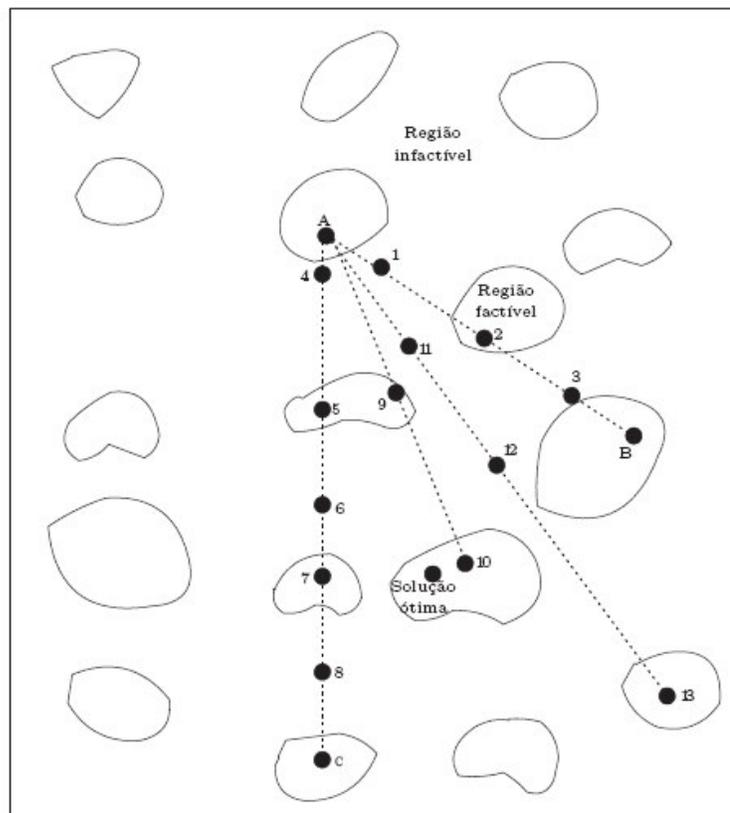


Figura 3.8 - Path relinking em Busca Tabu

3.5.4.3 Oscilação estratégica

A oscilação estratégica está baseada em três diferentes técnicas que são usadas alternadamente:

- Busca em regiões infactíveis com o objetivo de alcançar a fronteira e entrar na região factível.
- Busca na região factível para encontrar uma solução ótima local.
- Uma estratégia para sair da região factível e entrar novamente na região infactível.

A oscilação estratégica é mais eficiente quando usada em problemas onde o tamanho da região infactível é relativamente grande, como no caso do problema de reconfiguração em sistemas radiais de distribuição de grande porte.

3.5.5 Busca Tabu na maximização da margem de potência

Nesta dissertação foi implementado o algoritmo de otimização de Busca Tabu para a solução do problema de reconfiguração de sistemas de distribuição de energia elétrica com o objetivo de maximização da margem de potência. O ambiente de desenvolvimento escolhido foi o MATLAB 7 devido sua sintaxe simples.

No problema de reconfiguração em sistemas de distribuição existe a restrição de radialidade, que não permite uma configuração com barras e ramos ilhados e/ou malhas na rede elétrica. A rotina “caminhos” em [Yamamoto, 2004] permite avaliar se uma configuração é infactível através de um indicador binário com valor lógico 1. Caso a rede seja factível, a função “caminhos” retorna a matriz de configuração do sistema, com todos percursos possíveis desde a barra de subestação até cada barra terminal. Com isso podem-se aplicar as equações de fluxo de carga (3.2), (3.7) e (3.12).

O algoritmo de Busca Tabu pode ser descrito em linhas gerais pelo pseudo-código da Figura 3.9.

- 1 - $k=1$
- 2 - Solução inicial: M_0^C
- 3 - Configuração Inicial Factível: $ch = ch_0$
- 4 - Lista Tabu: $LT = \emptyset$
- 5 - Enquanto (Critério de Parada não for Satisfeito)
- 6 - Conjunto de configurações vizinhas factíveis: $V(ch)$
- 7 - Se $\exists ch_k \in V(ch)$ tal que $M_k^C > M_{k-1}^C$
- 8 - Nova configuração corrente: $ch = ch_k$
- 9 - $ch_{k-1} \in LT$ volta para 6.
- 10 - Caso contrário M_{k-1}^C é a melhor solução
- 11 - Atualizar Lista Tabu
- 12 - $k=k+1$
- 13 - Fim

Figura 3.9 – Pseudo-código Busca Tabu aumento da margem de potência

em que:

M_k^C é a margem de potência da barra crítica (menor margem) na configuração ch_k , dada por (2.68), em %, para ponto de operação na região normal de operação e por (2.70), em %, para ponto de operação na região anormal de operação.

Utilizou-se como critério de parada o número máximo de iterações, mas o algoritmo é flexível e pode ser escolhido outros critérios relativos às restrições operacionais em sistemas reais de distribuição. O número máximo de iterações e o tempo em que um atributo permanece proibido na lista tabu na rede de 32 barras foram respectivamente de 12 e 4 iterações, que são os mesmos adotados em [Guimarães, 2005].