



UNIVERSIDADE ESTADUAL PAULISTA

“JÚLIO DE MESQUITA FILHO”

Câmpus de Ilha Solteira–SP

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MAX ROBERT MARINHO

**Otimização do Problema de Reconfiguração de Sistemas de Distribuição de
Energia Elétrica por meio das Meta-Heurísticas Busca Tabu, GRASP e
*Path Relinking***

Ilha Solteira

2020

A decorative graphic in the bottom right corner of the page, consisting of several overlapping, semi-transparent blue shapes with a white polka-dot pattern, arranged in a geometric, crystalline structure.

MAX ROBERT MARINHO

**Otimização do Problema de Reconfiguração de Sistemas de Distribuição de
Energia Elétrica por meio das Meta-Heurísticas Busca Tabu, GRASP e
*Path Relinking***

Tese apresentada à Faculdade de Engenharia –
UNESP – Campus de Ilha Solteira, para
obtenção do título de Doutor em Engenharia
Elétrica.

Área do Conhecimento: Automação.

Rubén Augusto Romero Lázaro
Orientador

Ilha Solteira

2020

FICHA CATALOGRÁFICA
Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

M338o Marinho, Max Robert.
Otimização do problema de reconfiguração de sistemas de distribuição de energia elétrica por meio das meta-heurísticas busca tabu, GRASP e *path relinking* / Max Robert Marinho. -- Ilha Solteira: [s.n.], 2020
204 f. : il.

Tese (doutorado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2020

Orientador: Rubén Augusto Romero Lázaro
Inclui bibliografia

1. Busca tabu. 2. *Path relinking*. 3. GRASP. 4. Reconfiguração de sistemas de distribuição de energia elétrica. 5. Otimização de fluxo de potência.

Raiane da Silva Santos
Raiane da Silva Santos

CERTIFICADO DE APROVAÇÃO

TÍTULO DA TESE: Otimização do problema de reconfiguração de sistemas de distribuição de energia elétrica por meio das meta-heurísticas busca tabu, GRASP e path relinking

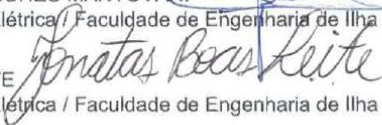
AUTOR: MAX ROBERT MARINHO

ORIENTADOR: RUBEN AUGUSTO ROMERO LAZARO

Aprovado como parte das exigências para obtenção do Título de Doutor em ENGENHARIA ELÉTRICA, área: Automação pela Comissão Examinadora:

Prof. Dr. RUBEN AUGUSTO ROMERO LAZARO, 
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira - UNESP

Prof. Dr. JOSE ROBERTO SANCHES MANTOVANI, 
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira - UNESP

Prof. Dr. JONATAS BOAS LEITE, 
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira - UNESP

Profa. Dra. FERNANDA CASEÑO TRINDADE ARIOLI, 
Departamento de Sistemas e Energia / Universidade Estadual de Campinas - UNICAMP

Prof. Dr. LUIS GUSTAVO WESZ DA SILVA, 
Campus de Itumbiara / Instituto Federal de Educação, Ciência e Tecnologia de Goiás - IFG

Ilha Solteira, 29 de abril de 2020

DEDICO

À minha esposa Carolina Garcia e minhas filhas Beatriz e Heloísa Garcia pelo amor, motivação e tempos dedicados.

Aos meus pais por todo o amparo, cuidado e carinho nesta jornada.

A todos meus familiares que torceram por mim.

AGRADECIMENTOS

Primeiramente a Deus, acima de tudo, que me fez trilhar por caminhos tortuosos, para que eu pudesse dar valor e tivesse orgulho de tudo o que até hoje fiz.

Ao meu orientador Prof. Dr. Ruben Romero Lázaro que muito de seu tempo e paciência doou para que este trabalho pudesse ter sido desenvolvido.

À minha esposa Carolina Garcia e filhas Beatriz e Heloísa pela paciência, incentivo e tempo de suas vidas que foram doados de bom grado para que eu pudesse ascender profissionalmente. Por terem me entendido quando suas atenções, amor e carinho não podiam ser correspondidos.

Aos meus pais, irmãos, cunhadas, cunhados e sogros por terem sempre me incentivado e compreendido quando meus pensamentos eram somente para o cunho científico.

A meus companheiros, amigos e parceiros de pesquisa, Prof. Me. Lucas Kriesel Sperotto e Dra. Eliane Silva de Souza, por muito terem me ajudado e orientado quando saídas não podiam mais ser vistas.

Aos meus amigos Sérgio, Ubirajara, Caio, Wesley, Carlos e Juvenal por também entenderem a minha necessidade com este trabalho científico e terem tomado para si responsabilidades as quais algumas eu deveria ter tomado.

Um agradecimento em especial ao Prof. Dr. Cláudio Kitano por ter sido o professor que primeiro acreditou em mim para o início de meu trabalho, e a todos os alunos de pós-graduação que participaram, junto a mim, desta minha jornada.

Agradeço à UNEMAT pelo meu afastamento para minha pura dedicação a esta qualificação e à UNESP por ter me aceito e acreditado em meu trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) com código de financiamento 001 e da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) através do Projeto Temático 2015/21972-6.

“Veni, vidi, vici” Júlio César

RESUMO

O problema de reconfiguração de sistemas de distribuição de energia elétrica consiste em encontrar uma configuração radial por meio da permutação do estado das chaves (abertura ou fechamento) dos ramos de um sistema elétrico. O objetivo é de se alcançar a minimização das perdas elétricas. Cada configuração radial só é considerada factível se respeitar certas restrições operacionais como o limite de tensão nas barras e os limites de correntes nos circuitos. O modelo tratado neste trabalho apresenta explosão combinatória e difícil tratabilidade por meio de métodos convencionais de otimização. O problema, computacionalmente falando, é considerado Não-Polinomial Completo (NPC), pois não possui uma resposta em tempo polinomial a partir de uma entrada definida. Neste trabalho são apresentadas três técnicas meta-heurísticas para se tratar o problema de reconfiguração de sistemas de distribuição de energia elétrica, totalmente diferentes entre uma e outra, atuando em conjunto, para somente um nível de demanda, no intuito de se encontrar a topologia ótima, com o objetivo de se minimizar as perdas elétricas ativas. Além disso, propôs-se modificar o paradigma clássico de implementação estático deste tipo de problema para o paradigma de programação dinâmica por meio de árvores com filhos variados a fim de que a estrutura de dados utilizada representasse fielmente um sistema de distribuição de energia elétrica na memória do computador. As meta-heurísticas implementadas foram a *Greedy Randomized Adaptive Search Procedure* (GRASP) como estratégia de obtenção de uma configuração inicial de boa qualidade; a Busca Tabu (BT) como procedimento de busca local; e a técnica de *Path Relinking* (PR) como mecanismo de pós-otimização. Todo o programa foi implementado em linguagem C com uso da IDE Code Blocks 17.12. Os testes foram realizados com sistemas de pequeno, médio e grande porte de 14, 33, 84, 136 e 415 barras. As técnicas comparativas envolveram resultados da literatura especializada e obtidos por meio de outros procedimentos meta-heurísticos.

Palavras-chave: Busca Tabu. *Path Relinking*. GRASP. Reconfiguração de sistemas de distribuição de energia elétrica. Otimização de fluxo de potência.

ABSTRACT

The distribution network reconfiguration problem is based on finding a radial configuration through the electrical system branches switches permutation (opening or closing). The objective is to achieve the electrical losses minimization. Each radial configuration is only considered feasible if it complies with certain operational constraints such as the voltage levels on the buses and the current limits on the circuits. The model treated in this work has a combinatorial explosion and is difficult to treat using conventional optimization methods. The problem, computationally speaking, is considered Non-Polynomial Complete (NPC), as it does not have a polynomial time response from a defined input. In this work, three meta-heuristic techniques are presented to deal with the distribution network reconfiguration, totally different from one to the other, acting together, for only one level of demand, in order to find the optimal topology, in order to minimize active electrical losses. In addition, it was proposed to modify the classic static implementation paradigm of this type of problem to the dynamic programming paradigm by means of trees with varied children so that the data structure used would faithfully represent an electrical energy distribution system in the computer memory. The implemented meta-heuristics were the Greedy Randomized Adaptive Search Procedure (GRASP) as a strategy to obtain a good quality initial configuration; Tabu Search (BT) as a local search procedure; and the Path Relinking (PR) technique as a post-optimization mechanism. The entire program was implemented in C language using Code Blocks 17.12 IDE. The tests were carried out with small, medium and large systems of 14, 33, 84, 136 and 415 buses. The comparative techniques involved results from the specialized literature and obtained through other meta-heuristic procedures.

Keywords: Tabu Search. Path Relinking. GRASP. Distribution network reconfiguration. Power flow optimization.

LISTA DE FIGURAS

Figura 1 – Construção de uma solução do problema do Caixeiro Viajante por meio do AHC.	14
Figura 2 – Processo de busca por vizinhança – SDH.	14
Figura 3 – Principais características da Busca Tabu.....	17
Figura 4 – Representação de soluções para o procedimento de PR para o problema do Caixeiro Viajante – (a) Solução inicial – (b) Solução guia.	27
Figura 5 – Sistema inicial de distribuição de 14 barras com 511 kW de perdas.	30
Figura 6 – Ilustração do grafo de uma árvore.	52
Figura 7 – Ilustração do grafo de uma árvore.	52
Figura 8 – Formas de percurso em uma árvore binária.....	54
Figura 9 – Sintaxe e representação de um registro.	56
Figura 10 – Fluxograma da proposta.....	57
Figura 11 – Algoritmo PRIM implementado.....	65
Figura 12 – Algoritmo GRASP implementado.....	67
Figura 13 – Algoritmo de busca local com a BT.....	72
Figura 14 - Algoritmo da proposta de PR.....	75
Figura 15 - Algoritmo de construção de uma solução radial para uma Árvore Dinâmica.	78
Figura 16 - Algoritmo para o procedimento de cálculo das correntes para o algoritmo de fluxo de potência.....	81
Figura 17 - Algoritmo para o procedimento de cálculo das tensões para o procedimento de fluxo de potência.....	82
Figura 18 - Algoritmo para o procedimento de cálculo das perdas elétricas para o procedimento de fluxo de potência.....	83
Figura 19 - Algoritmo que agrega as funcionalidades necessárias para o cálculo do fluxo de potência de uma solução radial.....	84
Figura 20 - Representação de ciclos internos dependentes de uma solução radial.....	86
Figura 21 - Representação em árvore do sistema da Figura 20.	86
Figura 22 - Algoritmo dos procedimentos de construção de M_ciclos e vet_ciclos.	88
Figura 23 - Sistema de 14 barras.	100
Figura 24 - Soluções obtidas pelo algoritmo GRASP para o sistema de 14 barras.	101
Figura 25 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 14 barras com solução base como configuração inicial - (a) SRV - (b) CRV.....	103
Figura 26 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 14 barras com configuração inicial obtida pelo PRIM e GRASP - (a) SRV - (b) CRV.....	103
Figura 27 - Melhores soluções obtidas no procedimento de busca local pela BT - (a) Solução base como solução inicial e SRV – (b) Solução base como solução inicial e CRV – (c) PRIM e GRASP como solução inicial e SRV – (d) PRIM e GRASP como solução inicial e CRV.....	104
Figura 28 - Histórico de soluções do PR – (a) Solução base como configuração inicial e SRV – (b) Solução base como configuração inicial e CRV – (c) PRIM e GRASP como configuração inicial e SRV/CRV.....	105
Figura 29 - Sistemas de 33 barras.....	106
Figura 30 - Soluções obtidas pelo algoritmo GRASP para o sistema de 33 barras.	107

Figura 31 – Histórico de solução geradas/avaliadas pelo procedimento BT/PR para o sistema de 33 barras com solução base como configuração inicial – (a) SRV – (b) CRV.	109
Figura 32 - Histórico de solução geradas/avaliadas pelo procedimento BT/PR para o sistema de 33 barras com PRIM e GRASP como configuração inicial – (a) SRV – (b) CRV.	110
Figura 33 - Melhores soluções da BT a cada iteração, para o sistema de 33 barras – (a) Solução base como configuração inicial e SRV – (b) Solução base como configuração inicial e CRV – (c) PRIM como solução inicial e SRV – (d) GRASP como solução inicial e CRV.	111
Figura 34 - Histórico de soluções do PR - (a) Solução base como configuração inicial e SRV - (b) Solução base como configuração inicial e CRV - (c) PRIM e GRASP como configuração inicial e SRV/CRV.....	112
Figura 35 - Sistemas de 84 barras.....	113
Figura 36 - Soluções obtidas pelo algoritmo GRASP para o sistema de 84 barras.	115
Figura 37 – Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 84 barras e SRV - (a) Solução base como configuração inicial – (b) PRIM como configuração inicial – (c) GRASP como configuração inicial.	117
Figura 38 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 84 barras e CRV - (a) Solução base como configuração inicial – (b) PRIM como configuração inicial – (c) GRASP como configuração inicial.	118
Figura 39 - Melhores soluções da BT a cada iteração, para o sistema de 84 barras – (a) Solução base como configuração inicial e SRV/CRV – (b) PRIM como configuração inicial e SRV/CRV – (c) GRASP como configuração inicial e SRV/CRV.	119
Figura 40 – Histórico de soluções do PR – (a) Solução base como configuração inicial e SRV – (b) Solução base como configuração inicial e CRV – (c) PRIM como configuração inicial e SRV – (d) PRIM como configuração inicial e CRV – (e) GRASP como configuração inicial e SRV – (f) GRASP como configuração inicial e CRV.....	121
Figura 41 - Sistema de 136 barras.	122
Figura 42 - Soluções obtidas pelo algoritmo GRASP para o sistema de 136 barras.	124
Figura 43 – Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 84 barras SRV – (a) Solução base como configuração inicial – (b) PRIM como configuração inicial – (c) GRASP como configuração inicial.	127
Figura 44 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 136 barras CRV - (a) Solução base como configuração inicial – (b) PRIM como configuração inicial – (c) GRASP como configuração inicial.	128
Figura 45 – Melhores soluções da BT a cada iteração, para o sistema de 136 barras – (a) Solução base como configuração inicial SRV – (b) Solução base como configuração inicial CRV - (c) PRIM como configuração inicial SRV – (d) PRIM como configuração inicial CRV - (e) GRASP como solução inicial SRV - (f) GRASP como solução inicial CRV.	129
Figura 46 - Histórico de soluções do PR – (a) Solução base como configuração inicial SRV – (b) Solução base como configuração inicial CRV – (c) PRIM como configuração inicial SRV – (d) PRIM como configuração inicial CRV – (e) GRASP como configuração inicial SRV – (f) GRASP como configuração inicial CRV.	131
Figura 47 - Sistema de 415 barras.	133
Figura 48 - Soluções obtidas pelo GRASP para o sistema de 415 barras – (a) $\alpha = 0,4$ – (b) $\alpha = 0,1$	135
Figura 49 – Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 415 barras com a solução base como configuração inicial – (a) SRV – (b) CRV.....	140

Figura 50 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 415 barras com o PRIM como configuração inicial – (a) SRV – (b) CRV.....	141
Figura 51 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 415 barras com o GRASP como configuração inicial e indicador de sensibilidade $\alpha = 0,4$ – (a) SRV – (b) CRV.	142
Figura 52 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 415 barras com o GRASP como configuração inicial e indicador de sensibilidade $\alpha = 0,1$ – (a) SRV – (b) CRV.	143
Figura 53 - Melhores soluções da BT – (a)-(b) Solução base como configuração inicial e SRV/CRV – (c)-(d) PRIM como configuração inicial SRV/CRV - (e)-(f) GRASP como configuração inicial, $\alpha = 0,4$ SRV/CRV – (g)-(h) GRASP como configuração inicial, $\alpha = 0,1$ SRV/CRV.....	144
Figura 54 - Histórico de soluções do PR – (a) Solução base como configuração inicial SRV – (b) Solução base como configuração inicial CRV – (c) PRIM como configuração inicial SRV – (d) PRIM como configuração inicial CRV.	146
Figura 55 - Histórico de soluções do PR – (a) GRASP como configuração inicial, $\alpha = 0,4$ SRV – (b) GRASP como configuração inicial, $\alpha = 0,4$ CRV – (c) GRASP como configuração inicial, $\alpha = 0,1$ SRV – (d) GRASP como configuração inicial, $\alpha = 0,1$ CRV.	147
Figura 56 - Histórico de soluções do procedimento de BT para o sistema de 14 barras – (a) Soluções geradas por iterações consecutivas com a solução base como configuração inicial – (b) Soluções ordenadas em ordem decrescente de média dos fluxos de carga dos ramos com chave fechada por iteração da BT com a solução base como configuração inicial – (c) Soluções geradas por iterações consecutivas com PRIM/GRASP como configuração inicial – (d) Soluções ordenadas em ordem decrescente de média dos fluxos de carga dos ramos com chave fechada por iteração da BT com PRIM/GRASP como configuração inicial.	149
Figura 57 - Histórico de soluções do procedimento de BT para o sistema de 33 barras – (a) Soluções geradas por iterações consecutivas com a solução base como configuração inicial – (b) Soluções ordenadas em ordem decrescente de média dos fluxos de carga dos ramos com chave fechada por iteração da BT com a solução base como configuração inicial – (c) Soluções geradas por iterações consecutivas com PRIM/GRASP como configuração inicial – (d) Soluções ordenadas em ordem decrescente de média dos fluxos de carga dos ramos com chave fechada por iteração da BT com PRIM/GRASP como configuração inicial.	150
Figura 58 - Histórico de soluções do procedimento de BT para o sistema de 84 barras – (a)-(c)-(e) Soluções geradas por iterações consecutivas com a solução base, PRIM e GRASP como configuração inicial, respectivamente – (b)-(d)-(f) Soluções ordenadas em ordem decrescente de média dos fluxos de carga dos ramos com chave fechada por iteração da BT com a solução base, PRIM e GRASP como configuração inicial, respectivamente.....	151
Figura 59 - Cálculo de corrente de potência.	171
Figura 60 - Cálculo de tensão na barra.	172
Figura 61 - Configuração radial do sistema de distribuição em Árvore Dinâmica – Passo 1.....	193
Figura 62 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 2.	194
Figura 63 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 3.	195
Figura 64 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 4.	196
Figura 65 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 5.	197
Figura 66 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 6.	197
Figura 67 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 7.	198
Figura 68 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 8.	199

Figura 69 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 9.	200
Figura 70 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 10.....	201
Figura 71 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 11.....	201
Figura 72 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 12.....	202
Figura 73 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 13.....	203
Figura 74 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 14.....	204

LISTA DE TABELAS

Tabela 1 - Representação de quatro iterações do procedimento de PR para o problema do Caixeiro Viajante.	28
Tabela 2 - Representação das soluções geradas em uma determinada vizinhança.	87
Tabela 3 - Preenchimento do vetor de ciclos.	89
Tabela 4 - Ramos válidos contidos no vetor de ciclos.	90
Tabela 5 – Status resultantes das chaves dos ramos da linha i de M_ciclos	90
Tabela 6 - Dimensão do espaço de soluções para diferentes sistemas e formas de codificação.	91
Tabela 7 - Tempos de execução (ms) totais para o procedimento de construção e destruição da árvore dinâmica e o procedimento de renumeração de barras.	95
Tabela 8 - Tempos de execução (ms) para o fluxo de potência em árvore dinâmica.	97
Tabela 9 - Estratégias de obtenção de uma solução inicial para o sistema de 14 barras.	101
Tabela 10 - Solução final obtida para o sistema de 14 barras.	102
Tabela 11 - Quantidades de soluções geradas/avaliadas e tempos de execução da BT e PR.	102
Tabela 12 - Estratégias de obtenção de solução inicial para o sistema de 33 barras.	107
Tabela 13 - Solução final obtida para o sistema de 33 barras.	108
Tabela 14 - Quantidades de soluções geradas/avaliadas, e tempos de execução das técnicas de BT e PR.	108
Tabela 15 - Estratégias de obtenção de solução inicial para o sistema de 84 barras.	114
Tabela 16 - Solução final obtida para o sistema de 84 barras.	115
Tabela 17 - Quantidade de soluções geradas/avaliadas, e tempos de execução das técnicas de BT e PR.	116
Tabela 18 - Estratégias de obtenção de solução inicial para o sistema de 136 barras.	123
Tabela 19 - Solução final obtida para o sistema de 136 barras.	125
Tabela 20 - Quantidades de soluções geradas/avaliadas e tempos de execução das técnicas BT e PR.	125
Tabela 21 - Estratégias de obtenção de solução inicial para o sistema de 415 barras.	134
Tabela 22 - Solução final obtida para o sistema de 415 barras.	136
Tabela 23 - Quantidades de soluções geradas/avaliadas, e tempos de execução das técnicas de BT e PR.	138
Tabela 24 - Dados do Sistema de 14 barras.	174
Tabela 25 - Dados do Sistema de 33 Barras.	175
Tabela 26 – Dados do sistema de 84 barras.	177
Tabela 27 – Dados do sistema de 136 barras.	179
Tabela 28 – Dados do sistema de 415 barras.	182

LISTA DE ABREVIATURAS

RSDEE	Reconfiguração de Sistemas de Distribuição de Energia Elétrica
GRASP	<i>Greedy Randomized Adaptive Procedure</i> – Procedimento Adaptativo Guloso Aleatório
BT	Busca Tabu
PR	<i>Path Relinking</i>
SDEE	Sistemas de Distribuição de Energia Elétrica
P	Polinomial
NP	Não Polinomial
NPC	Não Polinomial Completo
AHC	Algoritmo Heurístico Construtivo
SDH	<i>Steepest Descent Heuristic</i> – Algoritmo de Busca Através de Vizinhança
AG	Algoritmo Genético
SA	<i>Simulated Annealing</i> – Recozimento Simulado
RCL	<i>Restricted Candidate List</i> – Lista de Candidatos Restritos
UVDA	<i>Uniform Voltage Distribution Based Constructive Algorithm</i> – Algoritmo Construtivo Baseado em Distribuição Uniforme de Tensão
DSSHA	<i>Dynamic Switches Set Heuristic Algorithm</i> – Algoritmo Heurístico de Conjuntos de Chaves Dinâmicas
VNS	<i>Variable Neighborhood Search</i> – Busca em Vizinhança Variável
MC	Método da Compensação
MNR	Método de Newton-Raphson
RNA	Redes Neurais Artificiais
MBF	Método <i>Backward/Forward</i>
FBR	Funções de Base Radiais
LIFO	<i>Last In First Out</i> – O último que entra é o primeiro que sai
p.u.	Por Unidade
kV	Kilo Volt
kVA	Kilo Volt-Ampére
SRV	Sem Redução de Vizinhança
CRV	Com Redução de Vizinhança

LISTA DE SÍMBOLOS

Capítulo 2

min	Procedimento de Minimização de um problema
v	Função objetivo
x	Solução factível
X	Região factível
S	Espaço de soluções
\mathfrak{R}^n	Conjunto dos números reais
n	Tamanho do problema em tratamento
M_{ij}	Elemento da linha i e coluna j de uma matriz para o problema de Ordenamento Linear
S_{ini}	Solução inicial para o procedimento de <i>Path Relinking</i>
S_{guia}	Solução guia para o procedimento de <i>Path Relinking</i>
S_{int_i}	Soluções intermediárias do procedimento de <i>Path Relinking</i>
d	Qualidade da função objetivo para o problema do Caixeiro Viajante

Capítulo 3

v	Função objetivo
i	Condição inicial para percurso das barras de um sistema elétrico
n	Quantidade de barras de um sistema elétrico
P_i	Fluxo de potência ativa saindo de uma barra i
Q_i	Fluxo de potência reativa saindo de uma barra i
r_i	Resistência da linha de transmissão que está saindo da barra i
V_i	Magnitude da tensão na barra i
min	Procedimento de minimização de um problema
Ω_l	Conjunto de todos os ramos de um sistema elétrico
Ω_{b_i}	Conjunto de barras conectadas à barra i
Ω_b	Conjunto de todas as barras do sistema elétrico

P_{S_i}	Potência Ativa fornecida pela subestação na barra i
Q_{S_i}	Potência Reativa fornecida pela subestação na barra i
P_{D_i}	Demandas ativas da barra i
Q_{D_i}	Demandas reativas da barra i
P_{ij}	Fluxo de potência ativa no ramo pertencente às barras ij
Q_{ij}	Fluxo de potência reativa no ramo pertencente às barras ij
n_b	Número de barras do sistema elétrico
g_{ij}	Condutância do ramo pertencente às barras ij
V_i	Tensão na barra i
\underline{V}	Limite mínimo do módulo de tensão
\overline{V}	Limite máximo do módulo de tensão
θ_{ij}	Diferença angular entre as tensões das barras i e j
$I_{r_{ij}}$	Componente real do fluxo de corrente no ramo pertencente às barras ij
$I_{x_{ij}}$	Componente imaginária do fluxo de corrente no ramo pertencente às barras ij
\bar{I}_{ij}	Fluxo máximo de corrente do ramo pertencente às barras ij
n_{ij}	Valores binários que representam o status de chave do ramo pertencente às barras ij
d	Nível de demanda
c_d^{ls}	Custo das perdas elétricas no nível de demanda d
Δ_d	Período de duração do nível de demanda

Capítulo 4

n	Tamanho do problema em tratamento
$O(n)$	Complexidade de tempo linear de um algoritmo
$O(n^2)$	Complexidade de tempo quadrática de um algoritmo
$T(n)$	Tempo de execução de um algoritmo
$O(1)$	Complexidade constante
$O(\log \log n), O(\log n),$ $O(n \log n)$	Complexidades de tempo logarítmicas de um algoritmo
$O(n^k)$	Complexidade de tempo polinomial de um algoritmo

$O(k^n), O(n!), O(n^n)$	Complexidades de tempo exponenciais de um algoritmo
$dmin$	Variável que armazena o maior fluxo de potência aparente em uma iteração do algoritmo PRIM
i, j, m, k, l	Índice do ramo, barra do sistema elétrico, ou linha da matriz de ciclos
α	Indicador de aleatoriedade do algoritmo GRASP com intervalo $[0,1]$
Ω	Conjunto de ramos e barras do sistema elétrico (Domínio)
FC_{ij}^{min}	Fluxo de potência aparente mínimo
FC_{ij}^{max}	Fluxo de potência aparente máximo
$v1, v2$	Índices de barras do sistema elétrico

SUMÁRIO

1	INTRODUÇÃO	6
1.1	JUSTIFICATIVA	7
1.2	OBJETIVOS E CONTRIBUIÇÕES	8
1.3	ESTRUTURA DO TRABALHO	9
2	OTIMIZAÇÃO, HEURÍSTICAS E META-HEURÍSTICAS	11
2.1	HEURÍSTICAS	13
2.2	META-HEURÍSTICAS.....	15
2.2.1	Busca Tabu	17
2.2.1.1	Memória Adaptativa	18
2.2.1.2	Exploração Sensível.....	19
2.2.1.3	Exemplos de Aplicação	20
2.2.2	GRASP.....	21
2.2.2.1	Exemplos de Aplicação	23
2.2.3	Path Relinking.....	24
2.2.3.1	Exemplos de Aplicação	27
3	O PROBLEMA DA RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA - RSDEE	29
3.1	MODELO MATEMÁTICO DO PROBLEMA DE RSDEE	31
3.2	MÉTODOS DE OTIMIZAÇÃO APLICADOS AO PROBLEMA DE RSDEE.....	34
3.3	CÁLCULO DE FLUXO DE POTÊNCIA INCORPORADO AOS MÉTODOS DE OTIMIZAÇÃO	46
4	IMPLEMENTAÇÃO E METODOLOGIA ADOTADA	50
4.2	FLUXOGRAMA DA PROPOSTA.....	56
4.3	FORMATAÇÃO DOS DADOS E FUNCIONALIDADES	60
4.3.1	Formatação Inicial dos Dados.....	61
4.3.2	Configuração de uma Solução Inicial	63
4.3.2.1	Solução Base.....	64
4.3.2.2	O Algoritmo PRIM	64
4.3.2.3	O Algoritmo GRASP	66
4.3.3	Busca Tabu como procedimento de Busca Local	70
4.3.4	Path Relinking como procedimento de Pós-Otimização	74
4.3.3	Configuração de uma Solução Radial em uma Árvore Dinâmica	77
4.3.4	Perdas Elétricas de uma Solução	80
4.3.5	Estrutura e Tratamento de Vizinhança.....	85
4.3.6	Estratégia de Redução de Vizinhança.....	91
5	TESTES E RESULTADOS	94
5.1	CONSTRUÇÃO DE UMA SOLUÇÃO	95
5.2	CÁLCULO DO FLUXO DE POTÊNCIA.....	97

5.3	PROCEDIMENTO BT/GRASP/PR PARA O PROBLEMA DE RSDEE	99
5.3.1	Sistemas de 14 barras.....	100
5.3.2	Sistemas de 33 barras.....	106
5.3.3	Sistemas de 84 barras.....	113
5.3.4	Sistemas de 136 barras.....	122
5.3.5	Sistemas de 415 barras.....	132
5.4	REDUÇÃO DE VIZINHANÇA	148
5.5	DISCUSSÕES DOS RESULTADOS	152
6	CONSIDERAÇÕES FINAIS.....	156
6.1	TRABALHOS FUTUROS.....	156
	REFERÊNCIAS.....	160
	ANEXO A – PROBLEMAS DE ALTA COMPLEXIDADE	167
	ANEXO B – FLUXO DE POTÊNCIA EM SISTEMAS RADIAIS	170
	B.1CÁLCULO DA CORRENTE DE POTÊNCIA	170
	B.2CÁLCULO DE TENSÃO DA BARRA	171
	B.3CÁLCULO DAS PERDAS ELÉTRICAS NO SISTEMA DE DISTRIBUIÇÃO...172	
	B.4ALGORITMO DO PROCEDIMENTO DE FLUXO DE POTÊNCIA	173
	ANEXO C – DADOS DOS PROBLEMAS TESTADOS	174
	C.1SISTEMA DE 14 BARRAS	174
	C.2SISTEMA DE 33 BARRAS	175
	C.3SISTEMA DE 84 BARRAS	176
	C.4SISTEMA DE 136 BARRAS.....	178
	C.5SISTEMA DE 415 BARRAS.....	182
	ANEXO D – ILUSTRAÇÃO DO PROCEDIMENTO DE CONSTRUÇÃO DE UMA CONFIGURAÇÃO RADIAL EM ÁRVORE DINÂMICA	193

1 INTRODUÇÃO

O problema de reconfiguração de sistemas de distribuição de energia elétrica (RSDEE) tem o objetivo de encontrar uma topologia radial de um sistema de distribuição de energia elétrica, por meio de operações de chaveamentos para minimizar, normalmente, as perdas elétricas no sistema. A dimensão da rede pode sugerir que uma grande quantidade de combinações de chaves abertas e fechadas nos circuitos (ramos ativos e inativos) sejam avaliadas e o número de soluções com configuração radial deve ser analisada, trazendo uma explosão combinatória do número de configurações para análise.

As estratégias heurísticas e meta-heurísticas têm o objetivo de encontrar soluções de qualidade para problemas com grande capacidade combinatória sem a necessidade da modelagem matemática formal do problema. A qualidade das soluções encontradas é guiada por estratégias eficientes através de espaços de busca de soluções até que um critério de parada seja satisfeito.

Existe uma grande diversidade de procedimentos heurísticos e meta-heurísticos, cada um com suas especificidades que buscam, de forma diferente, a maneira com que se explora os espaços de soluções, a estratégia de construção de novas soluções, entre outras características.

Neste trabalho, é apresentada a aplicação das técnicas heurísticas e meta-heurísticas GRASP (*Greedy Randomized Adaptive Procedure*), Busca Tabu (BT) e *Path Relinking* (PR) para o tratamento do problema de RSDEE operando em um nível de demanda preestabelecido. Estes algoritmos atuarão em conjunto, agregados, onde o procedimento GRASP será utilizado somente como gerador de uma solução inicial, a BT como o procedimento de busca local a partir da solução inicial fornecida pelo GRASP, e então o PR atuará como técnica de pós-otimização de uma lista das melhores soluções obtidas pela BT. Também é proposto um método rápido e eficiente para o cálculo de fluxo de potência de sistemas de distribuição de energia elétrica (SDEE) radiais baseado em programação dinâmica, utilizando árvores de filhos variados (CORMEN *et al.*, 2001; DROZDEK, 2012) e a estratégia *backward/forward* (CHENG; SHIRMOHAMMADI, 1995).

Além da proposta de aplicação das três técnicas meta-heurísticas mencionadas anteriormente, em conjunto, este trabalho traz novas abordagens de como armazenar e tratar os dados do sistema elétrico. São propostos dois procedimentos, onde um deles representa a construção de uma configuração radial para um SDEE em árvore dinâmica, e o outro

representa a forma de leitura dessa configuração radial em árvore dinâmica para efetuar o cálculo do fluxo de potência.

A construção de uma configuração radial de um SDEE por meio de uma árvore dinâmica representaria o procedimento clássico de renumeração de barras que requer que as barras de um sistema sejam numeradas de forma sequencial para que o procedimento de cálculo do fluxo de potência possa ocorrer (RENATO CESPEDES G., 1990; SHIRMOHAMMADI, 1988). Essa nova estratégia de construção de uma solução não requer mais o procedimento de renumeração de barras, pois utiliza a própria informação da barra no momento de leitura da representação de uma configuração radial de um SDEE em árvore dinâmica.

O procedimento de cálculo de fluxo de potência realiza a leitura da configuração radial de um SDEE, construída em árvore dinâmica, utilizando conceitos de percurso em árvores, de forma recursiva (CORMEN *et al.*, 2001; DROZDEK, 2012). São apresentados os algoritmos para a construção de uma configuração radial de um problema de SDEE em árvore dinâmica e o método de percurso desta configuração radial para se realizar o procedimento de cálculo de fluxo de potência.

Os testes foram executados com SDEEs de pequeno, médio e grande porte e os resultados foram comparados com as técnicas de Gespedes (RENATO CESPEDES G., 1990; SHIRMOHAMMADI, 1988), Prim (CORMEN *et al.*, 2001; DROZDEK, 2012) e Possagnolo (POSSAGNOLO, 2015)

1.1 JUSTIFICATIVA

As perdas técnicas na distribuição de energia elétrica são da ordem de 8% no Brasil (ANEEL - AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA, [s.d.]) e, portanto, pesquisas que visem a redução da referida porcentagem tornam-se pertinentes no mercado, ainda mais se comportarem um baixo custo de implementação. A adoção de novas técnicas e metodologias de tratamento fazem com que as empresas possam otimizar o planejamento da operação e expansão de seus sistemas elétricos, o que faz aumentar sua eficiência.

As empresas distribuidoras e os consumidores se beneficiam com a redução das perdas técnicas como a redução de custos de operação e postergar a necessidade de reforços na rede

elétrica, melhorando o perfil de tensão do sistema. A redução das perdas implica em uma menor utilização de combustíveis fósseis com o menor uso de usinas termoeletricas, por exemplo.

1.2 OBJETIVOS E CONTRIBUIÇÕES

O principal objetivo deste trabalho é agregar as heurísticas e meta-heurísticas GRASP, BT e PR, para tratar o problema de RSDEE para um nível de demanda preestabelecidos.

O método GRASP foi escolhido por ser um procedimento construtivo que gera soluções com rapidez e boa qualidade. Métodos construtivos, não exatamente o GRASP, são muito utilizados para construção de soluções de boa qualidade para a resolução de problemas RSDEE.

A BT é uma meta-heurística que sempre obtém soluções de boa qualidade, senão ótimas. Pode haver uma grande quantidade de parâmetros, o que aumenta sua complexidade de implementação e tratamento. Cada parâmetro possui uma especificidade na BT com o objetivo de se buscar novas configurações e novos espaços de busca. É uma meta-heurística muito utilizada em diversos problemas de Pesquisa Operacional por características como bloqueio/liberação de elementos com recente troca de atributos; técnicas para a diversificação de soluções geradas; utilização da intensificação para verificar e avaliar novos espaços de busca promissores, entre outras.

O PR veio inicialmente como proposta de obter diferentes configurações para as soluções geradas e evolui para patamares superiores. Tal técnica pode atuar como uma meta-heurística pura ou como um procedimento de pré-otimização ou pós-otimização de outro processo heurístico ou meta-heurístico. A geração das soluções é realizada de forma muito rápida produzindo escassas soluções vizinhas, uma ou duas normalmente, mas são soluções que, normalmente, não são alcançadas pela busca.

Os três procedimentos aqui descritos já foram utilizados para se resolver o problema de RSDEE, alguns de forma agregada em pares, mas nunca os três procedimentos foram utilizados juntos, agregados, o que reforça esta proposta de estudo.

Uma nova estratégia para a redução de vizinhança é apresentada, visto que conforme se aumenta a complexidade dos sistemas elétricos adotados, a explosão combinatória torna-se

maior e os espaços de busca se tornam muito grandes, dependendo da vizinhança adotada e, torna-se necessário um tratamento mais refinado no espaço de busca no intuito de reduzi-los.

Outra adição importante a este trabalho foi trazer uma abordagem diferente da implementação, combinando os paradigmas estático e dinâmico, para se obter uma mesma representação entre a árvore do sistema radial e a árvore da estrutura de dados utilizada. Isto, juntamente com outras estratégias implementadas, fizeram com que se modificasse a forma com que as informações de um problema de RSDEE fossem armazenadas e tratadas, inovando em vários aspectos a forma com que se constrói uma configuração radial para um RSDEE e como se calcula seu fluxo de potência, resultando em uma nova maneira de tratar o problema em questão e de como pode ser estudado.

1.3 ESTRUTURA DO TRABALHO

O trabalho está organizado da seguinte forma:

No Capítulo 2, apresentam-se as definições de um processo de otimização, com ênfase especial para os procedimentos heurísticos e meta-heurísticos. São apresentados diversos problemas, sobre os quais são aplicadas as técnicas de otimização. Uma atenção maior é dada às definições de procedimentos meta-heurísticos com maior enfoque nas técnicas aqui abordadas.

O capítulo 3 trata a definição do problema de reconfiguração de sistemas de distribuição de energia elétrica - RSDEE - para um nível de demanda preestabelecido, apresentando as principais características do problema, sua modelagem matemática e uma revisão dos trabalhos mais referenciados dos últimos 10 dez anos sobre a aplicação de procedimentos de otimização ao problema de RSDEE, juntamente com um estudo sobre os procedimentos de otimização adotados para o cálculo de fluxo de potência.

No Capítulo 4, explicam-se a metodologia adotada, os procedimentos de implementação, as estruturas de dados utilizadas e suas razões, formatação de dados do problema de RSDEE, fluxograma do algoritmo como um todo, apresentação de algoritmos explicativos de suas principais funcionalidades e detalhamento das funções utilizadas com identificações passo-a-passo dos procedimentos implementados.

No Capítulo 5, apresentam-se as condições de testes e resultados obtidos comparando-se os resultados obtidos entre a metodologia proposta com os resultados da literatura especializada.

Para finalizar o documento, no Capítulo 6 traz-se as Considerações Finais e propostas de estudos futuros com sugestões de melhorias.

Como informações adicionais, mas não menos importantes, são apresentados os anexos. No Anexo A são apresentadas explicações resumidas de cada problema identificado no Capítulo 2. O Anexo B traz toda a descrição detalhada para o cálculo do problema de fluxo de potência em sistemas elétricos radiais. No Anexo C, apresentam-se os conjuntos de dados de teste dos sistemas elétricos adotados para 14, 33, 84, 136 e 417 barras e, no Anexo D, apresentam-se ilustrações, passo a passo, da metodologia utilizada para construir a configuração de uma solução para o problema RSDEE na estrutura de dados de árvore dinâmica.

2 OTIMIZAÇÃO, HEURÍSTICAS E META-HEURÍSTICAS

Um problema clássico e genérico de otimização pode ser modelado como a seguir:

$$\min\{v(x)|x \in X, X \subseteq S\} \quad (1)$$

onde S indica o espaço de soluções, X a região factível, $x \in X$ é uma solução factível e v uma função objetivo, $\sqrt{} : \mathfrak{R}^n \rightarrow \mathfrak{R}$. Caso S seja inteiro obtemos uma otimização combinatória, caso $S = \mathfrak{R}^n$ obtem-se uma otimização contínua. Uma solução ótima $x^* \in X$ que satisfaça (1) pode ser representada da seguinte forma (POSSAGNOLO, 2015):

$$v(x^*) \leq v(x), \forall x \in X \quad (2)$$

Utilizando procedimentos exatos, pode-se encontrar a solução ótima para a inequação (2) ou mostrar que a solução é ilimitada ($X = \emptyset$). O tempo de processamento para se encontrar tal solução deve ser finito e não muito extenso, podendo admitir certos graus de tolerância em alguns casos.

Os algoritmos são subdivididos inicialmente em três categorias sendo P (polinomial), NP (não-polinomial) e NPC (não polinomial completo). Tais categorias representam problemas resolvidos em tempo polinomial (P) ou não-polinomial (NP, NPC) indicando problemas tratáveis ou não, computacionalmente (CORMEN *et al.*, 2001).

Diversos problemas da área de Pesquisa Operacional, Engenharia Elétrica, Bioinformática, Matemática Aplicada, que assumem a forma da equação (1) são bastante complexos para que uma solução ótima seja encontrada em tempo aceitável. Tais problemas encontram-se na categoria NP-Completo (CORMEN *et al.*, 2001) por possuírem uma complexidade computacional alta, ou seja, não havendo uma resolução em tempo polinomial. Se para alguns desses problemas, houver soluções em tempo polinomial, isso indicaria que todos os problemas NP-Completo poderiam ser resolvidos em tempo computacional polinomial. Entretanto, mesmo quando o problema pode ser resolvido em tempo computacional polinomial, seu polinômio pode ter fatores de ordem tão elevada que o tempo computacional para se encontrar uma solução de boa qualidade ou quase ótima é muito extenso.

Os problemas não tratáveis estão incluídos na categoria NPC. As soluções para esses problemas envolvem análises combinatórias que se tornam mais complexas conforme a quantidade de dados de entrada (dimensão do problema) aumenta. As técnicas heurísticas e

meta-heurísticas atuam como algoritmos de análise combinatória e se encaixam na categoria de problemas do tipo P, e são muito utilizadas para resolver problemas das classes NPC e também NP. Todos os problemas tratados são reais e modelados matematicamente, dessa forma busca-se encontrar uma solução para o modelo matemático.

As complexidades dos algoritmos são medidas subjetivas, mas matematicamente consistentes (CORMEN *et al.*, 2001). Assim também são as meta-heurísticas, já que elas são utilizadas para encontrar soluções para o modelo matemático de um problema. As soluções encontradas são, ou devem ser, as mais próximas possíveis de uma solução do/no mundo real.

Heurísticas e meta-heurísticas são técnicas computacionais de análise combinatória que resolvem problemas computacionais, nos quais ainda não existem algoritmos eficientes, na tentativa de se encontrar a melhor solução (ótima global) ou soluções que estão entre as melhores (ótimos locais), em tempos computacionais aceitáveis. Para isso, devem ser atualizadas maneiras de se conseguir escapar de ótimos locais, melhorando ou piorando a solução encontrada por meio de alguma perturbação (vizinhança, intensificação, diversificação), para que se possa alcançar uma solução ótima global.

As meta-heurísticas são técnicas utilizadas para se resolver problemas de análise combinatória ou global. Possuem a grande vantagem de se adaptarem facilmente às características de um problema quando a complexidade é alterada. Diferente das técnicas clássicas de otimização, as meta-heurísticas não necessitam de serem guiadas pela modelagem matemática do problema (POSSAGNOLO, 2015).

Existem diversos problemas para os quais se torna necessária a utilização de algumas técnicas como: ordenamento linear (CHAOVALITWONGSE *et al.*, 2011; LAGUNA; MARTI; CAMPOS, 1999); sequenciamento de tarefas (AIEX; BINATO; RESENDE, 2003; GONZÁLEZ; VELA; VARELA, 2015; JIA; HU, 2014; PENG; LÜ; CHENG, 2015); minimização de banda (PIÑANA *et al.*, 2004); p-mediana (DÍAZ; FERNÁNDEZ, 2006); diversidade máxima e mínima (DUARTE; MARTÍ, 2007; MARTÍ; SANDOYA, 2013; RESENDE *et al.*, 2010); produção e distribuição (ARMENTANO; SHIGUEMOTO; LØKKETANGEN, 2011; BOUDIA; LOULY; PRINS, 2007); alocação de recursos (ALVAREZ-VALDES *et al.*, 2008); dimensionamento de lote (NASCIMENTO; RESENDE; TOLEDO, 2010); problemas globais de otimização em larga escala (DUARTE; MARTÍ; GORTAZAR, 2011); *truck* e *trailer* (VILLEGAS *et al.*, 2011); programação quadrática binária (WANG *et al.*, 2012); localização de rotas com dois escalões (NGUYEN; PRINS; PRODHON, 2012); projeto de rede com dois caminhos (BARBALHO *et al.*, 2013),

distribuição de rede elétrica (BENAVIDES *et al.*, 2013); caixeiro viajante (MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013); problema da mochila (ARIN; RABADI, 2016) e partição de números (RODRIGUEZ *et al.*, 2017). Uma descrição resumida de cada problema aqui descrito pode ser encontrada no Anexo A.

2.1 HEURÍSTICAS

As técnicas heurísticas são as de otimização que buscam soluções de boa qualidade em tempos computacionais aceitáveis. São aplicadas quando não se possui um método exato de otimização para um determinado problema, quando a solução ótima não é importante, quando se depara com limitações, sejam elas de tempo de execução ou de capacidade da máquina, e quando se pretende encontrar boas soluções iniciais para serem tomadas como ponto de partida de um processo de otimização mais elaborado.

Algumas das técnicas heurísticas mais conhecidas são: Algoritmos Heurísticos Construtivos (AHC), Algoritmos de Decomposição, Algoritmos de Divisão, Algoritmos de Redução, Algoritmos de Manipulação do Modelo Matemático, Algoritmos de Busca Através de Vizinhança (SDH). Após essas, serão explicados, resumidamente, o AHC e o SDH.

Conhecido também como AHC, o Algoritmo Heurístico Construtivo (SILVÉRIO; RODRIGUES; STEINER, 2013), é uma das técnicas mais utilizadas para se resolver problemas complexos. Adotando uma abordagem gulosa (CORMEN *et al.*, 2001), a técnica é executada construindo-se uma solução por meio da inserção de elementos passo a passo até que se complete a solução a qual, geralmente, é de boa qualidade. Os elementos inseridos na solução são verificados por indicadores de sensibilidade para se avaliar os melhores elementos para participarem da solução em construção.

Para o problema do Caixeiro Viajante (CORMEN *et al.*, 2001), por exemplo, o indicador de sensibilidade pode ser adotado como a menor distância entre a última cidade visitada e todas as cidades restantes ainda não visitadas, o que pode trazer certos problemas, pois é visível que, quando se inicia sempre o problema do Caixeiro Viajante com a mesma cidade utilizando-se este indicador de sensibilidade sempre será trilhado o mesmo caminho. A fim de se obter diferentes configurações de soluções devem ser utilizados diferentes elementos iniciais, além de diferentes indicadores de sensibilidades (CORMEN *et al.*, 2001;

SILVÉRIO; RODRIGUES; STEINER, 2013). Essas são as grandes diferenças entre os diferentes AHCs implementados.

A Figura 1 ilustra a construção de uma solução para o problema do Caixeiro Viajante por meio do AHC. Verifica-se que, utilizando-se de somente um indicador de sensibilidade e partindo-se sempre de um mesmo ponto inicial a solução final será sempre a mesma.

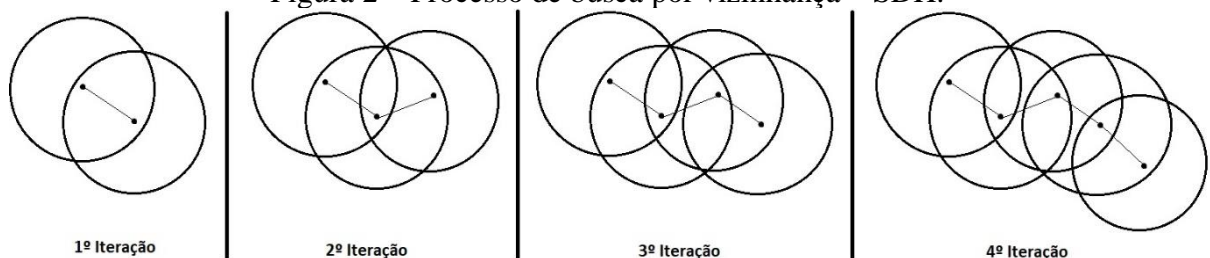
Figura 1 – Construção de uma solução do problema do Caixeiro Viajante por meio do AHC.



Fonte: Próprio autor.

O Algoritmo de Busca Através de Vizinhança (SDH – *Steepest Descent Heuristic*) (POSSAGNOLO, 2015) possui uma abordagem bem diferente do AHC. Enquanto o AHC trabalha com a construção de uma solução factível o SDH trabalha com a análise de soluções em um determinado espaço de busca. A Figura 2 apresenta um exemplo de como é realizado o processo de busca por vizinhança em um SDH.

Figura 2 – Processo de busca por vizinhança – SDH.



Fonte: Próprio autor.

O SDH parte de uma solução inicial factível, solução corrente, e por meio de determinados movimentos entre os elementos das soluções (troca de arcos de elementos 2-opt e 3-opt, troca entre pares de elementos) cria um espaço de busca no qual todas as soluções do espaço de busca são avaliadas, e no caso da melhor solução encontrada for melhor do que a solução corrente, esta se torna a nova solução corrente e o processo é reiniciado. O processo

pode finalizar de diversas maneiras como quantidade máxima de iterações ou quando em uma busca não encontram mais soluções melhores do que a solução corrente.

Heurísticas podem encontrar ou construir soluções de boa qualidade, mas elas podem falhar; ou exigirem um grande tempo de execução, quando se trata de encontrar as soluções de ótima qualidade.

Assim como em programação não linear soluções de boa qualidade podem se encontrar em espaços de soluções ótimas locais que necessitem de grandes esforços matemáticos/computacionais para que saiam desses espaços de soluções a fim de procurarem novos espaços de soluções com o objetivo de se atingir melhores resultados. As meta-heurísticas vem para cobrir estes tratamentos que requerem grandes esforços na busca de soluções de melhor qualidade inserindo regras, ou mesmo criando novas regras que tratem esses casos de forma mais específica (SILVÉRIO; RODRIGUES; STEINER, 2013).

2.2 META-HEURÍSTICAS

As meta-heurísticas são técnicas mais elaboradas de otimização com estratégias muito específicas para tratar diferentes problemas combinatórios. Na busca por soluções melhores, do que as soluções que as heurísticas podem encontrar, as meta-heurísticas trazem diferentes métodos de análise combinatória junto com habilidades que tratam a estagnação e a semelhança de soluções. Novamente, o processo heurístico não pode ser esquecido, uma vez que podem gerar, rapidamente, soluções de boa qualidade que sirvam como proposta de solução inicial para a realização de um processo meta-heurístico.

Basicamente as meta-heurísticas decorrem de características de duas heurísticas: busca heurística construtiva e busca através de vizinhança. A busca heurística construtiva normalmente (AHC, por exemplo) é utilizada para gerar soluções iniciais, já a busca através de vizinhança parte de uma solução obtida e realiza uma busca local em um espaço de soluções, na vizinhança da solução inicial, na tentativa de se encontrar soluções de melhor qualidade do que a inicial. Possui ainda, tratamentos específicos para a estagnação e semelhança de soluções.

Quando se pretende resolver um problema complexo de otimização usando meta-heurísticas, as seguintes estratégias devem ser seguidas:

- I. Uma forma de representar uma proposta de solução. Para o problema do caixeiro viajante, por exemplo, a proposta pode ser representada por um vetor unidimensional de tamanho n em que cada elemento do vetor é uma cidade (vértice do grafo) percorrida;
- II. Uma forma de verificar se a proposta de solução é factível ou não;
- III. Uma forma de avaliar a qualidade da solução. Nesse caso, basicamente se compara a função objetivo entre soluções e a que possuir melhor função objetivo se torna a nova solução incumbente;
- IV. Uma forma de gerar soluções vizinhas. Este procedimento é denominado de busca local e trata da elaboração de movimentos entre os elementos de uma dada solução para que esta gere soluções vizinhas no objetivo de encontrar melhores soluções.

Existem inúmeras técnicas meta-heurísticas como as descritas a seguir; as mais interessantes para este trabalho serão explicadas nas seções adiante, descrevendo aplicações para certos problemas:

- Algoritmo Genético - AG;
- Busca Tabu - BT;
- *Path Relinking* - PR;
- Chu-Beasley;
- *Simulated Annealing* - SA;
- GRASP – *Greedy Randomized Adaptive Search Procedure*;
- Colônia de Formigas;
- Colônia de Abelhas;
- Algoritmos Meméticos;
- Nuvem/Enxame de Partículas;
- Algoritmo da Formiga Leão;
- Algoritmo de Busca e de Otimização Cuco;
- Busca Harmônica;
- Evolução Diferencial;
- Otimização baseada em ensino-aprendizado.

2.2.1 Busca Tabu

Proposta, a princípio, por Fred Glover na década de 70 a BT é um algoritmo indicado para se revolver problemas de alta complexidade com estratégias que permitem a não estagnação em soluções ótimas locais. As técnicas adicionais, ou estratégias, que fazem com que a BT não estagne em soluções ótimas locais envolvem conceitos de memória adaptativa e de exploração sensível (GLOVER; LAGUNA, 2013). Na Figura 3, ilustram-se as principais características da BT.

Figura 3 – Principais características da Busca Tabu.

<p>Memória Adaptativa</p> <ul style="list-style-type: none"> Seletividade (incluindo esquecimento estratégico) Abstração e decomposição (usando memória explícita e por atributos) Tempo: <ul style="list-style-type: none"> Recência de eventos Frequência de eventos Diferenciação entre curto e longo prazo Qualidade e impacto <ul style="list-style-type: none"> Atração relativa quanto a escolhas alternativas Impacto de mudanças de relações em estrutura ou restrições Contexto <ul style="list-style-type: none"> Interdependência regional Interdependência estrutural Interdependência sequencial <p>Exploração Sensível</p> <ul style="list-style-type: none"> Imposição estratégica de proibições e induções (condições tabus e níveis de aspiração) Enfoque concentrado em boas regiões e em boas características das soluções (processos de intensificação) Caracterização e exploração de novas regiões promissoras (processo de diversificação) Padrões de busca não monótonos (oscilação estratégica) Integração e geração de novas soluções (PR)
--

Fonte: Adaptada de (GLOVER; LAGUNA, 2013).

A memória adaptativa permite a implementação de procedimentos que são capazes de realizar uma busca em um determinado espaço de solução de forma econômica e eficaz. Uma vez que a busca local é guiada por informações adquiridas durante a busca, a BT diverge de técnicas sem memória que se baseiam em processos semi-aleatórios que implementam uma

forma de amostragem. A exploração sensível, implementada de forma probabilística ou aleatória, parte da hipótese de que a escolha de uma estratégia ruim pode produzir mais informações para que a BT possa utilizar como parâmetros em suas estratégias de exploração dos espaços de busca (Glover e Laguna 2013).

2.2.1.1 *Memória Adaptativa*

Assim como um processo comum de busca local a BT é um procedimento iterativo de avaliação de solução num determinado espaço de busca adotando a melhor solução encontrada, quando houver, como a próxima solução corrente e reiniciando o processo até que se perfaça um determinado número de iterações, ou que não seja encontrada mais nenhuma solução melhor que a solução corrente no espaço de busca avaliado.

A BT permite que soluções que deteriorem a função objetivo sejam visitadas e adotadas como solução corrente. A estratégia resulta da utilização da memória de curto e longo prazo, perfazendo uma vizinhança modificada, que mantém históricos de soluções já visitadas, tratadas como um subconjunto. Para a memória de curto prazo são identificados elementos que diferem entre as vizinhanças modificadas (soluções de boa qualidade e deterioradas), já para a memória de longo prazo as soluções já encontradas de boa qualidade podem ser revisitadas.

Essa característica de vizinhança dinâmica necessita de movimentos de elementos diferentes para cada conjunto de vizinhança (construtiva e destrutiva) uma vez que tais movimentos podem vir a ser determinísticos produzindo sempre um mesmo conjunto de configurações quando se parte de uma mesma solução corrente. Há ainda, um gasto de memória para se armazenar as soluções encontradas, que, dependendo da dimensão do problema e da vizinhança adotada, a quantidade de soluções encontradas situa-se na casa dos milhares. Conta-se também com o consumo de tempo computacional para verificar se uma solução encontrada se encontra no histórico de soluções.

Outra forma de se trabalhar com a memória adaptativa na BT é por meio de memória de atributos. A estratégia serve para guiar a geração de soluções futuras por meio da avaliação dos atributos da solução que se modificam entre uma solução e outra. A recência controla os

atributos de soluções que se modificaram recentemente (últimas iterações), já a frequência avalia quantas vezes um atributo mudou ou não.

A memória baseada em recência seleciona atributos que aparecem em soluções recentes e os marca como tabu-ativos, e soluções que possuem tais elementos, ou combinações destes, são caracterizadas como tabu. Isso faz com que soluções recentes que pertençam à vizinhança modificada sejam ou não revisitadas (GLOVER; LAGUNA, 2013).

A memória baseada em frequência permite montar estratégias que podem ser avaliadas em duas diferentes medidas:

- Medida de transição: representam o número de iterações de que um atributo muda na solução visitada em uma trajetória;
- Medida de residência: número de iterações, pelas quais um atributo participou de uma solução visitada em uma trajetória, ou o número de instâncias onde um atributo pertenceu a um subconjunto de soluções.

O denominador pode ser representado por três tipos de métricas:

1. Número total de ocorrências de todos os eventos representados pelos numeradores;
2. A soma, ou média, dos numeradores;
3. O valor máximo dos numeradores.

Assim, a razão produz a frequência com que um atributo muda durante a geração das soluções e a frequência com que certos atributos participam de soluções (GLOVER; LAGUNA, 2013).

2.2.1.2 *Exploração Sensível*

A estratégia de exploração sensível envolve a aplicação das técnicas de memória em conjunto e/ou formas híbridas da BT com a adição de outras técnicas heurísticas, como o PR (ARIN; RABADI, 2016; BARBALHO *et al.*, 2013; CHAOVALITWONGSE *et al.*, 2011; LAGUNA; MARTI; CAMPOS, 1999; MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013; NGUYEN; PRINS; PRODHON, 2012; PENG; LÜ; CHENG, 2015; PIÑANA *et al.*, 2004; RESENDE *et al.*, 2010; VILLEGAS *et al.*, 2011). Um bom balanço entre a utilização dos aspectos de memória na BT nos leva a estratégias como a diversificação e intensificação.

A diversificação procura gerar soluções com o máximo de atributos, ou conjunto de atributos, diferentes entre elas. As memórias de curto prazo podem realizar tal tarefa, pois a análise de atributos entre soluções subsequentes pode proibir que certos atributos não possam ser modificados a cada movimento realizado, não permitindo que participem de um novo movimento durante uma quantidade de iterações do algoritmo, para o cuidado de tal elemento não retornar à sua posição anterior. Já o critério de aspiração remove tal proibição desde que o movimento a ser realizado melhore a função objetivo. Isso inibe que soluções sejam revisitadas e que novas configurações e espaços de busca de soluções sejam encontrados.

A intensificação possui a tarefa de escapar de espaços de busca com soluções ótimas locais para diferentes espaços de busca, às vezes deteriorando a função objetivo, a fim de buscar novas configurações de soluções, ou mesmo retornando para soluções de boa qualidade já visitadas, a fim de se aplicar uma busca mais profunda modificando-se a característica da vizinhança proposta. Para tal, utiliza-se uma lista de soluções elite que contém um subconjunto das melhores soluções já encontradas.

Ao processo de interação das técnicas de diversificação e intensificação percorrendo a fronteira de factibilidade dá-se o nome de oscilação estratégica. Outra característica que busca novas configurações de soluções é a aplicação da técnica de PR, a qual será explicada na seção a seguir.

2.2.1.3 Exemplos de Aplicação

Aplicando a BT para o problema do Caixeiro Viajante em um vetor unidimensional de tamanho n supõe-se que já se esteja partindo com uma solução inicial factível. A busca local por soluções de melhor qualidade possui um amplo espectro de movimentos já que não existem soluções infactíveis; dependendo do movimento adotado, aparecem somente soluções de melhor ou pior qualidade. Um dos movimentos mais utilizados é o 2-opt que permuta arcos de vértices produzindo uma grande quantidade de vizinhos de acordo com o tamanho do arco selecionado (MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013), mais precisamente $\frac{n(n-1)}{2}$ vizinhos para um problema de tamanho n .

Especificidades da BT, como os aspectos de memória, são facilmente inseridas no problema. Podem-se marcar a frequência com que certos elementos participaram das mesmas

posições nas melhores soluções encontradas e utilizar estes elementos nestas posições nas posições futuras. Outra estratégia é bloquear os movimentos de elementos que tenham acabado de mover por um número determinado de iterações, e permitindo a quebra dessa regra se o movimento gerar uma melhor solução. Uma última proposta é armazenar uma determinada quantidade de soluções que sirvam para comparação para que não sejam avaliadas novamente.

Adotando a característica de oscilação estratégica, a diversificação pode ficar a cargo da memória de curto prazo, uma vez que esta proíbe atributos em soluções posteriores criando soluções com quantidades maiores de elementos em diferentes posições. Na intensificação pode-se utilizar a lista de soluções elite para visitar configurações de boa qualidade e executar buscas com movimentos que explorem novos espaços de busca, ou adotar movimentos que deteriorem uma determinada solução, partindo para uma solução com pior função objetivo, para buscar novas configurações. A técnica de PR pode ser utilizada como processo de pós-otimização buscando novas soluções para cada par de soluções da lista de soluções elite.

Uma maneira de implementação da BT para o problema de Ordenamento Linear é a realização de trocas de elementos M_{ij} por M_{ji} de um determinado intervalo de uma coluna na tentativa de maximizar o setor acima da diagonal principal. A intensificação fica a cargo de uma lista Tabu que armazena os elementos que compõem uma solução em um movimento que denegriu a solução. A diversidade pode ser realizada pela frequência que um determinado setor escolhido, que é inversamente proporcional à sua chance de movimento.

A inclusão do PR pode ser realizada entre cada par de soluções elite, escolhendo o menor valor entre as posições dos elementos de um setor. Como as soluções intermediárias são muito próximas, uma busca local pode ser realizada a cada quatro iterações do PR (LAGUNA; MARTI; CAMPOS, 1999; WANG *et al.*, 2013).

2.2.2 GRASP

Idealizado por quem já tinha o domínio de técnicas heurísticas de otimização o GRASP – *Greedy Randomized Adaptive Search Procedure* - surgiu como uma união de duas técnicas heurísticas, o AHC e o SDH (FEO; RESENDE, 1989).

O algoritmo GRASP assume a seguinte estrutura para a solução de um problema genérico:

1. Implementar a fase de pré-processamento;
2. Realizar a fase de busca construtiva;
3. Realizar a fase de pós-processamento de busca local. Atualizar a incumbente se possível;
4. Se o critério de parada não for satisfeito, voltar ao passo 2. Caso contrário é indicado parar. A resposta é a solução armazenada.

Na fase de pré-processamento é realizado o estudo da estrutura do problema para que se avalie atributos ou conjunto de atributos a serem utilizados na fase construtiva, como características para o início do processo construtivo e a função que estabelece o indicador de sensibilidade para a construção de uma lista com as melhores soluções candidatas a participarem da construção da solução corrente diminuindo o espaço de soluções do problema.

O procedimento construtivo tem o objetivo de gerar uma solução de boa qualidade para o problema baseado em um algoritmo heurístico construtivo que escolhe, a cada passo, um elemento de uma lista de tamanho variável de elementos possíveis de fazerem parte da solução. Quando a solução estiver construída esta é submetida a um processo de busca local que consiste na busca em uma pequena vizinhança por soluções que possam ser melhores do que a obtida pelo GRASP. Caso a solução final seja melhor que a incumbente, atualiza-se a incumbente; e caso o critério de parada não seja satisfeito, uma nova iteração do método deve ocorrer (FEO; RESENDE, 1989).

A aleatoriedade do GRASP se situa na fórmula que define o espaço de busca no processo construtivo da solução inicial. A construção da solução ocorre inserindo um elemento da solução do espaço de busca por vez. Esse espaço de busca é reduzido, formando uma lista chamada de Lista de Candidatos Restritos – *RCL (Restricted Candidate List)* – onde a cada iteração um elemento é escolhido aleatoriamente para fazer parte da solução em construção. O controle da escolha dos elementos é realizado por um indicador de sensibilidade, que busca os elementos mais promissores que podem fazer parte da solução em construção. Desta forma a construção de uma solução se dá de forma rápida e geram-se soluções diferentes a cada momento que o algoritmo gera uma nova solução.

A criação da *RCL* também pode ser realizada de outras formas como funções de distribuição de probabilidade ou de pesos (FEO; RESENDE, 1989). Outra técnica que pode ser utilizada é o caso do GRASP Reativo (BOUDIA; LOULY; PRINS, 2007; DUARTE; MARTÍ, 2007) que controla a função gulosa do GRASP de acordo com o rendimento das soluções obtidas tornando o algoritmo mais aleatório ou mais determinístico.

A fase de busca local, também conhecida como processo de melhoria, foi originalmente formulada para ser executada com alguma estrutura de vizinhança, mas atualmente são utilizadas outras meta-heurísticas como BT, algoritmo genético, *simulated annealing*, entre outras. O principal objetivo é de se melhorar a solução obtida pelo processo construtivo do GRASP (FEO; RESENDE, 1989).

2.2.2.1 Exemplos de Aplicação

Ressalta-se uma forma de implementação do GRASP para o problema do Caixeiro Viajante. Inicialmente um vértice é selecionado de forma aleatória e se insere na solução corrente. Para cada último vértice da solução em construção avaliam-se todas as distâncias aos vértices não percorridos do grafo se escolhendo uma parcela, das menores distâncias, por exemplo, destes vértices ao último da solução em construção, configurando os elementos da lista restrita *RCL*.

Da lista, um vértice é aleatoriamente escolhido para fazer parte da solução, e o procedimento se repete até que a lista *RCL* seja vazia, e então o primeiro vértice escolhido é inserido ao final da solução (MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013).

Verifica-se que a falta de aleatoriedade por meio da escolha do próximo vértice, sendo de sempre escolhida pela menor distância encontrada a outro vértice, vai sempre produzir soluções de mesma configuração para um determinado vértice inicial, acabando com a diversidade das soluções encontradas. Esse problema é contornado de forma eficiente escolhendo em cada passo um elemento da lista *RCL*.

A busca local por soluções de melhor qualidade possui um amplo espectro de movimentos já que não existem soluções infactíveis, somente soluções de melhor ou pior qualidade. Um dos movimentos mais utilizados é o 2-opt que permuta arcos de vértices

produzindo uma grande quantidade de vizinhos de acordo com o tamanho do arco selecionado (MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013).

Considerando a aplicação do GRASP para o problema da Mochila, e, considerando-se que existe apenas uma mochila, uma solução inicial pode ser proposta criando-se um conjunto que contenha para cada item sua razão capacidade/custo, e escolher uma fração das melhores razões para se montar, no GRASP, a lista restrita de elementos *RCL*, e dessa lista um elemento é aleatoriamente escolhido para entrar na mochila. O procedimento termina quando não se consegue mais inserir novos elementos na mochila por um número determinado de iterações.

O método pode ser totalmente aleatório deixando a escolha dos elementos que irão participar da solução de forma aleatória. Na escolha de uma solução rápida, é eleito um elemento aleatório inicial da solução e o restante dos elementos é escolhido de acordo com as menores razões capacidade/custo, removendo a aleatoriedade.

Outra estratégia seria considerar as porcentagens de que um item é escolhido para fazer parte de uma solução. A lista restrita *RCL* é construída com elementos que possuam um intervalo de porcentagem, determinado por uma função. A fase de melhoria não é aplicada a todas as soluções factíveis encontradas, somente para uma pequena porcentagem das melhores soluções (ARIN; RABADI, 2016).

Dependendo do tipo de vizinhança adotada o comportamento de uma busca local pode gerar inúmeras soluções infactíveis, mas acaba gerando uma quantidade imensa de soluções, como a vizinhança 2-opt. Movimentos com menor impacto na solução corrente podem demorar em encontrar melhores soluções, mas tendem a gerar somente soluções factíveis, como a simples troca de um item por outro, ou remoção/adição de um item. A eficácia, eficiência e efetividade de uma vizinhança podem vir por estudos computacionais e podem depender da dimensão e características do problema.

2.2.3 Path Relinking

A modalidade surge como proposta de estratégia de intensificação, explorando a região existente entre duas soluções elite, provindas de uma BT ou busca dispersa (GLOVER, 1996).

A proposta consiste na escolha de duas soluções onde uma representa a solução inicial e a outra a solução guia, e por meio de troca de atributos da solução inicial pretende-se alcançar a solução guia. Isso faz com que se criem diversas soluções intermediárias no processo por meio de movimentos realizados em uma vizinhança mínima com tempo rápido de processamento. Outra característica da referida técnica é a grande diversidade de soluções que são geradas durante o processo, o que ajuda na exploração em espaços de soluções mais amplos.

A quantidade de soluções intermediárias geradas depende diretamente dos movimentos realizados entre uma solução intermediária e outra, normalmente produzindo uma ou duas soluções por iteração (GONZÁLEZ; VELA; VARELA, 2015).

Outras características são as direções com que se pode percorrer o espaço de busca, podendo o PR ser unidirecional (ALVAREZ-VALDES *et al.*, 2008; BARBALHO *et al.*, 2013; DÍAZ; FERNÁNDEZ, 2006; DUARTE; MARTÍ; GORTAZAR, 2011; MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013; RODRIGUEZ *et al.*, 2017; WANG *et al.*, 2012) inicial-guia/guia-inicial, ou bidirecional (NGUYEN; PRINS; PRODHON, 2012; PIÑANA *et al.*, 2004; RESENDE *et al.*, 2010), que avalia as soluções geradas nos respectivos sentidos.

O passo de execução em uma meta-heurística onde essa técnica pode ser inserida varia, podendo ser executada entre pares de soluções obtidas pela meta-heurística, entre uma solução da meta-heurística e uma solução elite (BARBALHO *et al.*, 2013; CHAOVALITWONGSE *et al.*, 2011; MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013; NGUYEN; PRINS; PRODHON, 2012; RESENDE *et al.*, 2010), somente entre soluções elite, ou combinações das últimas duas propostas (AIEX; BINATO; RESENDE, 2003; BOUDIA; LOULY; PRINS, 2007; MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013; NASCIMENTO; RESENDE; TOLEDO, 2010; NGUYEN; PRINS; PRODHON, 2012; RESENDE *et al.*, 2010; VILLEGAS *et al.*, 2011), formando o PR evolucionário.

A técnica de PR apresentada anteriormente é considerada como PR interior. Recentemente uma variação desta técnica foi proposta por Rodriguez *et al.*, 2017 de uma maneira que difere do procedimento padrão, o PR exterior. A premissa de escolha de duas soluções sendo tratadas com solução guia e inicial continua, mas os movimentos realizados diferem. Os movimentos realizados pelo PR exterior são as trocas de um atributo por vez para cada atributo igual que se encontre na mesma posição em ambas as soluções guia e inicial. O

processo termina quando nenhum movimento mais possa ser realizado. O objetivo não é de se chegar à outra solução, mas a diversificação.

A técnica de PR evoluiu de tal maneira que, por si só, é uma meta-heurística completa podendo compor as três etapas básicas de um procedimento meta-heurístico: configuração da solução inicial, busca local e pós-otimização. Para este trabalho, tal técnica foi utilizada como procedimento de pós-otimização na tentativa de se encontrar soluções com configurações ainda não visitadas, de forma rápida, no intuito de, sempre, buscar a melhoria da função objetivo da solução final.

O PR, para esta proposta de trabalho, é executado após todo o procedimento de busca local ter terminado, onde se gerou uma matriz com as dez melhores configurações, chamadas de soluções elite, encontradas para o problema de RSDEE em questão. O procedimento de pós-otimização ocorre entre as soluções elite, em somente uma direção (ALVAREZ-VALDES *et al.*, 2008; DÍAZ; FERNÁNDEZ, 2006; DUARTE; MARTÍ; GORTAZAR, 2011; GONZÁLEZ *et al.*, 2015; GONZÁLEZ; VELA; VARELA, 2015; LAGUNA; MARTI; CAMPOS, 1999; MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013; RODRIGUEZ *et al.*, 2017; WANG *et al.*, 2012).

Unindo a direcionalidade da geração das soluções intermediárias, e sobre quais soluções geradas a técnica de PR é executada, algumas estratégias podem ser melhor visualizadas a seguir:

- Entre soluções elite, nas duas direções (NASCIMENTO; RESENDE; TOLEDO, 2010; NGUYEN; PRINS; PRODHON, 2012; PIÑANA *et al.*, 2004);
- Entre soluções obtidas por uma heurística e uma solução elite, e depois entre todas as soluções elite (AIEX; BINATO; RESENDE, 2003; BOUDIA; LOULY; PRINS, 2007; MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013; NASCIMENTO; RESENDE; TOLEDO, 2010; NGUYEN; PRINS; PRODHON, 2012; VILLEGAS *et al.*, 2011);
- Somente entre uma solução obtida por uma heurística e uma solução elite (ARMENTANO; SHIGUEMOTO; LØKKETANGEN, 2011; BARBALHO *et al.*, 2013; CHAOVALITWONGSE *et al.*, 2011; JIA; HU, 2014; MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013; NASCIMENTO; RESENDE; TOLEDO, 2010; NGUYEN; PRINS; PRODHON, 2012);

- Entre duas soluções obtidas por uma heurística, no procedimento de busca local (ARIN; RABADI, 2016; BENAVIDES *et al.*, 2013; PENG; LÜ; CHENG, 2015).

2.2.3.1 Exemplos de Aplicação

Ainda sendo ilustrada pela metáfora do Caixeiro Viajante, a execução do PR no problema é de fácil implementação, também podendo executar movimentos bem simples como a troca de somente um par de elementos da solução, troca de blocos sequenciais de elementos, ou trocas de arestas (MESTRIA; SATORU OCHI; DE LIMA MARTINS, 2013). A distância entre as soluções podem ser a quantidade de elementos ou arestas diferentes entre as soluções inicial/intermediária e guia.

Para exemplificar a forma com que o PR pode ocorrer, para este tipo de problema, após uma busca local 2-opt, realizada em uma solução inicial, para um problema de dez cidades, chegou-se a uma solução ótima local, que será tratada como a solução inicial para o procedimento PR com outra solução, localizada em um conjunto de soluções elite, que será tratada como a solução guia. Na Figura 4, ilustram-se as devidas representações vetoriais e supostas distâncias percorridas.

Figura 4 – Representação de soluções para o procedimento de PR para o problema do Caixeiro Viajante – (a) Solução inicial – (b) Solução guia.

$$\begin{array}{ll}
 S_{ini} = (3,9,1,7,4,2,10,8,6,5) & S_{guia} = (2,8,3,6,5,10,1,4,9,7) \\
 d = 123 & d = 98 \\
 (a) & (b)
 \end{array}$$

Fonte: Próprio autor

O movimento de transição de uma solução intermediária para outra pode ser tratado como a troca de um par de elementos, sendo um dos elementos aquele que deve se encontrar em uma mesma posição entre as soluções intermediária e a guia. A análise para a escolha do elemento a ser analisado pode ser feita por meio de uma análise sequencial comparando-se as mesmas posições em ambas as soluções inicial/intermediária e guia. Na Tabela 1, ilustram-se as quatro primeiras iterações deste procedimento com supostas distâncias percorridas.

Assim, se analisam sequencialmente ambas as soluções S_{ini} e S_g , a qual gera a primeira solução intermediária S_{int_1} . O elemento da posição 1, difere em ambas as soluções: inicial e guia, representado os elementos 3 e 2, respectivamente. Busca-se então, na solução inicial o elemento que se encontra na posição 1 da solução guia, localizado na posição 6.

Tabela 1 - Representação de quatro iterações do procedimento de PR para o problema do Caixeiro Viajante.

Iteração	Soluções Intermediárias	Distância	Solução Guia
1	$S_{int_1} = (2,9,1,7,4,3,10,8,6,5,2)$	$d = 102$	$S_{guia} = (2,8,3,6,5,10,1,4,9,7,2)$ $d = 98$
2	$S_{int_2} = (2,8,1,7,4,3,10,9,6,5,2)$	$d = 96$	
3	$S_{int_3} = (2,8,3,7,4,1,10,9,6,5,2)$	$d = 100$	
4	$S_{int_4} = (2,8,3,6,4,1,10,9,7,5,2)$	$d = 99$	

Fonte: Próprio autor.

Encontrado o elemento realiza-se a troca entre os pares de posições 1 e 6, e 3 e 2, respectivamente, na solução inicial, trazendo o elemento 3 para a mesma posição em que se encontra na solução guia. O processo continua até que se alcance a solução guia.

3 O PROBLEMA DA RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA - RSDEE

O problema de RSDEE (BENAVIDES *et al.*, 2013; FRANCO *et al.*, 2012b; POSSAGNOLO, 2015) consiste em encontrar uma topologia radial através de operações de chaveamentos de abertura e fechamento de circuitos para um dado sistema de distribuição de energia elétrica. A topologia pode ser modificada através da abertura e do fechamento de chaves em certos ramos. O objetivo típico é de se encontrar uma topologia que possibilite obter as menores perdas elétricas do sistema de distribuição avaliado para um dado perfil de carga.

Tipicamente, a operação de um sistema de distribuição é em configuração radial, assim os circuitos que se encontram energizados (chaves fechadas nos devidos ramos) formam uma configuração semelhante a uma árvore geradora, como é conhecida em grafos. Os ramos que estão com a chave aberta (ramos não energizados) são chamados de ramos inativos, e os ramos que estão com a chave fechada são chamados de ramos ativos.

O processo para se encontrar uma solução ótima, tende a analisar implícita ou explicitamente todas as configurações radiais possíveis (força bruta), o que aumenta exponencialmente a quantidade de configurações possíveis de acordo com que se aumenta o número de chaves alocadas no sistema, fazendo com que a complexidade do problema aumente (explosão combinatória).

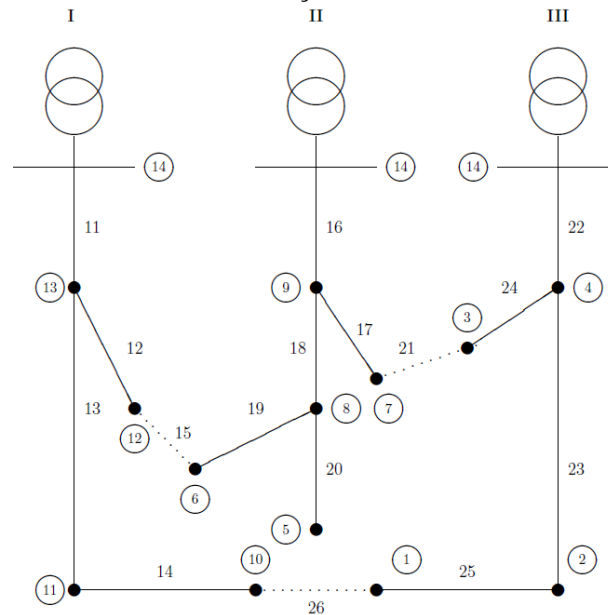
Em problemas como este, deve-se considerar características intrínsecas como confiabilidade da rede, que depende de conjuntos e taxas de falhas; tempos de interrupção de comunicação e consumo de energia de um determinado nó. Esta característica pode ser utilizada como parâmetros para gerar as falhas na rede que se reestrutura, e assim podem-se estudar os impactos da solução obtida. Outras características, que deve ser motivo de análise, são os limites superiores e inferiores de tensão e corrente das linhas e das subestações.

O sistema pode ser representado por um grafo onde se deve encontrar uma árvore que minimize uma função objetivo (minimização de perdas e/ou balanço de carga) que satisfaça as seguintes restrições:

- 1) Limite de tensão em cada barra do sistema;
- 2) Capacidade de carga dos ramos condutores e da subestação;
- 3) Fluxo de potência de acordo com as Leis de Kirchhoff;
- 4) Radialidade.

A restrição de configuração radial, conforme ressaltado anteriormente, é que torna difícil a modelagem do RSDEE, e também a sua natureza combinatória é de difícil tratamento. Na Figura 5 representa-se uma configuração inicial para o problema de RSDEE. Existem 190 árvores ou configurações radiais possíveis, e em um sistema real a explosão combinatória do problema é mais evidente, tornando difícil a solução deste problema por meio de métodos exaustivos, numeração explícita e métodos clássicos.

Figura 5 – Sistema inicial de distribuição de 14 barras com 511 kW de perdas.



Fonte: (CIVANLAR *et al.*, 1988).

Sendo o objetivo do problema minimizar as perdas de elétricas no sistema, dado o grafo da Figura 5, deve-se encontrar a configuração radial para o sistema que minimize essa função objetivo. Para a minimização de perdas a função objetivo assume a forma (3) (POSSAGNOLO, 2015):

$$v = \sum_{i=0}^{n-1} r_i \frac{P_i^2 + Q_i^2}{V_i^2} \quad (3)$$

onde P_i e Q_i representam a injeção de potência ativa e reativa que estão saindo da barra i , V_i indica a magnitude de tensão na barra i , r_i indica a resistência da linha que está saindo da barra i , e v que representa as perdas totais nos alimentadores. O procedimento de fluxo de potência para se encontrar as perdas de um sistema radial encontra-se de forma detalhada no Anexo B.

Verifica-se que uma topologia radial, como na Figura 5, possui a representação de uma árvore, segundo conceitos de grafos. Partindo deste conceito, árvores geradoras mínimas podem ser utilizadas para construir uma topologia radial para este problema. Os algoritmos de PRIM e de Kruskal (CORMEN *et al.*, 2001) trabalham com referências de pesos aos ramos, que podem ser baseados nas perdas, que dependendo da topologia do problema tais valores são diferentes. O algoritmo constrói uma solução de forma rápida, mas a função objetivo da solução construída é subótima para o problema de reconfiguração (POSSAGNOLO, 2015).

A construção de uma solução para o problema de RSDEE por meio do algoritmo de PRIM, por exemplo, sempre produz uma mesma configuração se partir sempre de um mesmo conjunto de pesos. Uma proposta interessante seria gerar perturbações no conjunto de pesos de forma que o conjunto obtivesse pequenas alterações podendo gerar soluções de diferentes configurações.

No caso da utilização do GRASP, para uma mesma topologia, diferentes soluções podem ser encontradas, razão esta que aumenta de acordo com a dimensão do problema. A *RCL* pode ser formada por uma parcela de ramos com os melhores pesos, e destes um é escolhido aleatoriamente.

A geração de uma nova topologia radial, por meio de fechamento e abertura de ramos, consiste em um problema combinatório linear e discreto. Encontrar soluções de qualidade, ou a solução ótima, requer que uma grande quantidade de configurações seja avaliada, o que pode tornar o processo exaustivo e caro computacionalmente. Devido a este fato são utilizadas meta-heurísticas para que produzam novas configurações, mas que não necessariamente todas sejam avaliadas e que o espaço de geração de novas soluções seja reduzido, para que não ocorra uma grande explosão na geração de novas soluções.

3.1 MODELO MATEMÁTICO DO PROBLEMA DE RSDEE

A formulação matemática para o RSDEE com o objetivo de minimizar as perdas de energia, considerando um nível de demanda, configuração radial e sistema simétrico e equilibrado, se dá a seguir (FRANCO *et al.*, 2012b):

$$\min v = \sum_{(ij) \in \Omega_l} (g_{ij} n_{ij} (V_i^2 + V_j^2 - 2V_i V_j \cos \theta_{ij})) \quad (4)$$

sujeito a:

$$P_{S_i} - P_{D_i} + \sum_{j \in \Omega_{b_i}} (n_{ij} P_{ij}) = 0 \quad \forall_i \in \Omega_b \quad (5)$$

$$Q_{S_i} - Q_{D_i} + \sum_{j \in \Omega_{b_i}} (n_{ij} Q_{ij}) = 0 \quad \forall_i \in \Omega_b \quad (6)$$

$$\underline{V} \leq V_i \leq \bar{V} \quad \forall_i \in \Omega_b \quad (7)$$

$$(I_{r_{ij}}^2 + I_{x_{ij}}^2) \leq \bar{I}_{ij}^2 n_{ij} \quad \forall_{(ij)} \in \Omega_l \quad (8)$$

$$\sum_{(ij) \in \Omega_l} n_{ij} = n_b - 1 \quad \forall_{(ij)} \in \Omega_l \quad (9)$$

$$n_{ij} \in \{0,1\} \quad \forall_{(ij)} \in \Omega_l \quad (10)$$

$$P_{ij} = g_{ij} V_i^2 - g_{ij} V_i V_j \cos \theta_{ij} - b_{ij} V_i V_j \sin \theta_{ij} \quad \forall_{(ij)} \in \Omega_l \quad (11)$$

$$Q_{ij} = -b_{ij} V_i^2 + b_{ij} V_i V_j \cos \theta_{ij} - g_{ij} V_i V_j \sin \theta_{ij} \quad \forall_{(ij)} \in \Omega_l \quad (12)$$

$$I_{r_{ij}} = g_{ij} (V_i \cos \theta_i - V_j \cos \theta_j) - b_{ij} (V_i \sin \theta_i - V_j \sin \theta_j) \quad \forall_{(ij)} \in \Omega_l \quad (13)$$

$$I_{x_{ij}} = g_{ij} (V_i \sin \theta_i - V_j \sin \theta_j) + b_{ij} (V_i \cos \theta_i - V_j \cos \theta_j) \quad \forall_{(ij)} \in \Omega_l \quad (14)$$

onde Ω_l representa o conjunto de todos os ramos do sistema (operantes e não-operantes); Ω_{b_i} representa o conjunto de barras conectadas à barra i ; Ω_b representa o conjunto de todas as barras do sistema de distribuição; P_{S_i}, Q_{S_i} indicam a potência ativa e reativa, respectivamente, fornecida pela subestação na barra i ; P_{D_i}, Q_{D_i} indicam, respectivamente, as demandas ativa e reativa da barra i ; P_{ij}, Q_{ij} representam, respectivamente, os fluxos de potência ativa e reativa no ramo ij ; n_b representa o número de barras do sistema de distribuição; g_{ij} é a condutância do ramo ij ; V_i, \underline{V} , e \bar{V} , representam, respectivamente, a tensão na barra i , e os limites mínimo e máximo dos módulos de tensão; θ_{ij} indica a diferença angular entre as tensões das barras i e j ; $I_{r_{ij}}, I_{x_{ij}}$ representam, respectivamente, as componentes reais e imaginárias do fluxo de corrente no ramo ij ; \bar{I}_{ij} indica o fluxo máximo de corrente do ramo ij ; e a variável n_{ij} possui valores binários e representa o estado operacional da chave no ramo ij ; os fluxos de potências ativas e reativas são representados por P_{ij} , e Q_{ij} , juntamente com a componente real e imaginária do fluxo de corrente $I_{r_{ij}}$ e $I_{x_{ij}}$; e b_{ij} representa a susceptância entre as barras ij .

A equação (4) representa a função objetivo que minimiza as perdas de energia no sistema; as equações (5) e (6) representam o balanço das potências ativas e reativas, respectivamente, em cada barra do sistema; as equações (7) e (8) representam, respectivamente, o cumprimento dos limites de módulo de tensão nas barras e dos limites de corrente nos ramos; a restrição (9) é a condição necessária para obter um sistema com

configuração radial; e a restrição (10) representa a natureza binária da variável de decisão n_{ij} (FRANCO *et al.*, 2012b).

O tratamento da modelagem matemática do problema RSDEE (4)-(14) é para somente um nível de demanda. A expansão das equações (4)-(14) para comportar vários níveis de demanda, resulta na equação de minimização de perdas da seguinte maneira:

$$\min v = \sum_{d \in \Omega_d} \sum_{(ij) \in \Omega_l} c_d^{ls} \Delta_d (g_{ij} n_{ij} (V_{i,d}^2 + V_{j,d}^2 - 2V_{i,d} V_{j,d} \cos \theta_{ij,d})) \quad (15)$$

onde c_d^{ls} representa o custo das perdas no nível de demanda d ; Δ_d representa o período de duração do nível de demanda; e assim atualizam-se as equações (5)-(8) e (11)-(14) com os fluxos de potências ativa e reativa, módulo de tensão e as componentes real e imaginária do fluxo de corrente, para cada nível de demanda.

A formulação matemática para o problema de RSDEE (4)-(14) representa um modelo de programação não-linear inteira mista, de difícil solução e com características de explosão combinatória e comportamento multimodal (POSSAGNOLO, 2015).

A proposta apresentada neste trabalho para o cálculo dos fluxos de potência ativa e reativa, juntamente com os valores das correntes e tensões para ambas as componentes real e imaginária se baseiam no procedimento *backward/forward* (RENATO CESPEDES G., 1990; SHIRMOHAMMADI, 1988; SHIRMOHAMMADI; HONG, 1989).

No passo *backward*, calculam-se as correntes, partindo das barras terminais da árvore do sistema radial até a subestação, onde cada barra i possui seu valor de corrente calculado por meio de um somatório de sua contribuição local de corrente e das correntes calculadas nas subárvores correspondentes à barra i . No passo *forward*, calculam-se as tensões, partindo da subestação, calculando a tensão das barras vizinhas da subestação (ligadas com a subestação) e depois nas barras vizinhas dessas barras até atingir as barras terminais. Assim, se for conhecida a tensão do extremo de um ramo mais próxima da subestação, então é possível encontrar a tensão no outro extremo. Os valores de potências das barras são funções não lineares das tensões e, portanto, a solução final pode ser alcançada de forma iterativa por meio de equações recorrentes (AFSARI *et al.*, 2010).

Assim, as correntes nos ramos, tensões nas barras e perdas elétricas podem ser encontradas utilizando as seguintes relações:

$$I_{kr} = \frac{(P_k V_{kr} + Q_k V_{ki})}{(V_{kr}^2 + V_{ki}^2)} \quad (16)$$

$$I_{ki} = \frac{(P_k V_{ki} + Q_k V_{kr})}{(V_{kr}^2 + V_{ki}^2)} \quad (17)$$

$$V_{mr} = V_{kr} - r_{km}I_{kmr} + x_{km}I_{kmi} \quad (18)$$

$$V_{mi} = V_{ki} - r_{km}I_{kmi} + x_{km}I_{kmr} \quad (19)$$

$$P_{kmp} = r_{km}I_{km}^2 \quad (20)$$

$$Q_{kmp} = r_{km}I_{km}^2 \quad (21)$$

sendo P_k e Q_k as potências ativa e reativa da carga na barra k , V_{kr} e V_{ki} as componentes real e imaginária de tensão na barra k cujo módulo é V_k , I_{kr} e I_{ki} , as componentes real e imaginária da corrente de carga na barra k , V_{mr} e V_{mi} , as componentes real e imaginária da tensão na barra k que está ligada com a barra que se encontra mais próxima da subestação e P_{kmp} e Q_{kmp} são as perdas ativa e reativa no ramo (k, m) (FRANCO *et al.*, 2012a). Uma forma detalhada do procedimento do cálculo de fluxo de potência para um sistema radial por meio do procedimento *backward/forward* é apresentada no Anexo B.

3.2 MÉTODOS DE OTIMIZAÇÃO APLICADOS AO PROBLEMA DE RSDEE

O trabalho de Civanlar *et al.* (1988) foi um dos primeiros a buscar melhorias no tratamento do problema de RSDEE. Ele avalia o problema para três subestações e concentra-se na troca dos ramos (*branch exchange*) a fim de obter as menores perdas de carga para cada subestação e sempre se manter a radialidade do sistema. Para problemas de maior dimensão os autores propuseram uma estratégia para reduzir o espaço de busca que calcula a quantidade de carga resultante da transferência de um grupo de elementos entre diferentes subestações.

Em Zhang *et al.* (2007) uma versão melhorada da BT é implementada juntamente com o operador da meta-heurística de AG, a operação de mutação. A partir do momento que uma solução ótima local é atingida as características específicas das informações tabu tornam-se cruciais para que o processo de busca se distancie destas regiões a fim de buscar novos espaços de soluções. Uma estratégia proposta foi o operador de mutação que faz com que o processo de busca se afaste imediatamente para fora de uma região, com certa probabilidade, após uma solução ótima local ser encontrada. Esse processo somente ocorre quando é notado que não está mais ocorrendo uma melhoria na solução corrente. A mutação atua abrindo e fechando pares de chaves pertencentes a laços internos independentes, para se manter a radialidade do sistema. É realizado o estudo de cada par de chaves em um laço interno do sistema com o intuito de buscar o par que tenha mudado seus *status* de chaves que produziram

a máxima redução de perdas de potência. Os aspectos de memória adaptativa não são apresentados pelo autor.

No trabalho de Abdelaziz *et al.* (2010) a BT proposta propõe uma modificação na qual a lista tabu possui tamanho variado conforme o problema, outra característica é o movimento multiplicativo que busca a diversificação do processo de busca. A avaliação da solução, ou seja, o cálculo de seu fluxo de potência e perdas elétricas, só acontece se as características do mecanismo de perturbação na geração de uma nova solução forem aceitas (os critérios tabu e de aspiração). O mecanismo de perturbação utilizado pelos autores possui três tipos diferentes de movimentos. Um deles parte fechando uma chave escolhida aleatoriamente, criando-se um ciclo, e realizando-se uma troca com todas as outras chaves. A cada troca realizada é verificada a radialidade do sistema, e no caso positivo calculam-se as perdas da nova solução. O movimento multiplicativo realiza o mesmo processo, mas partindo de um conjunto de chaves que possuam seus *status* modificados de forma aleatória ou sequencial. O terceiro movimento é uma variação do movimento multiplicativo, mas com uma restrição do valor máximo possível de chaves a serem escolhidas para se realizar a troca de atributos. A lista tabu armazena um movimento que foi aceito, e seu reverso a fim de que este não seja utilizado novamente, a não ser que respeite o critério de aspiração.

Os autores de Ferreira (2010) propõem por utilizar a meta-heurística *Simulated Annealing* para resolver o problema de RSDEE adotando distribuição trifásica assimétrica e não balanceada com geração distribuída. Utiliza-se a característica de memória da BT, armazenando soluções já visitadas de boa qualidade. O sistema é dividido em onde se possa localizar uma falha, sendo identificados os locais de eventos de contingência, isolamento das faltas, identificação de blocos de reconfiguração de candidatos e a identificação das chaves candidatas ns blocos. A vizinhança trabalha com o fechamento e abertura de pares de chaves de ciclos internos para se produzirem soluções com característica radial.

No trabalho de Swarnkar *et al.* (2011) é utilizado o AG onde um indivíduo de melhor aptidão é criado utilizando a técnica de troca de ramos, que ocorre a cada iteração, com estratégias que controlam a diversificação. A recombinação e a mutação ocorrem de forma comum a um AG, mas com uma razão fixa e variável, respectivamente. A mutação ocorre de acordo com a taxa inicial da mutação, e sua taxa final decresce linearmente. Mesmo produzindo resultados expressivos, as quantidades de indivíduos ineficazes nos processos de recombinação e de mutação são muito grandes, o que se tornou necessário, pela escolha dos

autores, a factibictibilização das soluções. As regras utilizadas para tal reduziram muito o tempo de factibilização de uma solução do algoritmo.

Guerra *et al.* (2012) apresentam uma nova meta-heurística, a Nuvem de Partículas ou Enxame de Partículas, onde cada partícula representa uma solução e as interações entre as partículas são a forma de transição entre soluções no espaço de busca. O comportamento de cada partícula baseia-se em sua experiência anterior e na experiência daquelas com que se relaciona. Assemelha-se muito com o AG no âmbito de que se deseja preservar posições que possuem maior aptidão, descartando as de menor aptidão; e difere do AG no quesito dos operadores de evolução, pois ao invés de utilizar técnicas de recombinação/mutação, a técnica ajusta os voos das partículas de acordo com sua própria experiência de voo e de suas companheiras.

A busca ocorre em um grupo de partículas que voam em direção a uma solução na procura por um ótimo local, guiadas por um processo determinístico. Nos voos ocorrem trocas de informações entre as partículas como função objetivo e posições locais, onde são realizados os movimentos de troca de atributos. Os movimentos são baseados por atrações entre partículas para a melhor posição descoberta, pela atração da partícula com sua melhor posição, e por um parâmetro que delimita os movimentos possíveis e realizados.

A forma utilizada para calcular a distância entre duas soluções foi se fechando todas as chaves de uma solução, formando um laço de chaves candidatas a serem abertas. Esse conjunto de chaves é comparado com outro conjunto de outra solução, e um vetor que armazena a distância das soluções é criado com as chaves que aparecem menos vezes em ambas as soluções.

Mendes *et al.* (2013) utilizaram o AG e o Memético com técnicas de paralelização. São avaliados dois tipos de problemas, um que trabalhe somente com chaves, e outro que adiciona os reguladores de tensão. A população é caracterizada por uma árvore ternária que mantém a melhor solução como raiz da árvore e de acordo com que a função objetivo decai, a solução situa-se nas camadas inferiores da árvore. Essa forma de visualização facilita certas tarefas como, por exemplo, a recombinação que ocorre somente entre soluções de nós adjacentes.

A operação de recombinação trabalha com probabilidades para passar características dos pais para os filhos, que dependem dos valores das características dos pais. A mutação ocorre para uma solução escolhida aleatoriamente, e de tal solução uma chave é escolhida de

forma aleatória e trocado seu estado de chave aberta/fechada. Para o problema que adiciona o regulador de tensão a mutação gera uma perturbação, baseada no valor da razão.

Para o algoritmo Memético, a busca local para o problema que trata somente das chaves, o estado operacional de todas as chaves são trocados e para cada estado de chave trocado que melhore a função objetivo o processo é reiniciado, caso contrário a chave retorna a seu estado anterior e o processo continua; para o problema que adiciona o regulador de tensão, após o término do processo de busca local das chaves, ele busca iterativamente aumentar/diminuir a razão em 1% de cada regulador de tensão, e quando detectada uma melhoria na função objetivo o processo é reiniciado, senão a razão retorna ao seu valor anterior e o processo continua. O algoritmo memético obteve os melhores resultados.

Bayat (2013) também apresentou um novo processo heurístico para a solução do problema de RSDEE, o UVDA – *Uniform Voltage Distribution Based Constructive Algorithm*. O algoritmo construtivo parte de uma subárvore e estuda, a cada inserção de um novo elemento para a solução, o fluxo de potência do sistema, até que todos os elementos estejam conectados. Quando ocorrem caminhos que levem a um mesmo elemento, é mantido o caminho de maior magnitude de tensão. Quando um elemento não puder ser alcançado por mais de um caminho, o elemento é adicionado à árvore em construção, e para a subárvore resultante são executadas operações de trocas de ramos buscando possíveis otimizações. No caso de ocorrerem trocas nos estados das chaves o algoritmo de fluxo de potência deve ser executado a troca de ramos continuada. O processo deve ser executado até quando todos os elementos estiverem inseridos na solução.

Como a inserção de novos elementos na subárvore mantém a radialidade do sistema, algumas otimizações no cálculo das perdas foram realizadas, como a utilização de tensões e correntes obtidas do fluxo de potência executado na configuração radial anterior, e cálculo de corrente dos elementos recentemente inseridos assumindo que suas tensões são iguais às dos elementos da configuração radial anterior, desde que os elementos sejam os mesmos. Isso faz com que cálculos redundantes sejam evitados, e simplifica os cálculos das perdas.

Em Torres *et al.* (2013), o método utilizado foi o AG. A construção de uma solução parte lendo as arestas do sistema por janelas de movimento, iniciando-se pelas barras conectadas às subestações, tratando o grafo como não direcionado e armazenando as arestas lidas (i, j) , não aceitando as arestas (j, i) quando lidas. Essa característica sempre produzirá

uma configuração radial para uma solução. A população inicial é encontrada por meio de uma leitura aleatória das arestas.

O conjunto de operadores recombinação e mutação atuam de forma bem inteligente produzindo somente soluções factíveis, o que agiliza o desempenho do algoritmo. A recombinação passa para o descendente as informações similares que ambos os pais possuem, e as informações restantes de ambos os pais são escolhidas aleatoriamente para terminar de construir a solução do descendente. A mutação ocorre inserindo-se uma chave fechada no sistema, gerado um laço que ao ser identificado abre outra chave aleatória do referido laço, mantendo-se a radialidade. A aptidão de cada solução está relacionada com as perdas, o número de subestações e o limite de tensão nas barras.

Tomoiagă *et al.* (2013) utilizaram os conceitos de grafos e os aplicaram em um AG, trabalhando com grafos conectados. A população inicial é encontrada por meio da técnica de troca de ramos, e a programação do algoritmo genético se baseia na técnica de grafos com adjacência de nós para a representação da árvore do sistema. Os métodos de seleção para os operadores são baseados em nichos de cromossomos, estudando os melhores conjuntos de cromossomos para que o movimento seja realizado. A seleção de um elemento depende da razão entre uma função que representa a distância entre dois cromossomos, e o somatório de todas as funções de distância dos cromossomos em um determinado nicho. A recombinação utiliza critérios que buscam a produção de soluções que não produzam ciclos nem que tenham elementos não conectados, levando em consideração o número de ramos, de vértices e de componentes conectados. A mutação ocorre no momento em que a recombinação produz um ciclo, repondo o excesso ou a insuficiência de elementos a fim de se manter o sistema como uma árvore. O último movimento realiza trocas de ramos quando existem elementos não conectados no grafo.

Benavides *et al.* (2013) tratam o problema de RSDEE para o problema de alocação de chaves por meio de três algoritmos construtivos e pós-otimização com o PR. O método possui a característica puramente gulosa que adiciona o melhor elemento à solução em construção, o semi-guloso que produz uma lista de candidatos (GRASP) a partir de todos os candidatos avaliados, e um de construção por amostra que trabalha de forma parecida ao GRASP, mas a lista de candidatos é uma amostra do conjunto.

O procedimento de busca local possui três estratégias e todas são aplicadas à cada método construtivo. O movimento principal é a alocação de uma chave de sua posição para outra. Um dos movimentos busca a primeira solução que obtiver uma melhor função objetivo;

o outro busca a solução que possui a melhor função objetivo do espaço de solução; e o último busca reduzir o espaço de busca utilizando somente uma parcela de chaves a serem avaliadas no sistema. O procedimento é relativamente rápido, mas os cálculos de confiabilidade podem gerar comprometer o desempenho.

O PR ocorre com os mesmos tipos de movimento da busca local, já que as soluções diferem somente entre as diferentes localizações das chaves. Entre a solução inicial/intermediária e a guia avaliam-se todas as localidades das chaves e realiza-se a inserção ou remoção de uma chave quando necessário.

As soluções intermediárias podem vir a ser ineficazes no quesito de quantidade de chaves utilizadas; somente as soluções com a quantidade de chaves ideal, ou menor, podem ser utilizadas como parâmetro de qualidade de solução. Uma ideia para se contornar esse fator é escolher dois pares de arestas entre a solução corrente e a guia que sejam diferentes de forma alternada, ou seja, um par com chave e sem chave e outro par sem chave e com chave para as soluções corrente e guia, respectivamente, e realizar as trocas das chaves entre o par de ramos da solução corrente.

Oliveira *et al.* (2014) adotaram uma abordagem heurística baseada em um conjunto dinâmico de chaves, DSSHA – *Dynamic Switches Set Heuristic Algorithm*: um conjunto de chaves é atualizado a cada troca de ramos (abertura/fechamento de chaves), atualizando o conjunto dinâmico de chaves. A atualização do conjunto de chaves previne a convergência prematura devido a uma chave não promissora no início do processo de reconfiguração poder ser reavaliada durante o processo. A radialidade sempre será mantida não sendo necessária tal avaliação.

A solução inicial é de característica radial e monta o primeiro conjunto dinâmico de chaves abertas. Uma chave então é aleatoriamente escolhida e fechada, formando um laço. Um fluxo de potência convencional não-linear é calculado para se determinar as características necessárias a fim de que seja calculado o índice de sensibilidade da reconfiguração. O referido índice é obtido para todas as chaves fechadas no intuito de se identificar qual chave deverá ser aberta para que se quebre o laço formado. A chave escolhida para ser aberta deverá ser a de menor índice de sensibilidade. O índice de sensibilidade de uma determinada chave é a razão entre o somatório das correntes das chaves do laço correspondente sobre o tempo operacional. O processo se repete até que a solução em construção seja igual à solução anteriormente obtida.

Em De Oliveira *et al.*, 2014b é apresentada uma metodologia que se baseia em uma meta-heurística bio-inspirada Sistema Artificial Imune. O Sistema Artificial Imune opera com padrões de reconhecimento, aprendizado de máquina e otimização. Um importante elemento de um sistema imune inato é uma célula B, a qual produz anticorpos em resposta a agentes patogênicos ou antígenos. A complementaridade entre o receptor e um dado antígeno define a afinidade de ligação entre a respectiva célula B e o agente patogênico. Quando a afinidade é maior que certo limite, a ligação receptor-antígeno é realizada, iniciando o processo de maturação, onde mecanismos de seleção de clone, clonagem e hipermutação somática permitem ao organismo produzir variedades de células B com alta afinidade com o antígeno. A afinidade de um anticorpo é inversamente proporcional ao seu custo de perda de energia. Após os cálculos das afinidades os anticorpos de melhor afinidade são escolhidos para a clonagem, e formam o novo conjunto de anticorpos.

Inicialmente é criado um repertório de anticorpos por meio do procedimento de troca de ramos, baseado na topologia do problema a fim de que soluções infactíveis não pertençam ao conjunto de soluções iniciais. A partir de uma solução factível é escolhida a quantidade de anticorpos que produzem novas soluções iniciais, a partir da primeira, modificando uma quantidade de características da solução (chaves) igual à quantidade de anticorpos. Uma análise do sistema deve ser realizada a cada troca de ramo para definir os ramos que conectam todas as barras e não formam laços.

A hipermutação somática entra no processo para manter a diversidade no conjunto de anticorpos e aumentar a afinidade das soluções candidatas. A aplicação desse mecanismo resulta em uma nova população, e a probabilidade de mutação de um anticorpo é inversamente proporcional à sua afinidade normalizada. Ou seja, as melhores soluções são mantidas e as piores passam pelo processo de hipermutação. Todo o processo de afinidade é realizado para o conjunto das soluções que passaram pelo processo de hipermutação, onde os melhores anticorpos são escolhidos para que substituam os piores anticorpos do conjunto de anticorpos atual, e o processo se reinicia.

A busca em vizinhança variável (VNS – *Variable Neighborhood Search*), que consiste em se trabalhar com diferentes espaços de busca, percorrendo por esses no momento em que não se encontram melhores soluções do que a solução corrente, apresentou resultados expressivos para o problema de RSDEE operando com vários níveis de demanda (POSSAGNOLO, 2015). A busca local visa encontrar os laços internos independentes do sistema elétrico e gera as novas configurações radiais fechando e abrindo ramos de ligações

dentro dos laços. Assim, sempre serão geradas configurações radiais, e conseqüentemente factíveis.

Um método meta-heurístico foi utilizado por (SHOUKOUHI; SHOJAEIAN, 2015) baseado no Algoritmo da Formiga Leão. O algoritmo imita o comportamento das formigas leão se alimentarem de formigas comuns por meio de armadilhas. A modelagem desse comportamento requer que as formigas comuns se movam no espaço de busca e as formigas leão são permitidas caçá-las utilizando as armadilhas. O movimento das formigas comuns foi modelado com caminhos aleatórios, já que na natureza são estocásticos. Quando uma formiga atinge o centro da armadilha, ela é devorada pela formiga leão, e uma nova armadilha se inicia. Isto pode ser caracterizado como uma solução sendo atraída para um espaço de uma solução ótima local/global. A melhor formiga leão obtida em cada iteração é salva mantendo uma lista de soluções elite.

O algoritmo apresenta uma habilidade de busca global de forma rápida e estável, que permite somente movimentos que melhorem a função objetivo da solução (formiga comum), uma vez que a formiga na armadilha se aproxima em direção à formiga leão, a qual pode ser considerada como uma solução ótima local/global atingida.

NGUYEN e TRUONG, 2015 utilizaram uma meta-heurística moderna, o Algoritmo de Busca Cuco, baseado no parasitismo de ninhada de algumas espécies de cuco, que põem seus ovos em ninhos de outros pássaros hospedeiros para resolver problemas de otimização.

Cada cuco põe um ovo por vez (solução designada) e o despeja em um ninho qualquer entre um conjunto finito de ninhos hospedeiros possíveis. Os melhores ninhos com a melhor qualidade de um ovo (melhor solução) serão mantidos para a próxima geração. O número de ninhos possíveis é fixo e o pássaro hospedeiro tem uma probabilidade de reconhecer o ovo diferente. Nesse caso o pássaro hospedeiro pode jogar o ovo diferente fora ou abandonar o ninho, para que possa construir um novo ninho.

A manutenção da radialidade é obtida mantendo-se o mesmo valor entre ramos inativos e chaves abertas. O número de chaves que devem ser abertas após uma reconfiguração é específico e tratado por um algoritmo variável, assim cada membro da solução inicial possui característica radial. Cada configuração radial do sistema é tratada como um ninho hospedeiro. Tem-se então um conjunto de ninhos iniciais, e a cada ninho é associado um conjunto de chaves abertas.

Cada ovo é caracterizado como uma solução que é gerada na inicialização, nos ninhos. Para os ninhos hospedeiros gerados na inicialização é executado o algoritmo de fluxo de potência e calculadas suas perdas. A população inicial dos ninhos hospedeiros é ajustada pelo melhor valor de função objetivo de cada ninho, e o melhor ninho é reservado.

A geração de novas soluções ocorre em todos os ninhos, a não ser no ninho de melhor função objetivo, e substituem-se seus valores de acordo com os novos ovos do cuco que são produzidos pela técnica do voo de Lévy baseada no valor da melhor função objetivo do ninho, a melhor função objetivo de todos os ninhos, e valores estocásticos.

Existe uma probabilidade de se descobrir um ovo diferente no ninho, forçando a geração de uma nova solução, com um processo similar ao de geração de uma solução inicial. O intuito é de se trocar o ovo indesejado por um com melhor função objetivo. Então são calculadas novamente as perdas das soluções, atualizada a lista constituída pelas soluções de melhores funções objetivo, armazenada a solução de melhor função objetivo e reiniciado o processo até perfazer um determinado número de iterações. Os resultados são mais eficazes que a técnica de enxame de partículas e muito próximos dos resultados do AG, mas em todas as simulações a solução ótima foi encontrada.

O trabalho de Li *et al.* (2016) utilizou os conceitos de grafos no algoritmo de Kruskal para se gerar árvores geradoras mínimas como soluções iniciais, sendo que a busca local é regida por duas técnicas elaboradas pelos autores, e um processo de pós-otimização. O procedimento se baseia em atribuir pesos às arestas e ativar as arestas de menor peso uma a uma, sempre se prevenindo a criação de laços, até que todos os elementos estejam conectados. A árvore resultante é considerada como uma árvore geradora mínima do sistema, um processo muito similar aos algoritmos de PRIM e Kruskal. O peso de cada aresta é calculado por uma razão entre as tensões mínima e máxima das barras pertencentes ao ramo.

Duas técnicas foram implementadas para aumentar a eficiência do procedimento, reduzindo seu tempo computacional, assim como a quantidade de chaves candidatas a modificarem suas condições (aberta e fechada) para a geração de uma nova configuração. Uma delas é calcular as quedas de tensão relativa entre duas barras em um ramo de chave aberta antes da transferência de carga, assim criou-se um limite de tensão que delimita quando uma chave é mais propensa a trocar sua característica. A outra técnica é de se reduzir as perdas somente se a carga é transferida de uma barra com baixa tensão para uma barra com alta tensão.

O algoritmo é composto por três procedimentos. O primeiro constrói uma solução por meio do conceito de árvore geradora mínima. É calculado o fluxo de potência para o sistema com todas as chaves fechadas e, então, calculam-se os pesos dos ramos. Executa-se o algoritmo de Kruskal e constrói-se o conjunto de chaves abertas verificando-se a solução construída, que no caso de ser factível é transferida para o próximo procedimento; caso contrário, deve-se restabelecer o peso do ramo mais leve.

O segundo procedimento parte da solução construída e aplica as regras de melhoria heurísticas dos cálculos de carga transferidas quando se trocam os atributos entre a chave aberta com maior diferença de tensão e a chave fechada de menor diferença de tensão. Calculam-se a queda de tensão e de carga da chave recentemente aberta, e se caso a queda de tensão for maior que um limite estabelecido esta chave é escolhida para trocar seu estado novamente com outra adjacente; se não se verifica o conjunto de chaves candidatas.

Quando o conjunto de chaves candidatas for vazio o último procedimento é executado meio do cálculo do fluxo de potência da solução corrente, começando por um processo de abertura e fechamento de chaves adjacentes na busca por redução da função objetivo, até que todas as chaves que estavam fechadas na solução do início do procedimento tenham trocado uma vez seus atributos.

Além de se produzir uma solução de boa qualidade para o estágio inicial com o método de árvore geradora mínima, as regras utilizadas para as possíveis trocas de atributos entre as chaves do sistema aumentam muito o desempenho do algoritmo permitindo-o encontrar soluções ótimas com grande eficiência, pois reduzem o espaço de busca por soluções e são guiadas a produzirem soluções melhores.

Bayat *et al.* (2016) expandiram a aplicação de sua moderna meta-heurística UVDA (BAYAT, 2013) não somente para o problema de RSDEE, mas também para o problema de localização e dimensionamento de geradores distribuídos. O UVDA é utilizado para se gerar uma solução de alta qualidade. A partir de uma solução inicial o problema de localização ótima de geradores distribuídos começa selecionando a barra com a menor magnitude de tensão como o candidato inicial para a instalação de um gerador distribuído. Em seguida aplicam-se mudanças na topologia do sistema adotando a barra selecionada como fonte (gerador/subestação). A partir da configuração resultante deve ser novamente executado o algoritmo UVDA e calculadas as perdas. No caso das perdas da solução forem menores que as perdas da solução gerada pelo UVDA no início do processo, devem-se atualizar a melhor

solução e realizar o mesmo procedimento para outra barra adjacente até que todas as barras tenham sido submetidas ao processo.

Para a admissão de múltiplos geradores, barras com valores mínimos de tensão são caracterizadas como nós raízes virtuais. Tal configuração passa novamente pelo algoritmo UVDA, as perdas são calculadas e comparações são realizadas para verificar se a função objetivo desta nova configuração obteve melhoria. O dimensionamento de cada gerador fica a cargo de uma avaliação de incremento de carga da configuração resultante. Os testes foram realizados em vários cenários e a técnica se mostrou mais eficiente do que algoritmos como AG, algoritmo genético refinado e algoritmo de busca harmônica.

Masteri e Venkatesh (2016) propõem um algoritmo de reconfiguração que utiliza técnicas clássicas de otimização não linear, com complementaridade, que garante uma solução ótima, em tempo real. Por meio da complementaridade, o problema de otimização não linear inteira mista de reconfiguração ótima com espaço de solução descontínuo é transformado em um problema de otimização não linear com um espaço de solução contínuo, tornando possível a utilização de técnicas de otimização não linear. As características de radialidade são inseridas no cálculo da função objetivo. A função objetivo construída é formada por uma minimização de três termos. O primeiro são as perdas em tempo real do sistema; o segundo garante que o produto dos estados das chaves seja igual a 0 (fechada) em cada laço onde somente uma chave por laço possui valor 1 (aberta); e o terceiro garante que somente estados não nulos (chave aberta) de uma chave em um laço seja igual a 1. Assim, uma solução ótima global é atingida quando o segundo e o terceiro termo têm valores iguais a zero.

Os testes foram realizados para sistemas de pequeno e de médio porte e comparados a técnicas como AG, métodos *fuzzy* multi-objetivo e otimização por enxame de partículas. Os resultados da metodologia proposto alcançaram as soluções reportadas na literatura especializada apenas com uma execução do algoritmo, mas os tempos de execução aumentaram de acordo com o tamanho do problema, mesmo quando comparado aos tempos de execução das outras técnicas avaliadas, que executaram de 8 a 200 vezes.

No trabalho de Abdelaziz (2017) é adotada uma estratégia de aumentar ou de se reduzir o tamanho de uma população, de forma adaptativa, de acordo com a situação da busca local, mostrando que a eficiência do algoritmo pode ter melhoria. Assim como um AG genético clássico o procedimento tem início com um determinado número de indivíduos para formar a população inicial. Estes indivíduos são submetidos à recombinação, a fim de que novas soluções sejam geradas, e então certas soluções são submetidas ao processo de

mutação, e uma nova população é definida para a próxima iteração. A adaptação do tamanho da população, crescendo e decrescendo, adotada neste trabalho, ocorre de acordo com a permanência de indivíduos de uma população para a outra de acordo com sua idade, por exemplo, o número de gerações/populações que tal indivíduo participou.

A partir da escolha da população inicial, uma parcela das soluções de melhor função objetivo é escolhida como soluções elite. Quando a quantidade da população for maior que a quantidade de soluções elite, os subsequentes indivíduos na população são escolhidos para serem submetidas ao procedimento de recombinação. Os indivíduos restantes, então são escolhidos para sofrer mutação. Para cada nova solução gerada são calculadas suas perdas e factibilidade, sendo descartadas as soluções infactíveis. A manutenção da permanência ou não de soluções entre iterações do algoritmo é realizada por uma função que determina o tempo de vida incorporado a cada solução, que aumenta a cada iteração do algoritmo. Quando uma solução atinge seu limite no tempo de vida, ela é descartada da população.

Em Kumar *et al.* (2018) propõe-se uma estrutura de planejamento coordenada para se resolver o problema de instalação e distribuição de geradores para se minimizar as perdas de um sistema. A meta-heurística de enxame de partícula é utilizada acompanhada de um operador baseado em troca de *bits* incorporado na velocidade da equação da meta-heurística. O algoritmo de enxame de partículas ocorre de forma comum, como em Guerra *et al.* (2012) onde uma população de partículas é gerada, cada partícula é uma possível solução e possuem parâmetros, como a posição e a velocidade, escolhidas aleatoriamente. A velocidade atua na atualização da posição de uma partícula, e a cada iteração ambos os parâmetros são atualizados de acordo com a melhor posição da partícula e a melhor posição de todas as partículas. Pode ser entendido que uma população possa ser caracterizada por um conjunto de topologias radiais e cada topologia radial é representada pelas chaves abertas. O operador de troca de *bits* se baseia na troca dos valores das chaves, representados por 1 e 0 no vetor de uma solução. O elemento que mudará de posição no vetor de solução é escolhido de acordo com a melhor posição da partícula e a partícula com a melhor posição, que também controla a atração de uma partícula para sua melhor posição e para a melhor entre todas as posições.

No espectro de pesquisa dos trabalhos analisados neste capítulo verifica-se que a maioria destes trabalhos utiliza algoritmos evolutivos e modernas meta-heurísticas, e poucos que trabalham com métodos heurísticos construtivos ágeis, assim como o GRASP, para soluções iniciais; poucos que trabalham a meta-heurística BT, podendo ser devido às suas características e parâmetros; e menos pesquisadores que propõem métodos de pós-otimização

como o PR. Isso traz novos caminhos para serem explorados na solução específica de problemas de RSDEE na intenção de se agregar estas técnicas, assim como é proposto neste trabalho.

3.3 CÁLCULO DE FLUXO DE POTÊNCIA INCORPORADO AOS MÉTODOS DE OTIMIZAÇÃO

O cálculo de fluxo de potência em Sistemas de Distribuição de Energia Elétrica (SDEE) é uma ferramenta computacional para engenheiros de sistemas de potência, pois é necessário para a análise operacional e para o planejamento de longo prazo de SDEE (LISBOA *et al.*, 2014). O cálculo de fluxo de potência tem como objetivo principal conhecer as magnitudes de tensão e ângulo nas barras e, conseqüentemente, o fluxo de potência ativa e reativa nas linhas (DUKPA; VENKATESH; EL-HAWARY, 2009; KARAMI; MOHAMMADI, 2008). Na década de 1950, muitas técnicas para o cálculo de fluxo de potência foram desenvolvidas, como a de Gauss-Seidel direta e indireta, Newton-Raphson e suas versões desacopladas para sistemas malhados (MOK *et al.*, 2010) e, posteriormente, métodos iterativos para sistemas radiais (RENATO CESPEDES G., 1990; SHIRMOHAMMADI, 1988).

Com o crescimento da complexidade dos sistemas de potência, métodos computacionais de análise de fluxo de potência mais sofisticados foram pesquisados para melhorar o planejamento, a operação e o controle desses sistemas. Em geral, essa tarefa consome muito tempo computacional devido ao conjunto de equações algébricas não-lineares que são resolvidas por técnicas numéricas iterativas, onde a aplicação de métodos numéricos não são os mais adequados para aplicações computacionais (KARAMI; MOHAMMADI, 2008).

Na literatura, os métodos mais comumente usados para o cálculo do fluxo de potência em SDEE radiais podem ser classificados como:

- Método da Compensação (MC);
- Método de Newton-Raphson (MNR);
- Redes Neurais Artificiais (RNAs);
- Método *Backward/Forward* (MBF) e suas variações.

O MC (SHIRMOHAMMADI, 1988; SHIRMOHAMMADI; HONG, 1989) aplica um método heurístico de abertura de chaves nos ramos para a obtenção de um sistema elétrico radial e, por meio das formulações básicas das Leis de Kirchhoff, calcula o fluxo de potência para sistemas fracamente ou fortemente malhados. A ideia principal é utilizar injeção de corrente, calculada pelo MC, nas barras terminais que representam a quebra de um ciclo. A técnica utiliza o procedimento de renumeração de barras e, nos sistemas elétricos tratados, no máximo 25% dos ramos possuem chaves, o que reduz a complexidade da análise.

O MNR (DUKPA; VENKATESH; EL-HAWARY, 2009; LI *et al.*, 2016a) é uma técnica numérica utilizada para encontrar soluções de um sistema não-linear de equações algébricas, o que justifica sua aplicação para o problema de cálculo de fluxo de potência. Em DUKPA; VENKATESH; EL-HAWARY, 2009, resolve-se um sistema de equações de primeira ordem pelo MNR que se estende para o Método da Continuação pela adição de um fator de carga à matriz de potências. Com a matriz de potências construída, realiza-se o cálculo de fluxo de potência. Essa técnica tem como um de seus objetivos o cálculo do ponto máximo de carga (ponto de colapso de tensão). Em Li *et al.*, 2016a, constrói-se uma matriz de incidência que informa se cada nó é a origem ou o final de um ramo em um sistema radial. Dessa matriz de incidência, derivam-se as matrizes de ramos e de conexões. Das matrizes resultantes, o cálculo do fluxo de potência foi proposto de duas maneiras: derivado das perdas de tensão dos ciclos e das diferenças do vetor de ângulo, baseado nas Leis de Kirchhoff, alcançando-se a resistência e reatância de forma matricial, com as potências ativas e reativas; e por meio da simplificação da matriz dos coeficientes de resistência e reatância, no caso de o sistema possuir uma boa condição de tensão, onde as amplitudes de tensão podem ser adquiridas de forma aproximada ao valor nominal

As RNAs utilizadas para o cálculo de fluxo de potência permitem a obtenção de resultados de forma praticamente instantânea. A RNA necessita ser treinada em um conjunto de casos limitados que componham as condições de operação para o sistema teste. Uma vez completo o treinamento para os ajustes das variáveis das funções de base radial (FBR), todos os fluxos de potência podem ser calculados rapidamente (KARAMI; MOHAMMADI, 2008). As FBRs são utilizadas para aproximar as funções de condição de operação do sistema, as quais representam as saídas da RNA, ou seja, os valores do cálculo do fluxo de potência; e as entradas da RNA são conjuntos de operação do sistema, cuidadosamente escolhidos, que possam simular o procedimento de cálculo de um fluxo de potência computacionalmente.

O MBF é um procedimento iterativo para o cálculo de fluxo de potência em dois passos, inicialmente proposto em Cheng; Shirmohammadi, 1995). No passo *backward*, calculam-se as correntes, partindo das barras terminais da árvore do sistema radial até a subestação, onde cada barra i possui seu valor de corrente calculado por meio de um somatório de sua contribuição local de corrente e das correntes calculadas nas subárvores correspondentes à barra i . No passo *forward*, calculam-se as tensões, partindo da subestação até as barras terminais por meio de uma distribuição plana de tensão do valor calculado de tensão em cada barra i para as barras que possuam um ramo ativo com a mesma. Os valores de potências das barras são funções não lineares das tensões e, portanto, a solução final pode ser alcançada de forma iterativa por meio de equações recorrentes (AFSARI *et al.*, 2010).

Em Li; Broadwater, 2003, o MBF é utilizado para calcular o fluxo de potência dividindo o sistema elétrico radial em subárvores, onde cada uma é manipulada por vez por meio de um algoritmo distribuído. Esse tratamento é realizado de forma sequencial em Mok *et al.*, 2010, o sistema também é dividido em subárvores.

Em Lisboa *et al.*, 2014, uma matriz de incidência é construída a partir do sistema de equações não lineares e o fluxo de potência é calculado sem a necessidade de inversão de matrizes com tempo de processamento linear.

Em Shakarami *et al.*, 2017, os ramos do sistema são representados por parâmetros das matrizes de impedância e de admitância. Por meio da aplicação da Teoria de Grafos, são criados conjuntos de caminhos entre os pares de vértices, os quais utilizam o caminho quando o conjunto de arestas e vértices são distintos, juntamente com os parâmetros das matrizes de impedância e de admitância para o cálculo do fluxo de potência. Esse procedimento faz com que métodos de otimização, como fatoração ou matrizes jacobianas, não necessitem ser utilizados. O procedimento de renumeração de barras foi adotado em todos esses trabalhos baseados no MBF.

A utilização do MBF sem a necessidade do procedimento de renumeração de barras demandou a inversão e a transformação de matrizes. Em Farag; Shatshat; Zidan, 2011, o cálculo do fluxo de potência depende principalmente da matriz de incidência de barras, que é a relação entre a injeção de correntes nas barras e de correntes nos ramos, e que deve ser invertida. Em Singh; Ghose, 2013, é proposta uma técnica similar que faz a construção e a transformação de matrizes modificadas de incidência de corrente e de potência. Em Gatak; Mukherjee, 2017, se vê uma nova construção de matrizes de impedância e de carga a fim de calcular as tensões nas barras de forma direta.

A dispensa do procedimento de renumeração de barras produz um aumento de complexidade para o cálculo do fluxo de potência, no mínimo quadrática, diferente da linearidade do procedimento clássico de renumeração de barras.

Finalmente, ambos os procedimentos *backward/forward* não precisam estar presentes em conjunto. Existem variações do MBF onde somente um dos procedimentos é utilizado. Em Afsari *et al.*, 2010; Augugliaro *et al.*, 2010 é utilizado somente o procedimento *backward* e os cálculos correspondentes ao procedimento *forward* são construídos por meio de funções de transformação ou aproximações. Em Hamouda; Zehar, 2011, é utilizado somente o procedimento *forward* e o cálculo correspondente ao procedimento *backward* é realizado em uma matriz que possui as informações dos ramos e de seus pares de nós, calculando as potências ativas e reativas.

A revisão da literatura foi realizada com o objetivo principal de investigar técnicas mais aprimoradas para o tratamento do problema de RSDEE e técnicas mais rápidas e eficientes para o cálculo de fluxo de potência de forma rápida e eficiente. A eficiência é necessária principalmente em problemas que buscam e avaliam grandes quantidades de soluções, como reconfiguração e restauração de SDEE (LISBOA *et al.*, 2014). Técnicas mais rápidas para o cálculo do fluxo de potência nem sempre são aquelas que apresentam a menor complexidade computacional, e as mais eficientes nem sempre são aquelas que negligenciam certos termos dos equacionamentos ou que utilizam aproximações ou transformações. Tratamentos mais aprimorados fazem com que uma estratégia de implementação de um algoritmo, que venha a possuir uma complexidade computacional maior, seja mais rápida, e que possa abordar todos os elementos das equações correspondentes, sendo também mais eficientes.

4 IMPLEMENTAÇÃO E METODOLOGIA ADOTADA

Neste capítulo se descreverá sobre as técnicas de programação utilizadas na construção do programa para o tratamento do problema de RSDEE, como os dados do sistema foram armazenados nas estruturas de dados escolhidas e organizados, as características do problema de RSDEE foram abordadas e como as particularidades das meta-heurísticas foram implementadas, explicando de forma detalhada as funcionalidades (funções/procedimento) do algoritmo.

Um programa de computador é constituído basicamente de dois procedimentos, a construção do algoritmo, e a utilização de estrutura de dados. Um algoritmo é um processo sistemático para computar um resultado a partir de dados de entrada podendo ser representado em pseudo-códigos e/ou fluxogramas. Já as estruturas de dados são instrumentos que organizam os dados e operam sobre eles (vetores, pilhas, filas, deque, árvores, tabelas *hash*, listas). Um programa é a união desses procedimentos perfazendo uma expressão em linguagem formal (inteligível por um computador) de um algoritmo (CORMEN *et al.*, 2001). A complexidade de um programa depende da eficiência de ambos os procedimentos descritos anteriormente.

A eficiência de um algoritmo, ou sua complexidade de tempo, calcula quanto “tempo” é necessário para se computar o resultado para uma instância do problema de tamanho n . As complexidades de pior caso consideram a instância que faz com que o algoritmo funcione de forma mais lenta, como querer se ordenar de forma crescente um vetor de números inteiros e distintos que está organizado de forma decrescente, onde todos os elementos terão que trocar de posição.

O caso médio pode ser calculado como uma média de todos os tempos possíveis de todas as instâncias. A eficiência de uma estrutura de dados é sua complexidade de espaço, ou seja, quanto “espaço de armazenamento” é preciso para armazenar a estrutura (pior caso e caso médio) (CORMEN *et al.*, 2001).

4.1 ESTRUTURAS DE DADOS ADOTADAS

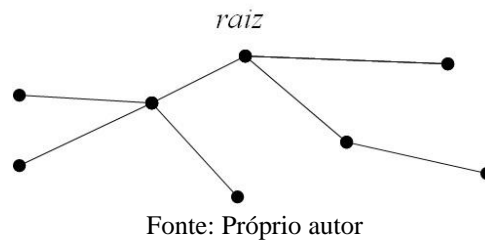
O problema de RSDEE foi implementado, em todos os trabalhos avaliados na literatura, de forma vetorial unidimensional e estática, ou em matrizes bidimensionais estáticas, mesmo sendo a configuração do sistema radial caracterizada como uma árvore.

Uma leitura dos dados em um vetor unidimensional possui complexidade $O(n)$ o que significa que podem percorrer todo o vetor, perfazendo n iterações, obtendo um tempo de execução proporcional à entrada n . Uma matriz bidimensional é representada como vetores de vetores perfazendo uma complexidade de $O(n^2)$, para o caso de um procedimento de leitura linear de seus dados. Pode-se verificar que aumentando a dimensão do problema (por exemplo, a quantidade de barras/chaves) dependendo da complexidade do algoritmo o tempo de execução cresce de forma exponencial. Algumas funções de complexidade estão informadas a seguir (CORMEN *et al.*, 2001; DROZDEK, 2012):

- $T(n) \in O(1)$: constante, mais rápido impossível, como a inserção e remoção de um dado em uma pilha;
- $T(n) \in O(\log \log n)$: super-rápido;
- $T(n) \in O(\log n)$: logarítmico, bastante eficaz, como se fosse buscar um elemento em uma árvore, um algoritmo com a característica de divisão e conquista;
- $T(n) \in O(n)$: linear, é considerado o mais eficiente que se pode esperar se algo não pode ser determinado sem examinar toda a entrada, como exibir todos os elementos de um vetor unidimensional;
- $T(n) \in O(n \log n)$: limite de muitos problemas práticos, o algoritmo de ordenação Merge Sort, por exemplo;
- $T(n) \in O(n^2)$: quadrático, como uma leitura de uma matriz bidimensional;
- $T(n) \in O(n^k)$: polinomial, aceitável para k pequeno;
- $T(n) \in O(k^n), O(n!), O(n^n)$: exponencial, que deve ser evitado.

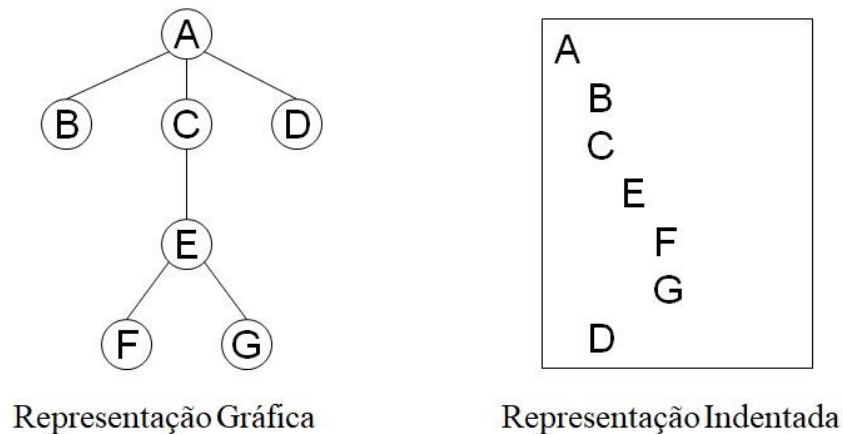
Cada estrutura de dados possui suas especificidades e complexidades diferentes para suas operações, sendo elas inserção, remoção, busca, ordenação, entre outras, podendo se ajustar aos mais diversos tipos de aplicações. Uma delas em particular é a árvore, representada com um grafo acíclico e conexo onde um dos vértices, o nó raiz, é diferenciado dos demais. Na Figura 6, ilustra-se o grafo de uma árvore.

Figura 6 – Ilustração do grafo de uma árvore.



Na Figura 7 pode se identificar que as árvores também podem ser utilizadas para representar hierarquias, e possuem diferentes tipos de representação.

Figura 7 – Ilustração do grafo de uma árvore.



(A(B)(C(E(F)(G)))(D))

Representação com Parênteses

Fonte: Próprio autor

Ainda, pela Figura 7, a “Representação Gráfica” é o tipo de representação que mais se utiliza quando se trata deste tipo de estrutura. Aqui são conhecidas algumas nomenclaturas e características como descritas a seguir, segundo conceitos computacionais e de grafos:

- “A” é o pai de “B”, “C” e “D”;
- “B”, “C” e “D” são filhos de “A”;
- “A” é um ancestral de “G”;
- “G” é um descendente de “A”;
- “B”, “D”, “F” e “G” são nós folha;

- “A”, “C” e “E” são nós internos;
- “A” é a raiz da árvore;
- O grau do nó “A” é 3;
- O comprimento do caminho entre “C” e “G” é 2;
- O nível de “A” é 1 e o de “G” é 4;
- A altura da árvore é 4.

Assim, o comprimento é a distância da quantidade de arestas no caminho entre dois nós; grau é a quantidade de elementos que um dado nó está conectado; nível de um nó é a camada que ele pertence; altura é o total de camadas de uma árvore.

As árvores são as estruturas de dados mais utilizadas na computação, são comumente implementadas seguindo o paradigma de programação dinâmica de memória que aumenta ou diminui o tamanho da estrutura de acordo com a necessidade de se armazenar ou remover informações.

No paradigma de programação estática um vetor unidimensional, quando criado, ocupa um espaço contínuo e contíguo da memória do computador, em outras palavras, ele ocupa um bloco de memória do tamanho que foi declarado, totalmente sequencial. Na programação dinâmica somente um elemento é armazenado na memória do computador, em uma posição “aleatória”, quando chamada sua diretiva de criação. Assim os elementos tendem a utilizar espaços de memória dispersos não mantendo uma sequência na hora de acessar os dados armazenados. Para isso, cada elemento possui, anexado, um ou mais ponteiros que armazenam uma ou mais posições de memória que indicam as posições de memória no computador de outros elementos ligados a ele, elaborando dessa forma uma maneira de percorrer por meio da estrutura de dados elemento a elemento.

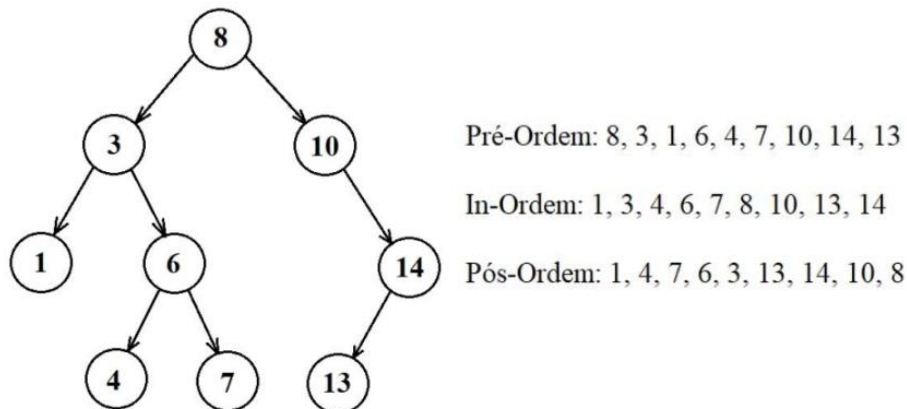
Existem diversos tipos de estruturas de dados baseadas em árvores: árvores binárias, AVL, ordenadas, 2-3, 2-3-4, branco-e-preto, espalhamento, B, B* entre outras, onde cada uma possui características diferentes como quantidade de elementos conectados (filhos), quantidades de elementos em cada posição (nó) e políticas próprias de inserção, remoção e busca (CORMEN *et al.*, 2001; DROZDEK, 2012).

Segundo a descrição do problema de RSDEE a característica fundamental do mesmo é a manutenção da radialidade de suas soluções. A configuração de uma solução para um problema de RSDEE possui característica acíclica e conexa, perfazendo as características de uma árvore. Dessa forma foi proposta a estratégia de implementar a forma com que uma estrutura radial se apresenta por meio da estrutura de dados de árvores.

Uma das razões desta escolha pode ser explicada pelas diferentes formas de percurso que podem ocorrer numa estrutura de dados deste tipo. Existem três procedimentos clássicos sobre a forma de se percorrer os nós de uma árvore, sendo eles os procedimentos: Pré-Ordem, In-Ordem e Pós-Ordem. Uma explicação resumida da maneira com que cada forma de percurso se baseia, é apresentada a seguir, e um exemplo é ilustrado pela Figura 8 com uma árvore binária:

- Pré-Ordem: raiz, esquerda, direita;
- In-Ordem: esquerda, raiz, direita;
- Pós-Ordem: esquerda, direita, raiz.

Figura 8 – Formas de percurso em uma árvore binária.



Fonte: Próprio autor

Cada método de percurso resulta em uma forma diferente do acesso à informação da árvore, resultando em uma diferente ordenação final de seus elementos. Para o caso de uma árvore binária como apresentada na Figura 8 o procedimento Pré-Ordem acessa a informação de um nó a primeira vez que o visita; o In-Ordem acessa a informação quando visita um nó pela segunda vez; e o Pós-Ordem quando o visita pela terceira vez.

Pelo próprio grafo da Figura 8 pode ser verificada uma estrutura radial, e pode se considerar que representa um sistema de distribuição de energia elétrica se forem tomados os nós como sendo as representações das barras, as ligações entre os nós caracterizando os ramos com chave fechada e o nó “8” como a subestação.

O procedimento utilizado neste trabalho para o cálculo de fluxo de potência trabalha com dois processos, *backward/forward* que calculam as correntes do sistema a partir das barras terminais até a subestação e calcula as perdas elétricas (processo *backward*). No caso

da diferença entre dois cálculos de perdas elétricas consecutivas não ser menor do que um limite adotado, calculam-se as novas tensões partindo da subestação até as barras terminais (processo *forward*) e novamente as correntes das barras terminais até a subestação, até que o limite de convergência seja alcançado (processo *backward*).

Como supracitado, tal passo é realizado em uma estrutura estática necessitando-se de vários acessos ao vetor até que todas as informações necessárias ao algoritmo de fluxo de potência sejam encontradas para cada procedimento *backward/forward*. A leitura de uma estrutura estática possui complexidade $O(n)$, pois, percorre todos os elementos de um vetor unidimensional. A leitura de todos os dados de uma árvore possui a mesma complexidade, mas a única diferença é a quantidade de vezes que uma leitura é realizada, para o procedimento *backward/forward*.

O processo de cálculo das correntes no sistema pode ser entendido como o método de percurso Pós-Ordem de uma árvore, onde ele acessa todos os filhos de uma árvore e leva a informação resultante ao pai, e assim, sucessivamente, uma vez que o valor total da corrente de um dado nó é o somatório de corrente de seus filhos. Entretanto, o processo de cálculo das tensões pode ser tratado com o procedimento de percurso em Pré-Ordem, uma vez que a informação das tensões em cada nó é atualizada conforme as impedâncias e correntes, e então replicadas a seus filhos.

Utilizando os procedimentos de percurso na estrutura de dados de árvores verifica-se que o fluxo de potência só percorre a estrutura uma única vez, para cada procedimento *backward/forward* do fluxo de potência, pelo princípio da recursividade. As funções utilizadas para se implementar o fluxo de potência e para se representar uma configuração radial em árvores terão suas funcionalidades explicadas mais detalhadamente nas seções a seguir.

Outra estrutura utilizada são as “estruturas”, ou *structs*, ou registros, que representam coleções de informações. A habilidade de trabalhar com diversos indivíduos e organizá-los em unidades coerentes é uma das características fundamentais da programação moderna. Funções e procedimentos podem manipular os dados de forma independentes ou não, por meio de um único nome (ROBERTS, 1995).

Explicando com outras palavras, ao invés de se operar uma grande quantidade de informações que possuem certas conexões, a forma mais fácil seria de se agrupar as informações por conexão e tratar cada conjunto de informação como um bloco único, como planilhas de informações sobre trabalhadores em uma empresa.

Ao contrário de se trabalhar com um vetor de tamanho considerável para se armazenar as informações de diversos trabalhadores, se construiria um vetor de registros onde cada elemento do vetor seria um registro que conteria todas as informações sobre cada trabalhador.

Basicamente a criação de um registro possui dois procedimentos:

1. Definir uma estrutura ou novo tipo de estrutura;
2. Declarar as variáveis referentes à estrutura.

Um exemplo de sintaxe pode ser visualizado na Figura 9 por meio de pseudo-código (Figura 9(a)) e como se apresentaria na linguagem de programação estruturada C (Figura 9(b)). O acesso à informação do registro é realizado pelo nome da estrutura, que no caso da Figura 9 é “Pessoa”. Assim a informação da idade, por exemplo, é acessada pela expressão “Pessoa.idade”. Tal forma de expressão indica qual variável está sendo acessada e de que registro, bem como vetores de vetores podem ser implementados, registros de registros também o podem, aumentando a complexidade de referenciá-los, tomando cuidado com a forma da expressão de acesso a cada informação.

Figura 9 – Sintaxe e representação de um registro.

<pre>tipo-de-definição estrutura{ Declaração de variáveis }nome-do-tipo-de-dado</pre> <p>(a)</p>	<pre>typedef struct{ int idade, rg; char nome[30]; }Pessoa;</pre> <p>(b)</p>
--	--

Fonte: Próprio autor

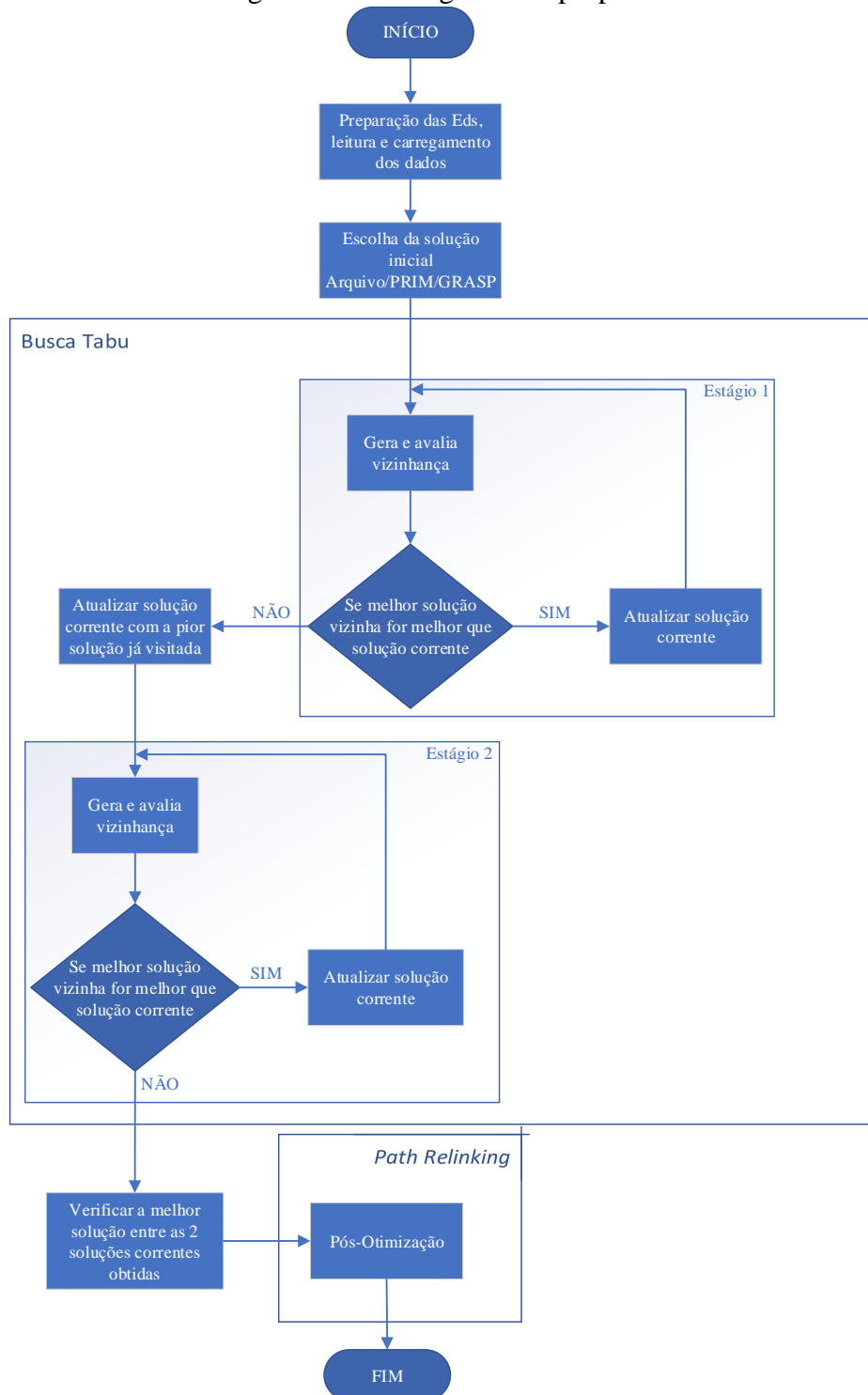
A estratégia foi adotada para este trabalho devido às informações dos nós, e das tensões e potências, possuírem particularidades que permitiram a formação de blocos coerentes de informações. Será explicado de forma mais detalhada, nas seções a seguir, as razões que levaram a utilizar esta estratégia de implementação e como os dados são formatados e organizados.

4.2 FLUXOGRAMA DA PROPOSTA

Antes de explicar de forma mais detalhada as formatações dos dados para as estruturas de dados utilizadas neste trabalho, e como os dados são trabalhados, é apresentada aqui a representação do fluxograma da estratégia implementada para o tratamento do problema de RSDEE por meio do conjunto das três meta-heurísticas descritas neste trabalho anteriormente.

O fluxograma da proposta implementada apresentado na Figura 10 ilustra, por meio de uma visão macro, a ordem em que os procedimentos implementados ocorrem. Nas seções posteriores serão apresentadas as funcionalidades mais pertinentes dos procedimentos implementados, bem como seus devidos algoritmos, dando uma explicação mais detalhada.

Figura 10 – Fluxograma da proposta.



Fonte: Próprio autor

Inicialmente é realizada a preparação das estruturas de dados para receber as informações provindas dos arquivos dos problemas de RSDEE, e declaradas as funções utilizadas no algoritmo. Em seguida são acessados os arquivos do problema de RSDEE a serem avaliados, e suas devidas informações carregadas nas estruturas de dados construídas previamente.

Carregadas as informações dos arquivos necessários, já se possui as principais informações necessárias, como o nó, ou barra, que representa a subestação, e o vetor estático que possui todas as informações do sistema. Constrói-se, então, uma árvore dinâmica onde o ponteiro que indica seu primeiro elemento (raiz da árvore) recebe como informação o nó que representa a subestação. Durante o procedimento da proposta implementada, essa árvore será preenchida, elemento a elemento, até formar uma configuração radial. O único nó que mantém sua configuração/posição inalterada é o nó raiz.

Para a escolha da construção de uma solução inicial para o problema de RSDEE foram implementadas três diferentes estratégias. A primeira adota a configuração de uma solução inicial provinda pelo próprio arquivo do SDEE; a segunda é a execução do algoritmo de PRIM gerando uma árvore geradora mínima por meio de pesos nas linhas de distribuição, ou arestas; a terceira estratégia baseou-se na heurística GRASP que assim como o PRIM cria uma árvore geradora mínima por meio dos pesos nas arestas, mas a escolha de uma nova aresta é realizada de forma aleatória a partir de uma pequena lista de possíveis arestas a serem escolhidas, a lista *RCL*.

A utilização de três diferentes estratégias de construção de uma solução inicial foi feita para avaliar a qualidade das soluções entre as estratégias adotadas, e de como a técnica proposta se comporta com cada solução inicial adotada. A discussão sobre a qualidade das soluções de cada estratégia será aprofundada no Capítulo 5, que trata dos testes e resultados obtidos. A forma com que as soluções são construídas pelo algoritmo de PRIM e a heurística GRASP, assim como os pesos adotados para tais técnicas, será tratada em seções posteriores.

A proposta de uma solução é construída e configurada em um vetor estático N , onde armazenam-se as informações de chaves abertas e fechadas de todo o sistema, e outras informações que serão descritas nas seções posteriores. A partir da leitura desse vetor estático constrói-se a árvore do sistema radial, que é composta por todas as barras do sistema, e pelos ramos que possuem a chave fechada no vetor estático N .

Construída a solução em árvore dinâmica, então, deve-se resolver um problema de fluxo de potência para encontrar as perdas elétricas da proposta de solução. Tal configuração

é adotada como solução inicial, ou solução corrente. Dá-se início então ao procedimento de busca local de uma solução com melhor função objetivo (perdas elétricas) por meio da BT.

A BT ocorre em dois estágios. O primeiro estágio parte da solução inicial gerada por um dos três procedimentos descritos anteriormente e busca a cada iteração a solução vizinha do espaço de busca que tenha as menores perdas elétricas. No caso da solução possuir uma melhor função objetivo do que a solução corrente, ela passa a ser a nova solução corrente e o processo se reinicia. Ainda, durante todo o procedimento do primeiro estágio reserva-se a solução de pior função objetivo já visitada. Uma vez que em uma iteração a solução corrente do primeiro estágio não sofra modificação, passa-se para o segundo estágio, o qual transfere a pior solução vizinha encontrada no primeiro estágio como solução corrente do segundo estágio e reinicia o procedimento de BT. Ao final do procedimento do segundo estágio mantém-se como a melhor solução, aquela de melhor qualidade entre as adquiridas no primeiro e segundo estágios. A BT tem sua execução interrompida quando alcança o limite de suas iterações previstas ou quando em uma iteração do segundo estágio a solução corrente não apresenta mudança (melhoria).

O movimento entre uma solução e outra, também conhecido como vizinhança, trabalha como em Zhang *et al.* (2007) e Possagnolo (2015). O objetivo é identificar os ciclos internos independentes e para estes ciclos se abre e fecharum par de chaves até que todas as combinações para cada ciclo sejam realizadas. Tal procedimento sempre garantirá uma topologia radial para o sistema.

Para cada configuração radial gerada, a mesma deve ser construída no formato da árvore dinâmica para que seja submetida ao procedimento de cálculo de fluxo de potência. A árvore é apagada, restando somente o nó raiz (subestação), a solução vizinha gerada é configurada no vetor estático N , e então novamente se constrói a solução na árvore dinâmica e se calcula suas perdas elétricas, assim como ocorreu para a solução inicial.

Para cada solução gerada, avaliam-se suas perdas elétricas que, se forem menores que as perdas elétricas da solução corrente, a configuração da solução armazenada no vetor estático N é transferida para o vetor estático N_{best} . A solução resultante no vetor estático N_{best} , ao final de cada iteração do procedimento de busca, é a melhor solução encontrada no espaço de busca de soluções.

Durante todo o procedimento da BT se reservam as dez melhores soluções já visitadas por meio de um a estrutura de dados do tipo Fila que trabalha com a política de inserção LIFO (*Last In First Out*). Uma solução participará desta Fila - se e somente se - sua função objetivo for melhor do que a função objetivo da solução armazenada na primeira posição dessa Fila.

Quando a Fila estiver cheia e uma nova solução se enquadrar para participar da mesma, a solução mais antiga inserida sai, liberando um novo espaço de inserção.

Terminado todo o procedimento da BT, inicia-se o processo de Pós-Otimização por meio da técnica de PR. O procedimento é executado para as dez soluções elite encontradas durante o procedimento de BT. O PR atua entre cada par de soluções elite, em uma direção somente, onde a solução inicial sempre possui uma função objetivo pior do que a solução guia. O objetivo é se criarem soluções intermediárias, utilizando-se da diversificação que a técnica propõe, igualando-se a solução intermediária com a guia por meio de um ramo por vez, tal escolha é realizada de maneira aleatória. O intuito é descobrir novas regiões de busca que possam gerar soluções ainda não visitadas.

Durante todo o procedimento do PR, a cada solução intermediária encontrada e avaliada, compara-se sua função objetivo com a função objetivo da melhor solução já visitada. No caso de ser melhor, atualiza-se a incumbente. Ao final do PR é apresentada a melhor solução encontrada.

4.3 FORMATAÇÃO DOS DADOS E FUNCIONALIDADES

Nesta seção será explicado detalhadamente como os dados foram armazenados nas estruturas de dados utilizadas, e como foram manipulados para as devidas funcionalidades implementadas. A proposta aqui implementada abrange as seguintes funcionalidades, que serão melhor explicadas nas próximas subseções:

- Estruturas de dados utilizadas e como os dados são armazenados;
- Estratégias de construção de uma configuração radial inicial (base, PRIM e GRASP);
- O procedimento de busca local realizado pela BT a partir da solução inicial;
- O procedimento de pós-otimização realizado pela técnica de PR;
- Representação de uma configuração radial utilizando árvores dinâmicas;
- Cálculo do fluxo de potência de uma configuração radial armazenada em uma árvore dinâmica;
- Buscar ciclos internos independentes e estratégia de vizinhança;
- Proposta alternativa de redução de vizinhança.

4.3.1 Formatação Inicial dos Dados

Para o início do fluxograma (Figura 10), necessita-se preparar as estruturas de dados para receber os dados contidos nos arquivos de um determinado problema de RSDEE. Os dados de um problema de RSDEE apresentam as seguintes informações:

- Números de barras do sistema da solução inicial;
- Número de ramos com chave fechada, configurando uma solução inicial;
- Nó de referência (subestação);
- Tensão do nó de referência (p.u.);
- Tensão base do sistema (kV);
- Potência base do sistema (kVA);
- Erro máximo permitido de perdas elétricas;
- Tabela de potências ativas, reativas indutivas e reativas capacitivas para cada barra do sistema;
- Tabela de impedância (resistência e reatância) para cada ramo do sistema.

O procedimento descrito neste trabalho realizou uma pequena adaptação nos arquivos, informando, ao início do arquivo, a quantidade total de ramos de ligação do sistema.

As informações iniciais do problema: quantidade total de barras, quantidade total de ramos, nó de referência da subestação, tensão do nó da subestação, tensão base do sistema, potência base do sistema e limite de erro do cálculo das perdas, foram armazenadas em variáveis específicas.

Notou-se que certas informações poderiam ser agrupadas, utilizando-se a estrutura de dados registro e se construíram quatro estruturas registro para representar os dados principais de um problema de RSDEE: *Node*, *Tensao_Carga*, *V_node*, e *V_aux*.

O vetor unidimensional estático de registro *Node*, com tamanho equivalente à quantidade total de ramos do sistema, foi construído e configurado para armazenar as informações vinculadas aos ramos de ligação. As informações, provindas dos arquivos de um problema de RSDEE, para cada elemento deste vetor, ou seja, para cada ramo de ligação do sistema, são as seguintes:

- *barra_1*, *barra_2*: representam os índices dos nós (vértices) do ramo de ligação em questão;

- *resist, reat*: variáveis que recebem os valores de impedância (resistência e reatância) do sistema do ramo em questão;
- *chave*: condição da chave (aberta “0”, ou fechada “1”) do ramo de ligação do ramo em questão;
- *I_real, I_imag*: valores reais e imaginárias das correntes dos ramos usados nos cálculos de fluxo de potência do ramo em questão.

Os valores das variáveis *barra_1* e *barra_2*, são os índices dos nós do sistema exatamente como lidos e carregados dos arquivos do sistema, assim como os valores de impedância. O valor da chave do ramo de ligação muda conforme uma solução é construída. Os valores reais e imaginários da corrente se atualizam conforme é executado o algoritmo de fluxo de potência.

O vetor *Node* contém todas as informações de todos os ramos do sistema. Esse vetor contém, sempre, uma configuração de chaves para os ramos que resulta em um sistema radial. A partir de uma configuração radial do vetor *Node* é construída a árvore dinâmica do sistema radial para ser realizado o cálculo de fluxo de potência.

Outro vetor unidimensional estático, mas com o tamanho equivalente à quantidade total de barras do sistema, é o vetor de registro *Tensao_Carga*. Este vetor agrupou informações pertinentes às barras do sistema, ou nós do sistema. As informações para cada elemento deste vetor são as seguintes:

- *pot_at, pot_reat*: informações referentes às potências ativas e reativas de cada barra;
- *V_real, V_imag*: valores das tensões reais e imaginárias de cada barra para o procedimento de fluxo de potência.

Assim como no vetor *Node* as informações das potências são lidas e carregadas dos arquivos do sistema de RSDEE e armazenadas no devido índice do vetor *Tensao_Carga* que representa o mesmo índice da barra no arquivo, ou seja, o vetor não é preenchido de forma sequencial e sim dependendo da ordem com que as barras se encontram dispostas no arquivo do problema de RSDEE. Os valores das tensões das barras só são calculados e atualizados durante o procedimento de fluxo de potência.

A razão da forma de preenchimento dos valores das potências no vetor *Tensao_Carga* facilita o cálculo de fluxo de potência. Quando se está trabalhando com o vetor *Node*, no procedimento de fluxo de potência, a informação de um ramo de ligação, que é como o

procedimento percorre a árvore, contém as informações das barras deste ramo. Os valores das tensões a serem calculados, por barra, são encontrados/acessados por acesso direto à memória no vetor *Tensão_Carga* pelos valores contidos nas variáveis *barra_1* e *barra_2* do vetor *Node*, eliminando um percurso vetorial.

Uma outra estrutura de registro principal é a estrutura dinâmica *V_Node*. O registro define um novo tipo de dado que será utilizado na construção dos elementos da árvore dinâmica. De forma geral, a árvore do sistema recebe a partir da leitura do vetor *Node* as informações dos elementos com chave fechada (ramos ativos) e constrói sua estrutura dinâmica somente com os índices das barras contidas nestes elementos. Os elementos (ramos de ligação) que possuem chave aberta não farão parte da solução construída pela árvore dinâmica.

A estrutura *V_Node* possui as seguintes informações:

- *v*: contém o índice do nó (vértice/barra);
- *cont_p*: armazena a quantidade de conexões da barra (filhos deste nó);
- *id*: variável de controle para auxílio no procedimento de buscar ciclos;
- *V_Node *p[]*: um ponteiro que possui um endereço de memória de um nó filho, representa o ramo ativo de cada filho a outro filho (nó/vértice/barra).

A última estrutura de registro utilizada é a estrutura dinâmica *V_aux*. Tal estrutura é configurada como uma lista dinâmica simplesmente encadeada. Ela armazena todos os índices das barras do sistema, menos a da subestação. Essa lista é utilizada como suporte para a construção da árvore dinâmica. Conforme uma nova barra é adicionada a estrutura de árvore *V_Node* tal índice é removido da lista *V_aux*, assim não se permite que um mesmo índice possa ser acessado/inserido mais de uma vez.

A partir deste momento, com as estruturas básicas explicadas e suas informações destacadas, pode-se entender melhor como as funcionalidades descritas a seguir são executadas.

4.3.2 Configuração de uma Solução Inicial

Nesta seção serão explicados os três procedimentos utilizados para se obter uma configuração inicial para o problema de RSDEE.

4.3.2.1 Solução Base

Todos os arquivos apresentam uma proposta de solução inicial de acordo com a configuração de certas informações contidas nos arquivos do sistema adotado.

Primeiramente, para um grafo de n nós e m arestas, sabe-se que uma árvore necessita de $n-1$ arestas que conectem todos os n nós. As informações das impedâncias dos ramos são descritas uma a uma e linha por linha no arquivo de sistemas. As $n-1$ linhas, geralmente, perfazem a proposta de uma solução inicial. Alguns arquivos não possuem essa configuração, onde a descrição de uma proposta inicial se encontra em sua referência literária específica.

Dessa forma, no início do algoritmo no momento de leitura e carregamento dos dados já se tem a informação da quantidade de barras do sistema (quantidade de nós/vértices) e conforme se carrega as informações das impedâncias no vetor V_Node , nas variáveis *resist* e *reat*, se fecham as chaves (*chave* = 1) somente das primeiras *barras-1* linhas das descrições das impedâncias dos ramos. Os ramos restantes terão suas chaves abertas (*chave* = 0), caracterizando assim uma configuração de uma solução radial no vetor V_Node .

4.3.2.2 O Algoritmo PRIM

Este algoritmo usado para a obtenção de uma solução inicial para o problema de RSDEE serve para comparar os resultados obtidos por meio desta técnica e o GRASP implementado.

O algoritmo de PRIM (CORMEN *et al.*, 2001; DROZDEK, 2012) é um algoritmo construtivo guloso para se encontrar uma árvore geradora mínima de um determinado grafo valorado (pesos relacionados às arestas) e não direcionado. O objetivo é obter um grafo que tenha todos os vértices conectados, e que caracterize uma árvore (sem a presença de ciclos) onde a soma dos pesos das arestas resultantes seja o mínimo.

A proposta de peso para os ramos de ligação para o sistema de RSDEE foi a utilização do fluxo de potência com todos os ramos fechados caracterizando uma rede malhada, e calculando o fluxo de potência aparente para cada ramo, o que leva em consideração a potência aparente.

Esses dados foram gerados e armazenados em um vetor estático FC_ini onde cada índice de posição deste vetor remete ao mesmo índice de posição do vetor de registro V_Node para que no momento de percurso de um vetor as informações do outro vetor fossem encontradas pela mesma posição do percurso por acesso direto à memória. A pequena diferença para com o algoritmo de PRIM é que para este problema se busca a soma resultante máxima de fluxo de potência para os ramos ativos, isto é, resolve-se o problema da árvore geradora máxima. A proposta do algoritmo de PRIM encontra-se na Figura 11.

Figura 11 – Algoritmo PRIM implementado.

1	Declaração de variáveis de controle
2	Escolha do ramo inicial
3	Fechar chave do ramo inicial
4	Executar ($barras - 1$) vezes
5	$dmin = 1.0$
6	Para cada par de ramos ij
7	Avaliar se um ramo i está conectado a um ramo j
8	Se $FC_ini[j] \geq dmin$
9	Avaliar se existe um ramo k conectado ao ramo j , que não esteja conectado ao ramo i
10	Se SIM, avalie um novo ramo j
11	Se NÃO, $dmin = FC_ini[j]$, salvar o índice do ramo j em m
12	Fechar a chave do ramo de índice m

Fonte: Próprio autor

A escolha do ramo inicial é realizada avaliando os pesos de todos os ramos pertencentes à subestação e escolhendo o de maior fluxo de potência para que o ramo tenha a chave fechada (Figura 11 - Linha 2-3). Como se sabe somente $barras - 1$ ramos poderão estar configurados com chave fechada para caracterizar a solução com a propriedade radial (árvore), executa-se somente esta quantidade de vezes ($barras - 1$) o algoritmo (Figura 11 – Linha 4). Assim, se inicia um procedimento de avaliação, a cada iteração, de todos os possíveis ramos a participar da solução em construção, e dentre possíveis ramos, somente o que possuir o maior fluxo de potência que obterá sua chave fechada (Figura 11 – Linhas 5-12).

Cada iteração inicia-se com a variável $dmin = 1$ com o objetivo de ser uma variável de controle do maior fluxo de potência avaliado e armazenado a cada iteração do procedimento (Figura 11 – Linha 5). A cada iteração do algoritmo ela armazena o maior fluxo de potência de um possível ramo avaliado, e a variável de controle m armazena o índice deste

ramo (Figura 11 – Linha 11), assim ramos possíveis com menores fluxos de carga não chegarão a ser avaliados.

A partir dos ramos terminais ativos (chave fechada) da solução em construção se buscam todos os ramos conectados à estes. Para cada ramo encontrado, se o fluxo de potência do ramo for maior do que a variável de controle $dmin$, o ramo é selecionado como possível de participar da solução corrente. Em seguida, a partir do ramo possível escolhido, novamente se percorrem ramos possíveis a partir deste na tentativa de se encontrar um ramo conectado a outro que já possua uma chave fechada (Figura 11 – Linhas 6-9). Caso o fato ocorra, o ideal é que se descarte o ramo possível escolhido e se escolha um novo (Figura 11 – Linha 10), o que evitaria a formação de ciclos na busca.

No término de cada iteração do procedimento de avaliação de ramos (Figura 11 – Linha 7) se tem armazenado o maior fluxo de potência de um ramo avaliado possível de entrar na solução em construção e o índice de tal ramo. Assim se fecha a chave de tal ramo, no devido índice armazenado em m no vetor *Node*. O processo reinicia retornando a variável $dmin$ a 1, e repete este processo $barras - 1$ vezes para que a quantidade correta de chaves sejam desligadas perfazendo uma estrutura radial no vetor *Node* e terminando de construir a configuração de uma solução.

4.3.2.3 O Algoritmo GRASP

O algoritmo GRASP, como explicado no Capítulo 2 é um algoritmo construtivo guloso generalizado e com restrições quanto à escolha dos elementos que farão parte da solução em construção, no intuito de que mesmo se partindo de um mesmo ponto de partida as soluções geradas serão de diferentes configurações, caracterizando um procedimento semi-aleatório.

Algumas características utilizadas no algoritmo de PRIM também foram utilizadas para este algoritmo GRASP. Os pesos dos ramos continuam sendo os valores das potências aparentes calculadas por um fluxo de potência em malha fechada do sistema.

A grande diferença dessa proposta quando comparada ao algoritmo PRIM é que a escolha do ramo possível no algoritmo de PRIM é o ramo com o maior fluxo de potência, enquanto que para o algoritmo GRASP o próximo elemento a ser escolhido vem de uma lista

restrita *RCL* dos ramos de maior fluxo de potência onde um é escolhido de forma aleatória. Desta forma diferentes configurações de soluções podem ser construídas. O algoritmo da proposta implementado para o GRASP encontra-se na Figura 12.

Figura 12 – Algoritmo GRASP implementado.

1	Declaração de variáveis de controle
2	Calcular a média do somatório dos fluxos de carga de todos os ramos do sistema, m_{Scf_1}
3	Iterar it vezes
4	Abrir as chaves de todos os ramos do sistema
5	Escolha do ramo inicial
6	Iniciar <i>RCL</i> e <i>index_RCL</i>
7	Criar <i>RCL</i> com os fluxos de carga dos ramos candidatos, ordenada em ordem decrescente
8	Criar <i>index_RCL</i> com os índices dos ramos candidatos
9	Atualizar <i>RCL</i> e <i>index_RCL</i> como lista restrita, escolher aleatoriamente um índice do vetor <i>index_RCL</i> , e fechar a chave deste ramo
10	Executar ($barras - 1$) vezes
11	Iniciar <i>RCL</i> e <i>index_RCL</i>
12	Para cada par de ramos ij
13	Avaliar se um ramo i está conectado a um ramo j
14	Avaliar se existe um ramo k conectado ao ramo j , que não esteja conectado ao ramo i
15	Se SIM, avalie um novo ramo j
16	Se NÃO
17	Insira o fluxo de potência do ramo escolhido no vetor <i>RCL</i> , na ordem decrescente
18	Insira o índice do ramo escolhido no vetor <i>index_RCL</i>
19	Atualizar <i>RCL</i> e <i>index_RCL</i> como lista restrita, escolher aleatoriamente um índice do vetor <i>index_RCL</i> , e fechar a chave deste ramo
20	Calcular a média do somatório dos fluxos de carga dos ramos fechados da solução construída, m_{Scf_2}
21	Caso $m_{Scf_2} > m_{Scf_1}$, atualize a solução como melhor solução, e $m_{Scf_1} = m_{Scf_2}$
22	Retornar a solução armazenada

Fonte: Próprio autor

Foram construídas duas listas restritas para esta proposta do algoritmo GRASP. A lista *RCL* contém todos os valores dos fluxos de carga dos ramos candidatos a participar da solução corrente ordenados de forma decrescente de fluxo de potência aparente, e a lista *index_RCL* contém todos os índices dos ramos da lista *RCL*, ambas implementadas como vetores estáticos. Os elementos de ambas as listas estão organizados da seguinte forma: o fluxo de potência aparente armazenado em uma determinada posição i no vetor da lista *RCL*

está relacionado ao índice desse mesmo ramo que está armazenado na mesma posição i no vetor da lista $index_RCL$. As listas apresentam todos os ramos candidatos a participarem da solução em construção.

A partir das listas RCL e $index_RCL$, de todos os ramos candidatos e de seus índices, parte-se para a construção da lista restrita do GRASP adaptando as listas pré-mencionadas. Em (22), encontra-se a equação que define a escolha dos ramos candidatos para permanecerem nas listas RCL e $index_RCL$ como elementos da lista restrita,

$$RCL = \{ij \in \Omega | FC_{ij}^{min} + \alpha(FC_{ij}^{max} - FC_{ij}^{min}) \leq FC_{ij} \leq FC_{ij}^{max}\} \quad (22)$$

onde ij representa o par de barras entre um ramo candidato, Ω representa o domínio, FC_{ij}^{min} e FC_{ij}^{max} indicam, respectivamente, os fluxos de potência aparente mínimo e máximo dentre os ramos candidatos, α representa um valor no intervalo $[0,1]$ que indica o medidor da aleatoriedade ou não do GRASP, e FC_{ij} é o valor do fluxo de potência aparente de um determinado ramo candidato verificando se está apto a participar da lista restrita RCL atualizada.

A partir do momento da declaração das variáveis de controle do algoritmo (Figura 12 – Linha 1) o algoritmo GRASP calcula uma média aritmética de todos os ramos ativos do sistema e armazena em m_Scf_1 (Figura 12 – Linha 2), e constrói um total de it soluções (Figura 12 – Linha 3) definido pelo usuário.

Dependendo da dimensão do sistema de distribuição de energia elétrica avaliado, a quantidade de diferentes configurações de soluções aumenta, sendo cabível um aumento gradativo da quantidade de soluções geradas pelo GRASP (it iterações).

Para cada geração de uma solução se abrem todas as chaves do sistema (Figura 12 – Linha 4) para que, então uma a uma chave seja fechada de acordo com a execução do algoritmo GRASP.

A escolha do ramo inicial (Figura 12 – Linhas 5-9) ocorre de forma similar ao PRIM, com pequenas modificações. Se percorre o vetor $Node$ pela busca do nó de referência da subestação e se inserem nas listas RCL e $index_RCL$ os fluxos de potência aparente dos ramos conectados à subestação e seus índices, respectivamente.

Como citado anteriormente, a lista RCL dos fluxos de potência aparente apresenta todos os ramos candidatos numa ordem decrescente, e para cada elemento da lista RCL o

índice do devido ramo está numa mesma posição da lista de índices dos ramos escolhidos, a lista *index_RCL* (Figura 12 – Linhas 6-8).

Em seguida atualizam-se as listas *RCL* e *index_RCL* para servirem como a lista restrita de fluxos de potência aparente, respeitando (16), restando somente os fluxos de potência aparente e índices de ramos que podem vir a participar da solução em construção. Então, dessas listas restritas atualizadas somente um elemento é aleatoriamente escolhido para fazer parte da solução em construção (Figura 12 – Linha 9), cuja chave do ramo de ligação é fechada. A partir da escolha do ramo inicial para o procedimento do algoritmo GRASP, parte-se para a escolha dos ramos restantes (Figura 12 – Linhas 10-19).

A forma com que se executa o procedimento de busca dos ramos de ligação restantes é muito próxima da forma com que o algoritmo PRIM foi implementado, adicionando a semi-aleatoriedade. Novamente, fixa-se na quantidade de iterações que o procedimento deve possuir para que construa uma solução radial (Figura 12 – Linha 10) executando *barras* – 1 vezes para a solução final caracterizar uma árvore.

A comparação entre os ramos que ocorre para o algoritmo GRASP (Figura 12 – Linhas 13-15) é praticamente idêntica à proposta do algoritmo PRIM, uma vez que esta comparação é realizada para que se evite a formação de ciclos. No momento que um ciclo não é detectado, a escolha dos possíveis ramos, a construção e configuração dos vetores de lista *RCL* e *index_RCL*, e a escolha de um ramo de forma aleatória a partir das listas restritas *RCL* e *index_RCL* atualizadas (Figura 12 – Linhas 16-19), possui a mesma forma com que se obteve o ramo inicial, conforme explicado anteriormente.

Para cada solução construída é calculada a média aritmética dos fluxos de potência aparente dos ramos com chave fechada e armazenada na variável de controle *m_Scf_2* (Figura 12 – Linha 20). Quando tal valor for maior do que a outra média armazenada *m_Scf_1* (Figura 12 – Linha 21), a solução encontrada passa a ser a melhor solução e atualizam-se os valores das médias de fluxo de potência aparente (Figura 12 – Linha 21) para a próxima iteração.

Dessa forma ao término do algoritmo GRASP a solução de maior média aritmética de fluxo de potência aparente será a solução adotada como solução inicial e a melhor solução encontrada pelo algoritmo.

4.3.3 Busca Tabu como procedimento de Busca Local

A BT foi escolhida para este trabalho para realizar a etapa de busca local por ser uma meta-heurística de busca exaustiva em um espaço de busca onde a melhor solução passa a ser a próxima solução corrente do próximo espaço de busca. O procedimento deve seguir até que seja respeitada uma determinada característica do problema, podendo ser quantidade máxima de iterações, a não melhoria da função objetivo por um número determinado de iterações, ou especificidades do problema em questão.

Verifica-se que dependendo da vizinhança adotada, os espaços de busca podem ser de grande dimensão, necessitando-se, então, de tratamentos específicos como redução de vizinhança ou mesmo a inserção das características da BT indicadas na Figura 3. Tais tratamentos reduzem o espaço de busca fazendo com que o algoritmo possa ser mais rápido em sua execução sem comprometer o resultado final.

A Diversificação foi implementada com termos de memória adaptativa de curto prazo onde os movimentos que geraram a melhor solução de um determinado espaço de busca ficam bloqueados por uma quantidade pré-determinada de iterações. A quantidade de movimentos a serem bloqueados e a quantidade de iterações que o bloqueio de tais movimentos permanecerá devem ser realizados por meio de testes, pois sistemas maiores demandam mais iterações o que requer que mais movimentos possam ser bloqueados e por mais tempo.

Diferente do procedimento clássico da memória de curto prazo onde existe o bloqueio de movimentos de certos atributos de uma solução e a aspiração dos mesmos, neste trabalho não foi implementada a aspiração dos elementos bloqueados, visto que os elementos bloqueados não interferem em soluções correntes futuras. A liberação de movimento dos atributos bloqueados se dá somente pelo término do período de iterações de bloqueio de cada atributo.

Outra característica implementada é a Intensificação que ocorre somente uma vez em todo o procedimento do algoritmo. São executados dois procedimentos de busca local no algoritmo, um antes e um após a intensificação, identificados como estágios um e dois da BT na Figura 10. O primeiro estágio é executado até se encontrar uma iteração da BT onde não se ocorre uma melhoria da função objetivo. Durante esse estágio reservam-se a melhor e a pior solução visitadas. Então, a pior solução visitada passa a ser a solução corrente para o início do segundo estágio.

O segundo estágio ocorre assim como o primeiro, e termina quando não encontra, em uma iteração, uma solução melhor do que a solução corrente. Ao término do segundo estágio compara-se a melhor solução já encontrada no primeiro estágio, com a melhor solução do segundo estágio, e escolhe a melhor entre elas como a melhor solução obtida no procedimento de busca local. Na Figura 13 encontra-se o algoritmo explicativo que descreve a forma do procedimento adotado para a BT como estratégia de busca local.

Antes do início da BT necessita-se construir a solução inicial por um dos três métodos descritos neste trabalho (do Arquivo, PRIM ou GRASP), configurando o vetor de ramos N com uma solução radial e então informando tal solução como a solução corrente (Figura 13 – Linha 1). Após esse passo, inicializa-se o vetor unidimensional de bloqueios de ramos, que possui o mesmo tamanho do vetor de ramos N , onde cada posição é preenchida com um valor inteiro que informa a quantidade de iterações da BT que aquele ramo está com seu movimento bloqueado (Figura 13 – Linha 2).

A BT (Figura 13 – Linhas 4-35) é executada até que sejam respeitadas uma de duas características. Uma delas é a BT executar até determinado limite de iterações pré-definidas (*bt_cont*) (Figura 13 – Linha 4), a outra característica ocorre quando em uma iteração a melhor solução de um determinado espaço de busca não possuir uma função objetivo melhor do que a solução corrente (Figura 13 – Linha 37).

Como informado na Figura 10 a BT ocorre em dois estágios. O primeiro estágio (Figura 13 – Linhas 5-21), a cada iteração busca a melhor solução de um espaço de busca e caso a melhor solução do espaço de busca possuir função objetivo melhor do que a função objetivo da solução corrente, esta passa a ser a nova solução corrente para a próxima iteração.

Uma vez que essa solução corrente não muda em uma iteração passa-se para o segundo estágio (Figura 13 – Linhas 22-37) armazenando-se a melhor solução já visitada até este momento, e atualizando a solução corrente com a pior solução corrente e reiniciando-se o procedimento de BT novamente, até que, mais uma vez a solução corrente não sofra mudança.

O primeiro estágio se inicia informando uma variável *booleana* (Pior_solução) com valor falso, o que indica que ainda não piorou a solução para o início do segundo estágio (Figura 13 – Linha 5).

Figura 13 – Algoritmo de busca local com a BT.

1	Gerar solução inicial e configurar como solução corrente
2	Iniciar vetor de bloqueios
3	Pior_solução = FALSE
4	Início da BT para <i>bt_cont</i> iterações
5	Se (Pior_solução == FALSO)
6	Buscar Ciclos
7	Atualiza vetor de bloqueios
8	Avaliar Vizinhança
9	Redução de vizinhança
10	Para cada solução possível de ser avaliada
11	Verificar factibilidade
12	Se (Factível)
13	Verificar e armazenar a melhor solução avaliada desta iteração
14	Verificar e armazenar a pior solução avaliada até este momento
15	Verificar participação na matriz de PR
16	Se a melhor solução da vizinhança for melhor que a corrente
17	Bloquear o movimento dos ramos que modificaram os status de chaves
18	Atualizar a incumbente com a melhor solução visitada
19	Senão
20	Armazenar a melhor solução encontrada até este momento
21	Atualizar a incumbente com a pior solução avaliada
22	Pior_solução = VERDADEIRO
23	Senão
24	Buscar Ciclos
25	Atualizar vetor de bloqueios
26	Avaliar Vizinhança
27	Redução de vizinhança
28	Para cada solução possível de ser avaliada
29	Verificar factibilidade
30	Se (Factível)
31	Verificar e armazenar a melhor solução avaliada desta iteração
32	Verificar participação na matriz de PR
33	Se a melhor solução da vizinhança for melhor que a incumbente
34	Bloquear o movimento dos ramos que modificaram os status de chaves
35	Atualizar a incumbente com a melhor solução avaliada
36	Senão
37	Terminar a BT
38	Verificar a melhor solução entre as melhores soluções encontradas antes e depois de piorar a solução
39	A apresentar a melhor solução encontrada no procedimento de BT

Fonte: Próprio autor

Antes da geração das soluções vizinhas atualiza-se o vetor de bloqueios reduzindo em um os valores contidos nos ramos que possuem bloqueios, fazendo com que estejam em uma iteração mais próximos de poderem modificar seu status de chave (Figura 13 – Linha 7).

Inicia-se, então, a avaliação da vizinhança por meio das permutações da matriz de ciclos (Figura 13 – Linhas 8-15). Cada solução gerada passa por um procedimento de redução de vizinhança que verifica se tal solução respeita os critérios para que esta seja avaliada (calculada suas perdas elétricas), e caso não respeitando simplesmente passa-se para a geração da próxima solução vizinha. Tal procedimento será melhor detalhado na subseção 4.3.6.

No caso de uma solução respeitar os critérios de redução de vizinhança, então são calculadas suas perdas elétricas (função objetivo) e, verifica-se a factibilidade da solução avaliada, que envolve a verificação dos limites inferiores e superiores de tensão (valores em p.u.) em todas as barras.

Uma solução sendo factível, ou seja, para todos os seus ramos ativos, a tensão está dentro dos limites inferior e superior, verifica-se se sua função objetivo é melhor do que a função objetivo da solução corrente. Em caso positivo armazena-se tal solução em um vetor N_best que possui as mesmas características do vetor N . Para cada solução factível também verifica-se se a função objetivo é a pior gerada até o momento, que em caso positivo armazena-se a solução no vetor N_worst (Figura 13 – Linhas 13-14), que também possui as mesmas características do vetor N . Ainda se checa se essa solução pode participar da matriz de soluções elite, que irá compor até as dez melhores soluções já avaliadas para a execução da técnica de PR (Figura 13 – Linha 15).

Esta matriz de soluções elite foi implementada como uma fila com a lógica de inserção LIFO, como descrito anteriormente, onde a entrada de uma nova solução nessa fila só será possível se e somente se a função objetivo de uma solução avaliada na vizinhança de uma iteração da BT for melhor do que a função objetivo da última solução armazenada na fila. Isso faz com que a matriz de soluções elite seja composta por até as dez melhores soluções obtidas em todo o procedimento de BT, uma vez que a verificação de participação de uma solução avaliada para compor a matriz de soluções elite ocorre em ambos os estágios da BT.

Terminado o procedimento de avaliação de uma determinada vizinhança, ou seja, terminada a avaliação de todas as soluções provindas de todas as permutações da matriz de ciclos, a solução resultante no vetor N_best possui a solução de melhor função objetivo da vizinhança. No caso de N_best possuir a função objetivo melhor do que a função objetivo da solução corrente (Figura 13 – Linha 16), atualiza-se o vetor de bloqueios de ramos informando que para os ramos que modificaram seu status de chave para construir a solução armazenada em N_best estão com seus movimento bloqueados por certo número de iterações (Figura 13 – Linha 17). Então se atualiza a solução corrente armazenada em N pela melhor

solução da vizinhança armazenada em N_{best} (Figura 13 – Linha 18) e se reinicia o processo do estágio 1 (Figura 13 – Linhas 5-18).

Caso a solução armazenada em N_{best} não possua uma função objetivo melhor do que a função objetivo da solução corrente (Figura 13 – Linha 19) armazena-se a melhor solução encontrada no estágio 1 (Figura 13 – Linha 20) em N_{top} , que possui as mesmas características do vetor N , e se atualiza a solução corrente com a pior solução já avaliada no primeiro estágio da BT que se encontra armazenada em N_{worst} (Figura 13 – Linha 21). Em seguida troca-se o valor *booleano* da variável *Pior_solução* para verdadeiro (Figura 13 – Linha 22) fazendo com que ative o segundo estágio da BT (Figura 13 – Linhas 23-37).

O segundo estágio inicia-se por meio da avaliação do valor booleano da variável *Pior_solução* sendo verdadeira (Figura 13 – Linha 23). O segundo estágio ocorre com as mesmas características do primeiro estágio, mas no procedimento de avaliação de uma solução vizinha não se reserva mais a solução de pior função objetivo. A melhor solução resultante do segundo estágio encontrar-se-á armazenada no vetor N_{best} (Figura 13 – Linha 35). No momento em que uma vizinhança avaliada não possuir uma solução armazenada em N_{best} cuja função objetivo seja melhor do que a função objetivo da solução corrente a BT finaliza sua execução (Figura 13 – Linha 37).

Terminado o procedimento da BT têm-se duas soluções alcançadas, sendo N_{top} alcançada no primeiro estágio e N_{best} alcançada no segundo estágio. Cada solução é a melhor solução alcançada em cada estágio. Assim, comparam-se ambas as soluções (Figura 13 – Linha 38) e a melhor dentre as duas é repassada para N_{top} e é apresentada como a melhor solução fornecida pela BT (Figura 13 – Linha 39).

4.3.4 Path Relinking como procedimento de Pós-Otimização

Como explicado no subcapítulo 2.2.3 a técnica de PR é utilizada como procedimento de pós-otimização e ocorre entre as dez melhores soluções denominadas soluções elite, obtidas durante o procedimento de BT, e armazenadas na matriz de soluções elite, ou matriz de PR.

A matriz de soluções elite foi implementada como uma Fila pela política de inserção LIFO, mas de forma estática. Essa estratégia faz com que a fila seja atualizada com seus

procedimentos de inserção e remoção. Isso faz com que os valores contidos na fila não estejam organizados de forma crescente ou decrescente. Na Figura 14 encontra-se o algoritmo explicativo desta estratégia.

Figura 14 - Algoritmo da proposta de PR.

1	Construir os vetores N_guia e N_int
2	Para i de 1 a 9
3	Para j de $i+1$ até 10
4	Calcular as perdas elétricas das soluções das linhas i,j da matriz de soluções elite
5	Armazenar em N_guia a solução de menor perda elétrica
6	Armazenar em N_int a solução de maior perda elétrica
7	Faça
8	$Similar = verdadeiro$
9	Inicializa vet_pr
10	Percorrer N_guia e N_int verificando a cada posição o status de chaves diferentes entre ambas
11	Se encontrada uma diferença entre uma chave
12	$Similar = falso$
13	Salva o índice do ramo em vet_pr
14	Se ($Similar == falso$)
15	Escolher um elemento aleatório de vet_pr e armazenar o índice em k
16	Salvar em $chave_int$ o status da chave de N_int armazenado em k
17	Calcular as perdas elétricas e construir a matriz de ciclos de N_int
18	Verificar factibilidade de N_int
19	Se factível, verificar se N_int melhor que N_top
20	Caso afirmativo, atualizar N_top com N_int
21	Verificar na matriz de ciclos a que ciclo l pertence à $chave_int$
22	Se ($chave_int == fechada$)
23	Fechar a chave do ramo aberto da linha l do ciclo de N_int
24	Abrir $chave_int$ em N_int
25	Senão
26	Abrir outra chave aleatória da linha l do ciclo de N_int
27	Fechar $chave_int$
28	Enquanto ($Similar == falso$)

Fonte: Próprio autor

A técnica de PR se inicia construindo os vetores N_guia e N_int , com as mesmas características do vetor de ramos N , que armazenam as características da solução guia e das soluções intermediárias, respectivamente (Figura 14 – Linha 1).

Em seguida percorre-se a matriz de soluções elite, onde a cada linha se compara com as linhas seguintes (Figura 14 – Linhas 2-3) até que todas as comparações tenham terminado. Cada linha da matriz de soluções elite possui as informações de uma configuração radial do

SDEE. Sendo assim, calculam-se as perdas elétricas das configurações radiais das duas linhas i,j da matriz de soluções elite em avaliação e as armazenam nos vetores de ramos N_{guia} e N_{int} , respectivamente, com as configurações radiais de menor e maior perdas elétricas (Figura 14 – Linhas 4-6). Isso significa que temos a melhor solução como guia, e a pior solução como a solução inicial.

Com as soluções, guia e inicial, configuradas, dá-se início ao processo de partir da solução inicial até alcançar a similaridade com a solução guia, por meio de trocas de certos atributos, gerando-se o espaço de busca com as soluções intermediárias (Figura 14 – Linhas 7-28).

O procedimento de geração de soluções intermediárias ocorre enquanto não houver similaridades entre a solução intermediária corrente e a solução guia (Figura 14 – Linha 28). Para isso inicializa-se uma variável *booleana Similar* com valor *verdadeiro* (Figura 14 – Linha 8). Em seguida constrói-se um vetor estático, chamado *vet_pr*, que armazenará os índices dos ramos que possuem o status de chave diferentes entre N_{guia} e N_{int} (Figura 14 – Linha 9). A comparação entre N_{guia} e N_{int} é realizada sempre entre as mesmas posições de elementos. Caso seja encontrada pelo menos uma diferença entre o status de chave das soluções pré-mencionadas atualiza-se a variável *Similar* com *falso*, indicando que a similaridade ainda não foi alcançada entre as soluções (Figura 14 – Linhas 10-13).

Terminado o preenchimento de *vet_pr* checa-se a variável de similaridade (Figura 14 – Linha 14). No caso da mesma ser *verdadeira*, significa que foi alcançada a similaridade entre os status de chaves dos ramos entre as soluções N_{guia} e a última solução intermediária, e então passa-se para as próximas soluções da matriz de soluções elite (linhas $i,j+1$ ou $i+1,j$) e se reinicia o processo. No caso de ser *falsa* escolhe-se aleatoriamente um elemento de *vet_pr*, e armazena o índice do elemento escolhido em k (Figura 14 – Linha 15). No elemento k de *vet_pr* situa-se o índice do ramo que trocará de atributo em N_{int} , para o mesmo atributo em N_{guia} , gerando uma nova solução intermediária.

Guarda-se em *chave_int* o status da chave de N_{int} localizado no ramo de índice armazenado em k (Figura 14 – Linha 16). Isto é necessário para mais à frente verificar se a chave está aberta ou fechada, o que interfere na abertura/fechamento de outras chaves do ciclo correspondente.

É calculada, então, a função objetivo de N_{int} e construída sua matriz de ciclos, assim como ocorre na BT (Figura 14 – Linha 17). A matriz de ciclos é importante para encontrar em

qual ciclo o ramo escolhido para trocar seu atributo pertence, para que então a permutação de chaves ocorra somente neste ciclo.

Prossegue-se para uma verificação de factibilidade da solução construída e avaliada em N_{int} e se em caso factível, verifica-se se sua função objetivo é melhor do que a função objetivo armazenada na melhor solução obtida até o momento, que encontra armazenada em N_{top} . Em caso positivo atualiza-se N_{top} com N_{int} (Figura 14 – Linhas 18-20).

Tendo já construída a matriz de ciclos avalia-se a coluna com o índice de ramo armazenado em k pela busca da linha l que corresponde ao ciclo em que o ramo escolhido para trocar de atributo faz parte (Figura 14 – Linha 21). Caso o ramo escolhido faça parte de mais de um ciclo, escolhe-se aleatoriamente uma linha l dentre as possíveis.

Escolhido o ramo, e seu ciclo correspondente, analisa-se seu status de chave ($chave_{int}$). Caso a chave escolhida ($chave_{int}$) possua status de *fechado*, deve-se fechar a única chave aberta do ciclo l correspondente, e abrir a chave escolhida ($chave_{int}$). Caso o status da chave escolhida ($chave_{int}$) seja *aberto*, avalia-se o ciclo l correspondente e, escolhe-se aleatoriamente uma chave fechada para ser aberta e, então, fecha-se a chave escolhida ($chave_{int}$). Todo esse procedimento ocorre trocando-se os atributos de chave dentro da própria solução intermediária N_{int} (Figura 14 – Linhas 22-27). A solução intermediária resultante a cada iteração (Figura 14 – Linha 28) já possui seus atributos trocados, e passa para a próxima iteração até que a similaridade seja alcançada.

4.3.3 Configuração de uma Solução Radial em uma Árvore Dinâmica

Uma das contribuições deste trabalho é o formato com que a configuração de uma solução é armazenada, não na estrutura estática *Node*, mas em uma estrutura dinâmica, a árvore. Reiterando, a árvore utilizada neste trabalho possui a característica de múltiplos filhos, ou seja, cada nó possui uma quantidade variada de filhos, indicando que uma barra pode estar conectada a diferentes quantidades de barras. A árvore proposta neste trabalho inicia por meio de um ponteiro denominado T_{Node} que aponta para o primeiro endereço de memória da árvore dinâmica, conhecido como raiz da árvore, onde está armazenado o índice da barra que representa a subestação do problema de SDEE em tratamento.

O procedimento de construção da configuração de uma solução para a árvore T_Node a partir da solução armazenada no vetor estático $Node$ ocorre por meio de duas funções, $VTree$ e $Tree$. A função $VTree$ cria uma lista dinâmica $vaux$ com todos os índices de barras, menos o da subestação, que já está inserido na estrutura T_Node e configurado como a raiz da árvore. Na Figura 15 ilustra-se o algoritmo do procedimento de construção de uma solução radial, a partir da leitura do vetor estático $Node$, para a estrutura de árvore dinâmica T_Node .

Figura 15 - Algoritmo de construção de uma solução radial para uma Árvore Dinâmica.

1	Declaração de variáveis de controle
2	Criar a lista dinâmica $vaux$ contendo todos os índices das barras, menos o da subestação
3	Partir do início de $vaux$
4	Enquanto ($vaux \neq \emptyset$) fazer
5	$Encontrado = FALSE$
6	Percorrer o vetor de ramos $Node$
7	Comparar se existe um ramo ativo que tenha como índices o índice do nó avaliado na árvore T_node e um elemento de $vaux$
8	Se SIM
9	Variável de controle $Encontrado = TRUE$
10	Adicione em 1 a quantidade de filhos do nó da árvore avaliado
11	Armazene o índice da barra encontrado na lista $vaux$ em v
12	Apague o elemento v da lista $vaux$
13	Saia do laço de percurso do vetor $Node$
14	Se $Encontrado = TRUE$
15	Criar um novo elemento dinâmico v_novo para a árvore dinâmica
16	Armazenar o índice v em v_novo
17	Conectar v_novo como filho do nó avaliado da árvore
18	Repassar o procedimento a partir da Linha 3 para o nó da árvore v_novo
19	Retornar ao início de $vaux$
20	Se $Encontrado = FALSE$
21	Passar para o próximo elemento de $vaux$

Fonte: Próprio autor

A função $Tree$ lê a lista dinâmica $vaux$ e compara se existe um ramo ativo em $Node$ que possua como índices de um ramo os índices avaliados na árvore T_Node e na lista dinâmica $vaux$. Quando um ramo é encontrado, remove-se o índice escolhido da lista dinâmica $vaux$ e o insere como filho do nó avaliado na árvore T_node . O procedimento termina quando a lista dinâmica $vaux$ se encontra vazia, indicando que todas as barras foram inseridas.

A função *VTree* implementada compõe a primeira função para se criar uma lista simplesmente encadeada (lista dinâmica unidirecional) *vaux* (Figura 15 – Linha 2). A função *Tree* é acionada ainda dentro da função *VTree*, e trabalha com o recurso de recursividade, ou seja, é uma função que chama a si própria (Figura 15 – Linhas 3-20) (CORMEN *et al.*, 2001; DROZDEK, 2012).

A recursividade é utilizada para percorrer os nós da árvore e para cada nó avalia-se a existência de um ramo com o valor do índice de uma barra armazenado no nó. No caso de existir um ramo, um novo nó é criado e anexado como filho, e se parte para uma nova busca a partir do nó filho. No caso de não existir um ramo a função recursiva termina sua execução e retorna, em tempo de execução, para um ponto anterior da recursividade, ou seja, o ponto anterior da função no nó-pai. Assim, para o nó pai verifica-se a existência de mais ramos ativos e todo o processo se repete, até que não se tenham mais elementos para se inserir na árvore.

Parte-se do início da árvore dinâmica *T_node* e da lista dinâmica de índices de barras *vaux* (Figura 15 – Linha 4), e enquanto a lista dinâmica não for vazia, percorre-se o vetor de ramos *Node* se um certo elemento *i* de *vaux* que possua um ramo, com chave fechada, com o valor do índice armazenado no nó da árvore avaliado (Figura 15 – Linha 6-7). Existindo tal ramo, armazena-se o índice encontrado em *vaux* em *v*, e o apaga da lista *vaux*; inclui-se um novo ponteiro para o nó da árvore, indicando a presença de um novo filho para este nó; uma variável de controle indica que um ramo ativo foi encontrado (*Encontrado = verdadeiro/falso*); e a busca por um ramo termina (Figura 15 – Linhas 8-13).

A avaliação da variável de controle *Encontrado* tem o objetivo de controlar a recursividade do procedimento. No momento que seu valor for *verdadeiro* indica que um ramo ativo foi encontrado e que o índice avaliado na lista dinâmica *vaux* será inserido como filho do nó avaliado na árvore, e novamente o procedimento de busca por um novo índice/ramo terá início a partir do nó recentemente inserido na árvore; no caso do valor ser *falso* avança-se a busca comparativa para o próximo elemento da lista dinâmica *vaux* (Figura 15 – Linhas 14-21).

Quando toda uma busca comparativa é realizada para um nó em avaliação da árvore e a variável de controle *Encontrado* continua com seu valor inicial *FALSE* isso indica que para aquele nó em avaliação na árvore não se tem nenhum ramo de ligação com chave

fechada, e então esse passo da recursividade é finalizado e retorna-se para a linha de execução onde foi acionado, no seu nó pai, que continua a busca comparativa onde estava.

As comparações (Figura 15 – Linha 7) são sempre realizadas entre o índice armazenado no nó avaliado da árvore e um elemento da lista dinâmica, e permitem que seja verificado se este par de índices possui um ramo de ligação com chave fechada no vetor de ramos *Node*. O elemento da lista dinâmica avalia uma próxima posição somente após verificar todo o vetor *Node* (Figura 15 – Linha 21) e não ter encontrado nenhum ramo que possua os índices contidos no nó da árvore e na lista dinâmica *vaux*.

No Anexo D apresenta ilustrações passo a passo, com descrição de cada estágio, a forma com que se constrói uma solução radial para uma árvore dinâmica utilizando um sistema elétrico fictício com 10 barras e 12 ramos.

4.3.4 Perdas Elétricas de uma Solução

O cálculo do fluxo de potência para uma configuração radial de um sistema elétrico possui dois procedimentos principais (*backward/forward*) onde cada um possui tarefas específicas, e tais procedimentos são realizados até que as perdas calculadas, para cada percurso realizado pelo mesmo, sejam menores do que um limite adotado. Esse cálculo foi realizado para a solução construída na estrutura de árvore dinâmica, uma vez que as perdas elétricas são obtidas a partir de uma solução com características de árvore. Nota-se, também, que ambos os procedimentos para o fluxo de potência *backward-forward* possuem o mesmo comportamento nas duas formas de percurso em uma árvore dinâmica, pós-ordem e pré-ordem, respectivamente.

No procedimento *backward* a forma de percurso pós-ordem em uma árvore avalia um nó após todas as suas subárvores serem avaliadas, ou seja, um nó atualiza seu valor de corrente somente após todos os valores de corrente de suas subárvores serem atualizados. No procedimento *forward* a forma de percurso pré-ordem pode ser identificada uma vez que a avaliação de uma(s) subárvore(s) só é realizada após a avaliação do nó pai dessa(s) subárvore(s).

Foram implementadas um total de sete funções para a solução do fluxo de potência. Duas funções executaram o cálculo da corrente, para os valores real e imaginário

(*Tree_pos_corrente_imag* e *Tree_pos_corrente_real*); duas para calcular os valores das tensões, para os valores real e imaginário (*Tree_pre_tensao_imag* e *Tree_pre_tensao_real*); duas funções para calcular as perdas elétricas ativa e reativa do sistema (*Perdas_Ativas* e *Perdas_Reativas*); e uma última função (*FCR*) que organiza a ordem de chamada dessas funções mencionadas anteriormente para que o cálculo de fluxo de potência ocorra até convergir para um determinado limite. Na Figura 16 apresenta-se o algoritmo utilizado para o cálculo de ambos componentes da corrente, real e imaginária, pois mesmo utilizando termos diferentes para o cálculo de cada componente o procedimento algorítmico é o mesmo.

Figura 16 - Algoritmo para o procedimento de cálculo das correntes para o algoritmo de fluxo de potência.

1	Declaração de variáveis de controle
2	Percorrer e avaliar cada ponteiro do nó da árvore em avaliação
3	Se (ponteiro = <i>NULO</i>)
4	Retornar 0
5	Senão
6	Armazenar o índice de barra do nó-filho do nó em avaliação em <i>v1</i>
7	Encontrar o ramo ativo em <i>Node</i> que possua como par de índices de barra o índice da barra do nó em avaliação e de seu nó-filho <i>v1</i>
8	Calcular o valor da corrente <i>i_real/i_imag</i> do nó corrente
9	Atualizar o valor das correntes <i>I_real/I_imag</i> como somatório <i>soma</i> de <i>i_imag/i_real</i> repassando o procedimento da linha 2 para o nó-filho
10	Se (ponteiro+1 = <i>NULO</i>)
11	Retorne <i>soma</i>

Fonte: Próprio autor

Encontrado o ramo, calculam-se as correntes *i_real* e *i_imag* sem estarem atualizadas pelas correntes dos ramos ativos subsequentes (Figura 16 – Linha 8). Em seguida, realiza-se o procedimento de atualização (somatório) das correntes dos ramos, pelas variáveis *I_real* e *I_imag*, acionando-se novamente o procedimento de avaliação dos nós da árvore da Figura 16 - Linha 2 (Figura 16 – Linha 9) somando-se o resultado obtido anteriormente em *i_real* e *i_imag*. Quando o próximo ponteiro, do vetor que está sendo atualmente avaliado for *NULO*, se retorna o somatório final de uma dada componente da corrente para a sua respectiva variável em um determinado elemento do vetor *Node* (Figura 16 – Linhas 10-11).

Explicando em outras palavras, calculam-se as correntes de cada barra partindo do nó raiz (subestação) até os nós folha (barras terminais), e recursivamente, percorre-se a árvore do sistema percorrendo-se dos nós folha ao nó raiz, atualizando os valores de corrente de um nó

pai pela soma das correntes de seus nós filhos. O algoritmo para essa estratégia pode ser visualizado pela Figura 17.

Figura 17 - Algoritmo para o procedimento de cálculo das tensões para o procedimento de fluxo de potência.

1	Declaração de variáveis de controle
2	Percorrer e avaliar cada ponteiro do nó da árvore em avaliação
3	Se (ponteiro = <i>NULO</i>)
4	Retorne
5	Senão
6	Armazenar o índice de barra do nó-filho do nó em avaliação em <i>v1</i>
7	Encontrar o ramo de ligação em <i>Node</i> que possua com par de índices de barra o índice da barra do nó em avaliação e de seu nó-filho <i>v1</i>
8	Calcular o valor das tensões <i>V_real/V_imag</i> do nó corrente
9	Repassar o procedimento a partir do Linha 2 para o nó-filho

Fonte: Próprio autor

De forma semelhante ao cálculo das correntes, ocorre o cálculo das tensões do sistema para ambas as componentes, real e imaginário, sem a necessidade do somatório, uma vez que os valores de tensão de nó (barra) são replicados aos seus filhos (barras conectadas). Novamente, mesmo utilizando termos diferentes para o cálculo de cada componente o procedimento algorítmico é o mesmo.

Dessa forma o algoritmo avalia para cada ponteiro (ramo) em cada nó (Figura 17 – Linha 2). No caso de um ponteiro ser *NULO*, a função retorna para o passo anterior, onde foi chamada e continua sua execução a partir daquele ponto nó (Figura 17 – Linha 3-4). No momento em que um ponteiro indica um nó-filho, o índice da barra do nó filho é armazenado na variável *v1*, e assim como no cálculo das correntes, busca-se no vetor de ramos *Node*, um índice de um ramo que possua como índices de barras os índices dos nó/nó-filho sendo avaliados (Figura 17 – Linhas 5-7).

Calculam-se, então, as componentes de tensão real e imaginário para a barra com o índice *v1* do nó-filho (Figura 17 - Linha 8) e aciona a mesma função a partir da Linha 2 para o nó-filho. Em outras palavras, parte-se do nó configurado como raiz da árvore, então, repassa-se seus valores de tensão para os nós filhos que atualizam seus valores de tensão, que repassam novamente para seus filhos em um processo contínuo e repetitivo até que se alcancem os nós folha (barras terminais).

Para efetuar o cálculo das perdas elétricas, duas funções, praticamente idênticas, foram implementadas, a *Perdas_Ativas* e a *Perdas_Reativas* que calculam as perdas. Na Figura 18 ilustra-se o procedimento algorítmico para o cálculo das perdas elétricas.

Figura 18 - Algoritmo para o procedimento de cálculo das perdas elétricas para o procedimento de fluxo de potência.

1	Declaração de variáveis de controle
2	Iniciar $p_{at} = 0$ e $p_{reat} = 0$
3	Percorrer o vetor de ramos <i>Node</i> , e avaliar cada elemento
4	Se ($Node[i].chave = 1$)
5	Atualizar p_{at} e p_{reat}
6	Apresentar p_{at} e p_{reat}

Fonte: Próprio autor

A única diferença entre as duas funções de cálculo das perdas elétricas ativas e reativas são os elementos utilizados para obtenção de seus valores, sendo a reatância utilizada para o cálculo das perdas reativas, e a resistência utilizada para o cálculo das perdas ativas. Ambos os fatores são informados na impedância dos ramos.

O procedimento não percorre a árvore, mas o vetor de ramos *Node*. Iniciam-se as variáveis que ao final do processo possuirão os resultados devidos dos fatores de potência ativa e reativa, p_{at} e p_{reat} , respectivamente (Figura 18 – Linha 2). O algoritmo faz uma varredura no vetor *Node* acessando os dados necessários de seus elementos somente quando encontra um elemento que tenha a configuração da chave do ramo como fechada ($chave = 1$) (Figura 18 – Linhas3-4).

Encontrado um elemento válido em *Node*, as variáveis p_{at} e p_{reat} atualizam seus valores, utilizando os valores atualizados das componentes de corrente real e imaginária (I_{real} e I_{imag}) e os valores de reatância e resistência, adequadamente para cada tipo de potência (Figura 18 – Linha 5). Todas as variáveis necessárias para a atualização do valor de potência (resistência, reatância, componentes da corrente atualizadas) encontram-se no elemento em *Node* avaliado a cada iteração. Ao final do procedimento os resultados restantes nas variáveis p_{at} e p_{reat} são apresentados (Figura 18 – Linha 6).

A última função implementada, a *FCR*, agrega e agrupa todas as anteriores, posicionando-as e acionando-as adequadamente, e calculando a convergência das perdas calculadas a fim de verificar se o procedimento deve ser repetido e calcular as novas perdas,

ou se o sistema atingiu a convergência respeitando o erro limite. Na Figura 19, apresenta-se o algoritmo para a função *FCR*.

Figura 19 - Algoritmo que agrega as funcionalidades necessárias para o cálculo do fluxo de potência de uma solução radial.

```

1   Declaração de variáveis de controle
2    $P_{per\_1} = 0.0, \delta_P = 1.0$ 
3   Enquanto ( $\delta_P > \lim$ )
4       Após a primeira iteração
5            $Tree\_pre\_tensao\_real$ 
6            $Tree\_pre\_tensao\_imag$ 
7       Percorrer o vetor de ramos  $Node$ 
8            $Node[i].I\_real = 0$ 
9            $Node[i].I\_imag = 0$ 
10       $Tree\_pos\_corrente\_real$ 
11       $Tree\_pos\_corrente\_imag$ 
12       $p\_at = Perdas\_Ativas$ 
13       $p\_reat = Perdas\_Reativas$ 
14       $P_{per\_2} = p\_at$ 
15       $\delta_P$  recebe o maior valor de ( $P_{per\_2} - P_{per\_1}$ )
16      Se ( $\delta_P > \lim$ )
17           $P_{per\_1} = P_{per\_2}$ 

```

Fonte: Próprio autor

Inicia-se o algoritmo com as perdas calculadas $P_{per_1} = 0.0, \delta_P = 1.0$, e a variável \lim recebendo o limite de convergência (erro permitido) do arquivo (Figura 19 – Linha 2). O algoritmo, a cada iteração (Figura 19 – Linha 3), recalcula os valores de tensão e corrente para ambas suas componentes real e imaginária (Figura 19 – Linhas 4-11) e obtém os valores das perdas ativas e reativas (Figura 19 – Linhas 12-13). A estratégia utilizada para este trabalho levou em consideração somente as perdas ativas do sistema elétrico para o cálculo da convergência.

A perda ativa calculada a cada iteração, P_{per_2} , (Figura 19 – Linhas 14), é confrontada com a perda ativa armazenada em P_{per_1} , e o maior valor entre as duas é armazenado na variável δ_P (Figura 19 – Linha 15). Realiza-se, então uma verificação para apurar se δ_P ultrapassou o valor do limite de convergência \lim (Figura 19 – Linha 16). No caso de a verificação ser positiva, atualiza-se P_{per_1} para a próxima iteração; caso seja negativa se passa para a próxima iteração sem tal alteração (Figura 19 – Linha 17). Todo o procedimento se repete até que a convergência seja alcançada (Figura 19 – Linha 3).

4.3.5 Estrutura e Tratamento de Vizinhança

Todo processo heurístico e meta-heurístico necessita de uma metodologia que possa controlar a geração de novas soluções a partir de uma dada solução, explorando o espaço de busca de soluções. Esse procedimento é conhecido como vizinhança, ou seja, são as soluções do espaço de busca geradas a partir de uma única solução por meio de estratégias definidas. Tais estratégias são movimentos realizados com os elementos da solução podendo ser troca de um elemento com outro, 2-opt, 3-opt, troca de conjunto de elementos, entre outros. O objetivo é a geração de soluções com diferentes configurações.

Ressalta-se que, em um espaço de busca de soluções, muitas delas podem vir a ser muito próximas, ou seja, com configurações que se diferenciam por um ou dois elementos somente, podendo trazer pequenas melhorias para uma determinada solução, percorrendo-se de forma lenta as vizinhanças a cada iteração no desejo de se alcançar a melhor solução. Isso acabaria por produzir muitas buscas (espaços de busca visitados) podendo consumir muito tempo de execução.

Soluções para tratar esse problema podem ser vistas na BT, por exemplo, nas estratégias de frequência, diversificação e intensificação. Tais estratégias atuam no conjunto das soluções vizinhas de um espaço de busca, modificando-as, forçando que certas soluções “saiam” do espaço de busca atual com a finalidade de tentar buscar novos espaços de busca mais interessantes.

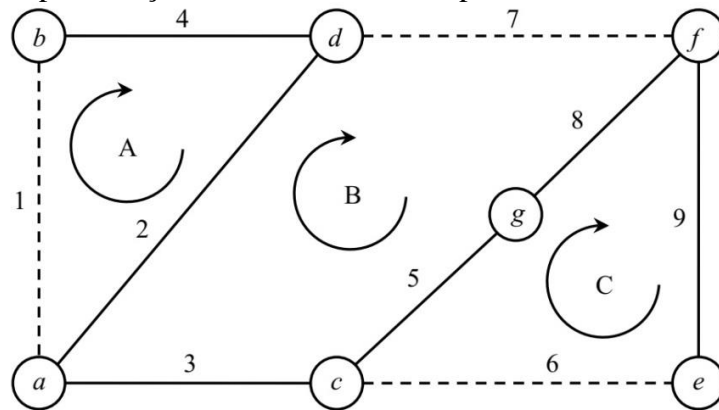
A estrutura de vizinhança utilizada neste trabalho deve respeitar a manutenção da radialidade de uma configuração proposta. A proposta utilizada segue os mesmos preceitos de Zhang *et al.* (2007) e Possagnolo (2015). Basicamente, para cada solução corrente se avaliam seus ciclos internos independentes, e para cada ciclo se avalia uma permuta das características entre uma chave fechada e a chave aberta que perfaz o ciclo, para todas as chaves fechadas. Todas as soluções geradas serão radiais, não existindo a necessidade de tratamento para soluções não radiais.

Para exemplificar o procedimento, adotar-se-á a Figura 20 para representar a configuração de uma solução inicial para um sistema de distribuição de energia elétrica com topologia radial, contendo 7 barras (*a-g*), um total de 9 ramos, e a barra *a* indicando a subestação do sistema. Para a configuração apresentada os ramos ativos 2, 3, 4, 5, 8, 9 apresentam chave fechada, e os ciclos internos independentes são representados pelas malhas

A, B e C. A representação da árvore para o sistema adotado na Figura 20 encontra-se apresentada na Figura 21.

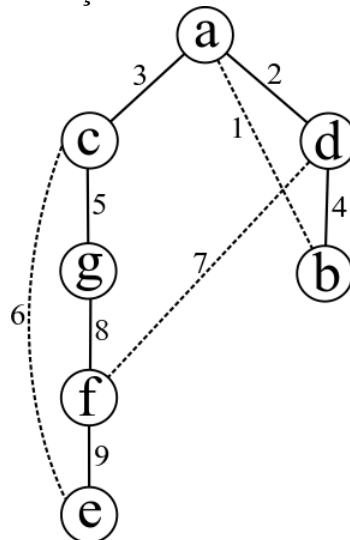
Seguindo a estrutura de vizinhança adotada, para cada malha, se fecha a chave do ramo que perfaz uma malha fechada e se abre outra, dentro da mesma malha, e fica mantida a radialidade da solução. Quando terminado o processo de geração de soluções vizinhas de uma malha, passa-se para a geração de soluções da outra malha, retornando a configuração dos ramos da malha anterior, assim como se encontram na solução inicial.

Figura 20 - Representação de ciclos internos dependentes de uma solução radial.



Fonte: Retirada de Possagnolo 2015.

Figura 21 - Representação em árvore do sistema da Figura 20.



Fonte: Próprio autor

Por meio desse método a malha A irá apresentar duas configurações vizinhas, a malha B, quatro configurações e a malha C, três configurações, totalizando nove soluções vizinhas. Na Tabela 2 apresentam-se, de forma mais clara, as configurações geradas para cada malha avaliada, denotando os ramos com chave fechada e aberta de cada solução.

O procedimento ocorre em duas etapas. Uma delas como um processo matricial que constrói uma matriz de ciclos, M_{ciclos} , que conterà as configurações de chaves de cada malha que se caracterizou um ciclo interno independente. A matriz possui uma quantidade de linhas que representa a quantidade de malhas, e uma quantidade de colunas que representa o total de ramos do sistema.

Tabela 2 - Representação das soluções geradas em uma determinada vizinhança.

Malha	Chave a ser fechada	Chave a ser aberta	Soluções vizinhas	
			Chaves Fechadas	Chaves Abertas
A	1	2	1,3,4,5,8,9	2,6,7
		4	1,2,3,5,8,9	4,6,7
B	7	2	3,4,5,7,8,9	1,2,6
		3	2,4,5,7,8,9	1,3,6
		5	2,3,4,7,8,9	1,5,6
		8	2,3,4,5,7,9	1,6,8
C	6	5	2,3,4,6,8,9	1,5,7
		8	2,3,4,5,6,9	1,7,8
		9	2,3,4,5,6,8	1,7,9

Fonte: Próprio autor

O segundo processo é uma função acionada pelo procedimento da matriz de ciclos que preenche suas linhas uma a uma, por meio da construção de um vetor de ciclos, vet_{ciclos} . Esse vetor, a cada iteração, possui as informações de chaves que perfazem os ciclos de cada malha do sistema, e repassa para a matriz de ciclos M_{ciclos} . Em outras palavras, cada linha da matriz de ciclos foi copiada de uma iteração do preenchimento do vetor de ciclos.

O vetor de ciclos é uma matriz com duas linhas, comportando em suas colunas *ramos* elementos. A construção do vetor de ciclos é realizada por meio da leitura da árvore da solução corrente T_{Node} , identificando os ramos com chave aberta pertencentes a cada ciclo, e então percorrendo a árvore na busca de ambos os índices de barra desses ramos. Em cada linha do vetor de ciclos encontram-se os índices de barras percorridos para cada índice de barra do ramo com chave aberta. Em seguida refina-se o vetor para não acessar os índices de

barras que não fazem parte do ciclo. Para cada dois elementos consecutivos de cada linha do vetor de ciclos busca-se no vetor de ramos da solução corrente N , o status de chave e atualiza este valor no mesmo índice de ramo na linha da matriz de ciclos M_ciclos em avaliação.

Após o procedimento de construção da matriz de ciclos, tem-se em cada uma de suas linhas a representação das chaves dos ramos de ligação para cada malha do sistema. Avalia-se então cada linha da matriz e geram-se as soluções vizinhas, para cada linha, fechando a chave de valor 2 e abrindo as chaves de valor 1, uma a uma. Cada permuta configura uma solução radial nova para o sistema e alteram-se os devidos valores no vetor de ramos $Node$.

O algoritmo que apresenta o procedimento de construção da matriz de ciclos, bem como a construção do vetor de subciclos que perfaz a estruturação de cada linha da matriz de ciclos, pode ser visualizado na Figura 22.

Figura 22 - Algoritmo dos procedimentos de construção de M_ciclos e vet_ciclos .

1	Iniciar M_ciclos
2	Percorrer as linhas da matriz de ciclos M_ciclos e para cada linha i
3	Percorrer o vetor de ramos N da solução em avaliação
4	Iniciar vet_ciclos
5	Se (chave == aberta)
6	Salvar os índices $v1$ e $v2$ da barra do ramo aberto
7	Percorrer a árvore em busca de $v1$ e armazenar a ordem dos índices de barra percorridos na primeira linha de vet_ciclos
8	Percorrer a árvore em busca de $v2$ e armazenar a ordem dos índices de barra percorridos na segunda linha de vet_ciclos
9	Percorrer vet_ciclos e encontrar as posições j,k entre as linhas 1 e 2 que possuem a mesma informação
10	Percorrer cada linha de vet_ciclos a partir da coluna j,k respectivamente
11	Para cada linha, verificar a chave do ramo entre 2 posições consecutivas com a chave deste mesmo ramo no vetor de ramos N
12	Informar o status desta chave em M_ciclos no mesmo índice do ramo

Fonte: Próprio autor

O início do procedimento se dá por meio da inicialização da matriz de ciclos M_ciclos zerando todos os seus elementos. A matriz possui como quantidade de linhas a mesma quantidade de ciclos do sistema (quantidade de chaves abertas de configuração radial da solução), e possui como quantidade de colunas a mesma quantidade total de ramos do sistema (Figura 22 – Linha 1).

Inicia-se o preenchimento de cada linha da matriz de ciclos com os status de chaves de cada ramo de cada ciclo encontrado (Figura 22 – Linhas 2-12).

Para cada linha i de M_{ciclos} em avaliação, inicializa-se o vetor de ciclos vet_{ciclos} zerando todos os seus elementos e percorre-se o vetor de ramos da solução corrente N pelos ramos com chave aberta que pertencem a um ciclo (Figura 22 – Linhas 2-5).

Para cada ramo encontrado com chave aberta armazenam-se em $v1$ e $v2$ os índices de barra do ramo encontrado. Em seguida percorre-se a árvore dinâmica duas vezes em busca de cada índice de barra $v1$ e $v2$, cessando sua busca quando os índices são encontrados. Cada linha do vetor de ciclos é preenchida com os índices de barra percorridos em cada busca na árvore dinâmica. Isso faz com que cada linha do vetor de ciclos possua o “caminho” percorrido, por meio dos índices de barra, da raiz da árvore até cada um dos índices do ramo com chave aberta selecionado em N (Figura 22 – Linhas 6-8).

Para exemplificar a configuração do vetor de ciclos para um ciclo, escolheu-se o ramo de índice 7 da Figura 21, para que o ciclo correspondente a ele possa ser construído. Assim, as variáveis $v1$ e $v2$ indicam, respectivamente, os índices de barras d e f . Na Tabela 3 apresenta-se como o vetor de ciclos está configurado após seu preenchimento.

Tabela 3 - Preenchimento do vetor de ciclos.

Ramo de chave aberta escolhido		Vetor de ciclos				
Índice	Barras	Linha 1	D	a		
7	$v1 = d; v2 = f$	Linha 2	F	g	c	a

Fonte: Próprio autor

Nesse ponto, a depender do ciclo avaliado o vetor de ciclos, ainda não aparecerão os dados adequados do ciclo a ser avaliado, necessita-se de um tratamento específico pois poderão vir a existir muitos elementos iguais nas primeiras posições de ambas as linhas do vetor de ciclos. Em outras palavras, o “caminho” até a primeira barra pertencente ao ciclo em avaliação pode ser o mesmo. No caso da Tabela 3, isso não ocorreu.

Inicia-se uma busca no vetor de ciclos até que se encontre um valor idêntico entre as posições de ambas as linhas, e armazena este índice de posição em j para a primeira linha e em k para a segunda linha (Figura 22 – Linha 9). A partir destas posições os elementos restantes de ambas as linhas irão compor as barras do ciclo em questão.

Realiza-se, então, a leitura do vetor de ciclos para que, com essas informações, possa-se preencher a linha i de M_{ciclos} com os status de chaves atualizados de cada ramo. Para isso realiza-se a leitura de cada linha do vetor de ciclos a partir da posição j , e cada par consecutivo de elementos, ou barras, busca-se no vetor de ramos N o índice do ramo formado por estas barras, verifica-se seu status de chave e atualiza na linha i , no mesmo índice de ramo o status de chave adequado (Figura 22 – Linhas 10-12). Na Tabela 4 encontram-se as informações de cada par de elementos consecutivos (barras), e seus valores correspondente para a matriz de ciclos.

Tabela 4 - Ramos válidos contidos no vetor de ciclos.

	Vetor de ciclos		Ramo com chave aberta
	Linha 1	Linha 2	
Ramos	(a,d)	$(f,g), (c,g), (a,c)$	(d,f)
Valor	1	2	3

Fonte: Próprio autor

Os ramos encontrados na primeira e segunda linha do vetor de ciclos possuem valores 1 e 2, respectivamente, nos mesmos índices na linha i de M_{ciclos} . O ramo que apresenta a chave aberta do ciclo (ramo com barras $v1$ e $v2$) é designado com valor 3 no mesmo índice na linha i de M_{ciclos} . O restante dos elementos da linha i de M_{ciclos} continua com o valor zero. Essa forma de numeração para os ramos com a chave fechada serve para indicar quais ramos participaram de cada uma das linhas do vetor de ciclos. Na Tabela 5 ilustra-se a situação de como a linha i da matriz de ciclos encontra-se depois de corretamente preenchida com os devidos status de chaves dos ramos.

Tabela 5 – Status resultantes das chaves dos ramos da linha i de M_{ciclos} .

Linha i da Matriz de ciclos									
Posição	1	2	3	4	5	6	7	8	9
Status de chave	0	1	2	0	2	0	3	2	0

Fonte: Próprio autor

4.3.6 Estratégia de Redução de Vizinhança

Uma estratégia de redução de vizinhança é aquela que faz com que soluções em um determinado espaço de busca não sejam avaliadas devido a um determinado fator, fazendo com que as soluções de uma determinada vizinhança não sejam todas avaliadas. Isso permite uma redução do espaço de busca fazendo assim com que a heurística ou meta-heurística percorra os espaços de busca de forma mais rápida e com a mesma frequência.

A principal razão de uma estratégia de redução de vizinhança para o problema de RSDEE é a complexidade que o problema pode alcançar na quantidade de chaves para os ramos de um determinado sistema. Um sistema com poucas barras e muitas chaves possui uma maior complexidade do que um sistema com muitas barras e poucas chaves. Para este trabalho é considerado que todos os ramos possuem uma chave, o que significa que cada sistema possui o máximo de complexidade possível. Isso faz com que a quantidade de soluções possíveis de serem geradas seja ampliada (árvores geradoras) de forma exponencial.

Na Tabela 6, ilustram-se as dimensões do espaço de busca para os sistemas utilizados neste trabalho abordando diferentes formas de codificação do problema. A Tabela 6, nr representa a quantidade de ramos do sistema, na qual é verificado que $\frac{na}{2^{nr}}$ é o número de combinações possíveis desta proposta de trabalho e $C(nr, nb - 1)$ é o número de combinações de nr elementos, tomando $nb - 1$, onde nb representa a quantidade de barras do sistema.

Tabela 6 - Dimensão do espaço de soluções para diferentes sistemas e formas de codificação

Sistema	nr	2^{nr}	Nro. De árvores geradoras (na)	$C(nr, nb - 1)$	$\frac{na}{C(nr, nb - 1)}$	$\frac{na}{2^{nr}}$
14 barras	16	65536	190	560	0,3393	$2,8991 \cdot 10^{-3}$
34 barras	37	$1,3744 \cdot 10^{11}$	50751	435897	0,1164	$3,6925 \cdot 10^{-7}$
84 barras	96	$7,9228 \cdot 10^{28}$	$3,5196 \cdot 10^{11}$	$4,0363 \cdot 10^{15}$	$8,7199 \cdot 10^{-5}$	$4,4423 \cdot 10^{-18}$
136 barras	156	$9,1344 \cdot 10^{46}$	$2,2686 \cdot 10^{18}$	$5,4351 \cdot 10^{25}$	$4,1740 \cdot 10^{-8}$	$2,4835 \cdot 10^{-29}$
415 barras	473	$7,2116 \cdot 10^{142}$	$9,6586 \cdot 10^{51}$	$1,0832 \cdot 10^{76}$	$8,5900 \cdot 10^{-25}$	$3,8150 \cdot 10^{-91}$

Fonte: Retirada e adaptada de (POSSAGNOLO, 2015)

A terceira coluna da Tabela 6, indicada por 2^{nr} , se refere à estratégia de implementação binária 0-1 adotada para este trabalho, onde as informações dos ramos de um sistema são informados por meio de um vetor de comprimento nr (vetor *Node*), em que uma

chave fechada é representada pelo valor 1 (um) e uma chave aberta é representada pelo valor 0 (zero). A quinta coluna da Tabela 5, indicada por $C(nr, nb - 1)$, representa a estratégia de codificação adotada por, Possagnolo, 2015, normalmente utilizada na literatura, a qual armazena somente os índices dos ramos com chave aberta. A quarta coluna da Tabela 5, indicada por na , apresenta o número de árvores geradoras que podem ser construídas para cada sistema, ou seja, mostra a quantidade máxima de configurações radiais que um determinado SDEE possui. Esse valor é calculado pelo Teorema da Matriz-Árvore.

Nota-se que na proposta de implementação adotada neste trabalho, a codificação binária (2^{nr}), possui um espaço de busca muito maior do que a proposta implementada por Possagnolo, 2015 ($C(nr, nb - 1)$). Para ambas as estratégias, para cada solução ainda verifica-se a factibilidade, por meio da avaliação dos limites de tensão e a manutenibilidade da radialidade que é realizada automaticamente, pela estrutura de vizinhança adotada.

As colunas 6 e 7 da Tabela 6 representam as relações entre o número de árvores geradoras possíveis e o número total de soluções possíveis, para as técnicas de Possagnolo, 2015 e adotada para este trabalho. A coluna 7 mostra resultados bem menores do que os da coluna 6 reforçando o que Possagnolo, 2015 informou esclarecendo que uma estratégia de codificação que gere somente configurações com topologias radiais deve ser utilizada, estratégia, esta, adotada neste trabalho.

Assim, reforça-se a necessidade de um procedimento de redução de vizinhança para o problema de RSDEE, pois dependendo da estratégia de vizinhança adotada o espaço de busca torna-se muito grande e o custo computacional para avaliar todas as soluções deste espaço de busca torna-se computacionalmente moroso.

A proposta de redução de vizinhança adotada neste trabalho surgiu a partir da implementação do GRASP por meio da utilização dos fluxos de potência aparente dos ramos em uma rede malhada armazenado no vetor FC_{ini} . Como a proposta de construção das soluções no GRASP aborda a estratégia de utilização da média aritmética do somatório dos fluxos de potência aparente em uma rede malhada, e a solução de maior média tende a ser a solução com as menores perdas, utilizou-se dessa característica para implementar a técnica de redução de vizinhança.

A proposta consiste no somatório dos fluxos de potência aparentes armazenados no vetor FC_{ini} , dos ramos com status de chave fechada, e então calcula-se a média contabilizando todos os ramos do sistema, não somente os que ajudaram a contabilizar essa média.

Este procedimento é efetuado para a solução corrente e para cada solução vizinha gerada. Caso a média da solução vizinha for menor que 1% em relação à média da solução corrente, esta solução vizinha é então avaliada, ou seja, constrói-se sua árvore dinâmica, calcula-se seu fluxo de potência, analisa-se sua factibilidade, checa-se sua entrada no conjunto de soluções elite, e se esta é melhor do que a solução corrente ou a pior solução visitada.

A taxa percentual de verificação adotada foi a de 1%, pois foi a que apresentou os melhores resultados. O aumento da taxa traz um aumento na quantidade de soluções avaliadas, e a redução da mesma faz com que o caminho de soluções geradas nos espaços de busca não seja tão eficaz.

O fato ocorre devido à incerteza de que uma solução com um determinado valor alto de média do somatório do fluxo de potência aparente dos ramos fechados apresentarem uma melhor ou pior função objetivo. Assim, a taxa permite que soluções com estas médias 1% piores do que a média da solução corrente possam ser avaliadas.

5 TESTES E RESULTADOS

O conjunto de meta-heurísticas GRASP/BT/PR, juntamente com as novas estratégias de tratamento e armazenamento das soluções e a inovação da técnica de cálculo de fluxo de potência propostas neste trabalho para resolver o problema de RSDEE considerando um único nível de demanda foram implementadas em linguagem estruturada C, por meio da IDE *CodeBlocks*, utilizando o compilador MingW32, em um notebook com processador Intel Core i7-8750 de 2,2 GHz e 16 GB de memória RAM. Para validação e verificação da metodologia proposta foram realizados testes numéricos para sistemas de 14 barras, apresentado por Civanlar *et al.* 1988; 33 barras, apresentado por Baran e Wu 1989; 84 barras, apresentado por Chiou *et al.* 2005; 136 barras, apresentado por Mantovani *et al.* 2000, e 417 barras, apresentado por I. J. Ramirez-Rosado 1998.

Os resultados das técnicas propostas neste trabalho (T0) foram comparados com resultados obtidos por Isabella F. Feiteira; Leonardo H.Macedo; Rubén Romero, 2018 (T3); Possagnolo, 2015 (T4); Renato Cespedes G. 1990 (T2); Shirmohammadi, 1998 (T1). No Apêndice C se encontram as informações detalhadas dos arquivos utilizados para testes neste trabalho. As técnicas comparativas foram executadas com auxílio do programa MATLAB.

A utilização da solução inicial encontrada nos arquivos de dados para o problema de RSDEE, e a encontrada pela aplicação do algoritmo de PRIM, serve para comparar com a estratégia de aplicação do algoritmo GRASP e avaliar a qualidade das soluções fornecidas pelo algoritmo GRASP. O indicador de sensibilidade α do GRASP foi ajustado para 0,4 para todos os sistemas. No sistema de 417 barras utilizaram-se dois valores para o indicador de sensibilidade, sendo de 0,4, como já mencionado, e de 0,1. Explicações mais detalhadas serão discutidas nos próprios resultados obtidos.

Ainda, os resultados apresentados sobre a solução inicial obtida pelo GRASP, neste trabalho, são informados por meio da obtenção de uma solução, e de dez soluções, onde a melhor solução dentre as dez geradas passa a ser a solução inicial que será submetida ao procedimento de busca local pela BT. Na apresentação dos resultados, as figuras ilustradas que representam graficamente o sistema de distribuição elétrica contendo as barras e os ramos de ligação, as barras são identificadas por números inteiros e os ramos por números inteiros entre parênteses. Tais numerações são exatamente os índices armazenados nas devidas estruturas de dados utilizadas neste trabalho e comentadas nos capítulos anteriores. Ao final

deste capítulo serão discutidas as vantagens e desvantagens da aplicação do algoritmo proposto.

5.1 CONSTRUÇÃO DE UMA SOLUÇÃO

Nesta seção avalia-se o rendimento da técnica proposta para a construção de uma configuração radial em árvore dinâmica comparando-se com a técnica clássica de renumeração de barras, uma vez que ambas as técnicas são o pré-processamento da configuração radial em estudo para a posterior execução do algoritmo de fluxo de potência a fim de se encontrar as devidas perdas elétricas.

Para efetuar as medidas de tempo de execução cada teste foi executado 1, 10, 100 e 1000 vezes para uma mesma configuração de um SDEE radial. Isso é necessário porque alguns tempos envolvidos são tão pequenos que o método de obtenção do tempo de execução não é mensurável.

Na Tabela 7 apresentamos os tempos obtidos comparando-se o procedimento de construção e destruição de uma configuração radial de um SDEE em árvore dinâmica proposto neste trabalho, e o procedimento tradicional de renumeração de barras proposto por Shirmohammadi (1988). O cálculo do tempo de execução da construção e destruição da árvore dinâmica (alocação e desalocação dinâmica) torna-se necessário porque no caso de uma nova configuração radial de um SDEE for construída, essa árvore deverá ser reconstruída, enquanto que para o procedimento clássico simplesmente se renumera a ordem das barras.

Tabela 7 - Tempos de execução (ms) totais para o procedimento de construção e destruição da árvore dinâmica e o procedimento de renumeração de barras.

Sistemas	Tempos de Execução (ms)							
	1x		10x		100x		1000x	
	T0	T1	T0	T1	T0	T1	T0	T1
14	< 10 ⁻⁶	8,009	< 10 ⁻⁶	8,920	1,000	20,698	3,000	118,073
33	< 10 ⁻⁶	8,041	< 10 ⁻⁶	13,156	3,000	38,043	34,000	287,835
84	1,000	8,522	3,000	14,468	27,000	72,211	276,000	693,376
136	1,000	10,716	12,000	24,119	115,000	144,494	1152,000	1470,255
417	27,000	17,951	271,000	87,52	2692,000	812,764	26822,000	8924,094

Fonte: Próprio autor

Quando se observa a Tabela 7 de forma vertical, analisando os tempos de acordo com o tamanho da entrada (SDEE adotado), verifica-se a linearidade no crescimento do tempo para a proposta de T1, e a característica exponencial no crescimento do tempo de execução para a técnica proposta, que mesmo possuindo uma complexidade de tempo cúbica, apresentou melhor rendimento para sistemas de pequeno e médio porte. Observa-se que para todos os tempos avaliados, até o SDEE de 136 barras o algoritmo proposto possui menores tempos em comparação com a técnica de T1

Entretanto, para o SDEE de 417 barras, todos os tempos obtidos com a técnica de T1 foram menores que os tempos do algoritmo proposto. Isso indica que a complexidade do algoritmo de T1 tem maior sobrecarga de tempo que independe do tamanho da entrada, o que é significativo para a execução de pequenos problemas, diferente do procedimento proposto neste trabalho o qual depende da entrada.

O baixo rendimento da técnica proposta para sistemas de grande porte pode ser explicado pela grande carga de procedimentos recursivos codificados e deve ser melhorada em pesquisas futuras. Cada chamada recursiva gera uma sobrecarga de tempo e memória durante a execução, pois, para cada chamada, é necessário criar toda uma nova estrutura de memória para as variáveis utilizadas internamente naquela função chamada.

No caso de análise horizontal da Tabela 7, avaliando os tempos de execução para diferentes repetições de um mesmo sistema elétrico, ambas as técnicas apresentaram uma linearidade no crescimento dos tempos de execução, ou seja, ambas apresentaram uma característica proporcional de crescimento em seus tempos de execução conforme se aumentava a quantidade de iterações.

O fato do tempo crescer linearmente, ao multiplicarmos o número de execuções sucessivas indica que os procedimentos executados são estáveis do ponto de vista do uso do processador e da memória. A técnica de T1 apresentou tal característica a partir das 100 iterações, enquanto a técnica proposta apresentou a partir de uma iteração, mostrando uma maior estabilidade. Esse bom comportamento da técnica proposta faz com que se possa estimar o tempo total de execução de um algoritmo para n soluções previstas a partir do tempo de execução de somente uma solução para um determinado sistema elétrico.

5.2 CÁLCULO DO FLUXO DE POTÊNCIA

Nesta seção avalia-se o rendimento da aplicação da técnica proposta para o cálculo do fluxo de potência em um SDEE radial em árvore dinâmica de filhos variados em comparação com técnicas que utilizam a mesma estratégia *backward/forward* para o cálculo de fluxo de potência em SDEE radiais. As técnicas comparativas são as de Shirmohammadi (1988) e a de Renato Cespedes G. (1990).

Para cada sistema o teste foi executado somente para uma configuração de um SDEE radial, a mesma utilizada para os cálculos da Tabela 7 e, então, calculado seu fluxo de potência por meio dos procedimentos *backward/forward* se iterando 1, 10, 100 e 1000 vezes o algoritmo de fluxo de potência.

A necessidade de se executar o fluxo de potência inúmeras vezes se justifica pela característica intrínseca de problemas como reconfiguração e restauração de SDEEs que requerem que inúmeras configurações radiais sejam avaliadas. Os resultados dos tempos obtidos são apresentados na Tabela 8.

Tabela 8 - Tempos de execução (ms) para o fluxo de potência em árvore dinâmica.

Sistemas	Tempos de Execução (ms)											
	1x			10x			100x			1000x		
	T0	T1	T2	T0	T1	T2	T0	T1	T2	T0	T1	T2
14	0,000	5,889	6,065	0,000	6,575	6,159	0,000	12,855	12,844	4,000	85,029	76,742
33	0,000	5,489	5,827	0,000	7,093	7,034	1,000	18,339	17,546	16,000	119,927	119,606
84	0,000	6,380	5,682	1,000	7,583	7,255	9,000	24,293	23,019	85,000	188,789	188,983
136	0,000	5,829	5,668	2,000	7,755	7,604	22,000	27,912	27,507	214,000	229,788	224,026
417	2,000	8,171	6,111	17,000	11,018	11,762	168,000	62,068	61,787	1677,000	586,093	561,323

Fonte: Próprio autor

Ao se avaliar a Tabela 8 no sentido vertical, levando em consideração o tamanho da entrada, as técnicas T1 e T2 apresentam uma característica linear de crescimento nos tempos de execução, enquanto a técnica proposta cresce de forma exponencial, como esperado pela sua complexidade quadrática. Os tempos de execução obtidos para a técnica proposta de cálculo de fluxo de potência são menores do que os tempos das técnicas T1 e T2 para sistemas de pequeno e médio porte (14, 33, 84 e 136 barras).

A técnica de cálculo de fluxo de potência proposta perde rendimento para sistemas de grande porte devido à grande quantidade de procedimentos recursivos. Verifica-se que assim como o procedimento de renumeração de barras de T1, a complexidade do algoritmo de cálculo de fluxo de potência de T1 e T2 apresenta maior sobrecarga de tempo que independe do tamanho da entrada, o que é significativo para a execução de pequenos problemas, diferente do procedimento de fluxo de potência proposto que é dependente do tamanho da entrada.

Considerando, agora, a Tabela 8 em seu sentido horizontal, que avalia os tempos de execução para diferentes quantidades de iterações de um mesmo sistema elétrico, ambas as técnicas apresentaram uma linearidade em seus tempos de execução, ou seja, ambas apresentaram uma característica proporcional de crescimento em seus tempos de execução conforme se aumenta a quantidade de iterações, assim como verificado na Tabela 7.

Pode ser observado que para as técnicas T1 e T2 os tempos de execução não crescem com uma proporcionalidade regular, enquanto que para a técnica proposta os tempos de execução crescem de forma regular de acordo com a quantidade de repetições do algoritmo, o que indica uma melhor estabilidade computacional do algoritmo. Assim como a técnica avaliada na Tabela 1, os resultados da técnica avaliada na Tabela 8 permitem que se possa estimar o tempo total de execução de um algoritmo para o cálculo de fluxo de potência para n soluções a partir do tempo de execução de somente uma solução para um determinado sistema elétrico.

Diferentes configurações de SDEE radiais para um mesmo sistema elétrico não afetam os tempos de execução, pois o percurso na árvore dinâmica é estável e seus elementos são sempre acessados somente uma única vez, fazendo com que a diferença dos tempos de execução de um fluxo de potência para a árvore dinâmica não sejam para diferentes configurações de um mesmo SDEE radial, mas sim entre SDEE radiais de diferentes dimensões.

Diferentemente do que acontece na Tabela 7, na Tabela 8 o tempo obtido para uma execução do sistema de 417 barras com o algoritmo proposto é menor que o tempo obtido por T1 e T2 em uma iteração, já quando repetimos o procedimento (10x, 100x, 1000x) os tempos obtidos pelo algoritmo proposto são maiores que os tempos obtidos pela técnica de T2 e T3. Isso indica que o algoritmo proposto possui uma sobrecarga de tempo de execução que depende da entrada e fica mais significativo quando o procedimento é repetido mais de uma vez. Podemos associar isso ao procedimento de destruição da árvore, ou seja, a manipulação

da memória entre sucessivas execuções, que depende da entrada já que quanto maior o sistema, maior é o número de posições de memória para alocar e desalocar entre as repetições.

A redução de ciclos nesta proposta para um mesmo sistema elétrico reduziria o tempo do cálculo do fluxo de potência porque reduziria a quantidade de ramos presentes no vetor *Node* no momento de busca das informações necessárias. O aumento de ciclos não traria grandes aumentos para os tempos de execução para ambas as estratégias apresentadas neste trabalho. A redução ou aumento de ciclos afetaria somente a quantidade de elementos do vetor *Node*, pois novos ramos seriam inseridos. A estratégia de construção de um SDEE radial em árvore dinâmica teria pouco impacto em seu tempo de execução, pois a quantidade de ramos ativos para a representação de um SDEE radial não muda para um mesmo sistema elétrico, mas a busca no vetor *Node* pelo próximo elemento a ser inserido na árvore dinâmica teria um aumento ou redução em seu tempo de execução que dependeria da quantidade de ciclos removidos ou acrescentados. O procedimento *backward/forward* implementado também não apresentaria grandes impactos em seu tempo de execução porque a quantidade de elementos presentes na árvore dinâmica de um SDEE radial não muda e se acessa cada elemento da árvore dinâmica somente uma única vez, sem a necessidade do tratamento de ramos inativos.

5.3 PROCEDIMENTO BT/GRASP/PR PARA O PROBLEMA DE RSDEE

Nesta seção serão avaliados os resultados da aplicação do conjunto de heurísticas e meta-heurísticas Busca Tabu, GRASP e *Path Relinking* aplicados à resolução do problema de RSDEE. Os sistemas que serão avaliados, como mencionados anteriormente, serão de 14, 36, 84, 136 e 417 barras, conferindo testes para SDEE radiais de pequeno, médio e grande porte. Para cada sistema serão avaliadas as seguintes características:

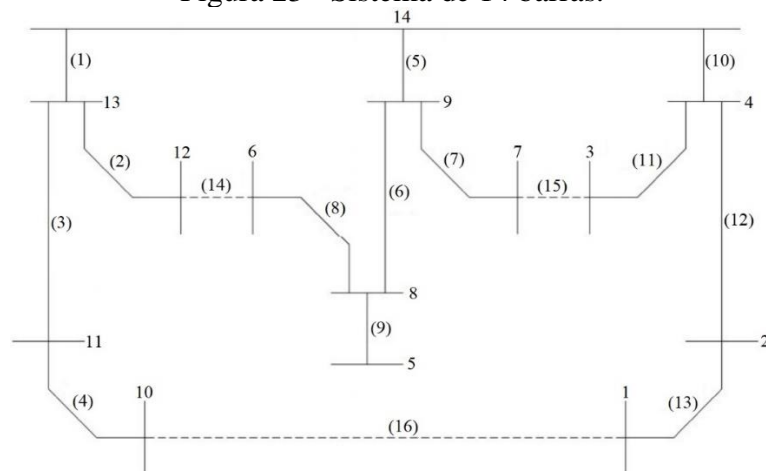
- A construção de uma configuração inicial, seja esta a configuração base, obtida por meio dos arquivos, ou das técnicas heurísticas de PRIM e GRASP. O rendimento da proposta será comparado com as técnicas de Possagnolo (2015) e de Isabella F. Feiteira; Leonardo H. Macedo; Rubén Romero (2018);
- A aplicação do conjunto de heurísticas e meta-heurísticas GRASP/BT/PR, propostos neste trabalho, ao problema de RSDEE. A técnica comparativa utilizada será a de Possagnolo 2015;

- Avaliação da técnica alternativa proposta para a redução de vizinhança SRV (Sem Redução de Vizinhança) e CRV (Com Redução de Vizinhança).

5.3.1 Sistemas de 14 barras

O sistema de 14 barras é apresentado por Civanlar *et al.* (1988). Possui um total de 14 barras e 16 ramos de ligação. A barra configurada como subestação possui índice 14, sua tensão base é de 23 kV com 1 p.u., as impedâncias fornecidas são por meio de cálculo percentual, a potência base é de 100.000 kVA e o erro limite de convergência é de 10^{-8} . A Figura 23 apresenta o diagrama do sistema de 14 barras com a configuração base provinda do arquivo, contendo três ramos de ligação com chaves abertas, cujos índices são 14, 15 e 16.

Figura 23 - Sistema de 14 barras.



Fonte: Adaptada de Civanlar *et al.* 1988 e Marinho 2017

Na Tabela 9 apresentam-se as estratégias utilizadas para obtenção de uma solução inicial para o problema RSDEE com 14 barras, juntamente com a técnica comparativa. São apresentadas as chaves abertas das melhores configurações obtidas, suas devidas perdas elétricas ativas e seus tempos de execução. As técnicas propostas neste trabalho estão indicadas pela referência T0, e a técnica comparativa pela referência T4 (ISABELLA F. FEITEIRA; LEONARDO H. MACEDO; RUBÉN ROMERO, 2018). Verifica-se um ganho significativo na obtenção de uma solução inicial pela técnica GRASP implementada neste

trabalho, no qual, o alto tempo obtido para as 10 soluções obtidas pelo GRASP será explicado.

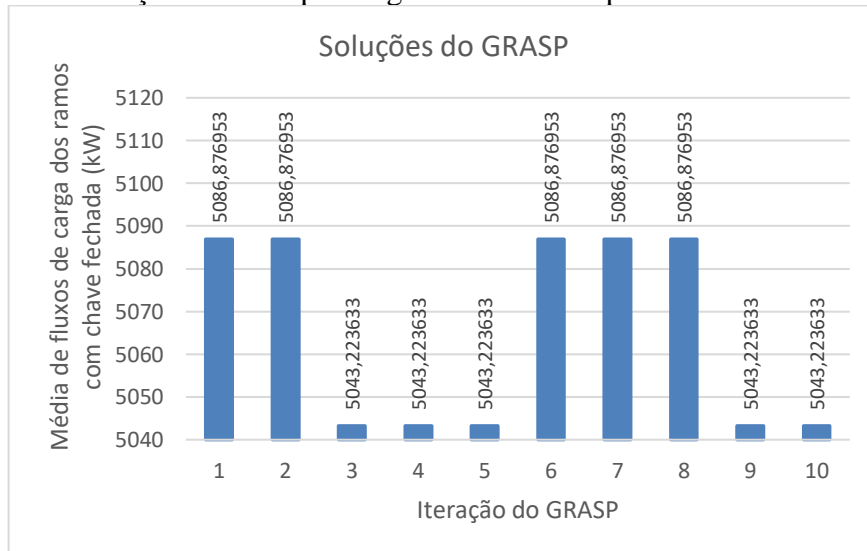
Tabela 9 - Estratégias de obtenção de uma solução inicial para o sistema de 14 barras.

Solução Inicial			
Configuração	Chaves abertas	Perdas Ativas (kW)	Tempo(s)
Base – T0	14, 15, 16	511,435606	$< 10^{-3}$
PRIM - T0	7, 8, 16	466,126855	$< 10^{-3}$
GRASP – Fase Construtiva - 1 solução - T0	7, 8, 16	466,126855	0,014
GRASP – Fase Construtiva – 10 soluções - T0	7, 8, 16	466,126855	3,126
GRASP – Fase Construtiva – T4 – 1 solução	7, 14, 16	483,868900	0,028

Fonte: Próprio autor

Na Figura 24 apresentam-se as configurações encontradas durante 10 iterações da fase construtiva do algoritmo GRASP para o problema de RSDEE com 14 barras. Devido à dimensão do sistema ser pequena, poucas configurações diferentes podem ser geradas, repetindo, assim, certas configurações. A configuração resultante, que será tratada como solução inicial pelo procedimento de BT, é a de maior média aritmética do somatório de fluxos de potência aparente dos ramos com chave fechada obtida pelo GRASP.

Figura 24 - Soluções obtidas pelo algoritmo GRASP para o sistema de 14 barras.



Fonte: Próprio autor

Na Tabela 10 apresentam-se os resultados obtidos pela BT e PR implementadas para cada estratégia de obtenção de solução inicial utilizada. A Tabela 11 traz as quantidades de soluções geradas pelas técnicas BT e PR e seus devidos tempos de execução em separado. Isso faz com que se visualize o rendimento de cada técnica em separado.

Tabela 10 - Solução final obtida para o sistema de 14 barras.

Solução Final Obtida				
Configuração		Chaves abertas	Perdas Ativas (kW)	Tempo(s)
SRV	Base/BT/PR	7, 8, 16	466,126855	24,996
	PRIM/BT/PR			1,890
	GRASP/BT/PR			5,626
CRV	Base/BT/PR			9,028
	PRIM/BT/PR			0,002
	GRASP/BT/PR			3,141
GRASP – T3			466,126700	0,116
PRIM/BVNS – T4				0,610
PRIM/VND – T4				0,455

Fonte: Próprio autor

Tabela 11 - Quantidades de soluções geradas/avaliadas e tempos de execução da BT e PR.

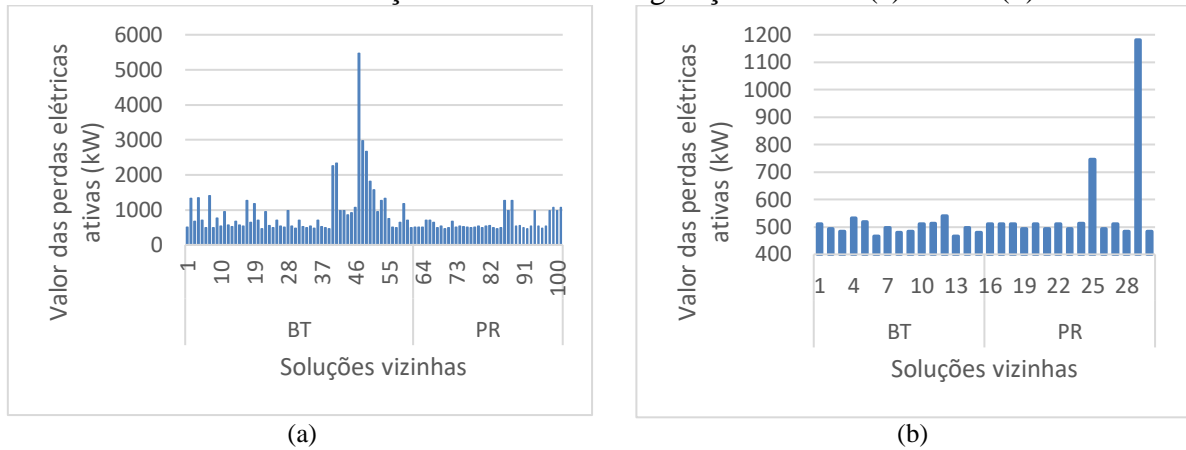
Técnicas Adotadas			Soluções geradas	Tempos de Execução (s)
	Solução Inicial	Heurísticas		
SRV	Base	BT	61	$< 10^{-3}$
		PR	40	24,996
	Prim/Grasp	BT	44	0,000
		PR	3	1,890
CRV	Base	BT	15	$< 10^{-3}$
		PR	15	9,026
	Prim/Grasp	BT	6	0,000
		PR	3	0,002

Fonte: Próprio autor

Verifica-se que a BT possui um rendimento extremamente superior ao PR, mesmo possuindo mais soluções geradas/avaliadas. Isso é devido à forma de implementação da técnica de BT e características da técnica de PR. Este resultado possui semelhanças em testes posteriores, e será melhor explicada no término deste capítulo.

Na Figura 25 apresentam-se os gráficos que representam todas as soluções geradas e avaliadas pelo procedimento de BT/PR implementado com a SRV (Figura 25(a)) e a CRV (Figura 25(b)) utilizando a solução base como a configuração inicial.

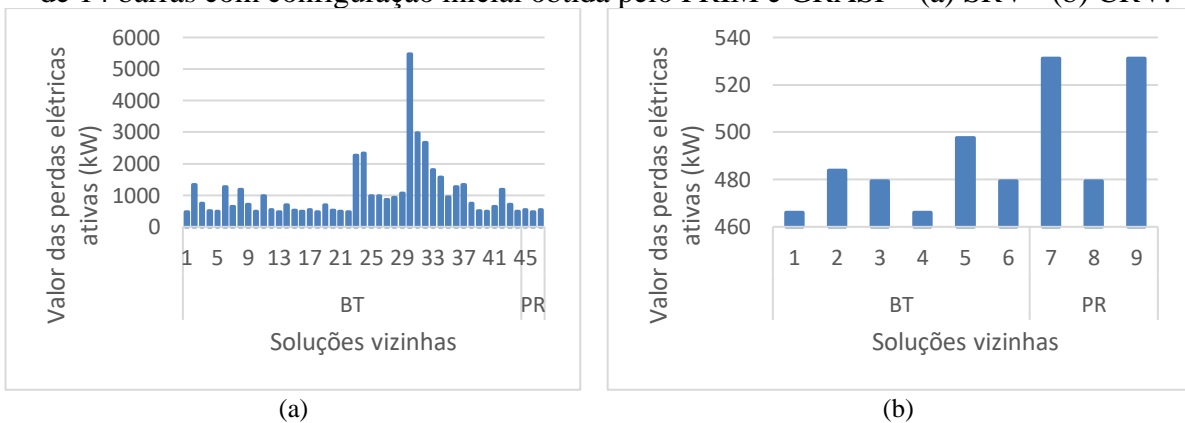
Figura 25 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 14 barras com solução base como configuração inicial - (a) SRV - (b) CRV.



Fonte: Próprio autor

Na Figura 26 apresentam-se os gráficos pertinentes ao histórico de soluções geradas e avaliadas para o conjunto BT/PR onde a solução inicial é obtida pelas técnicas de PRIM e GRASP. Também são apresentadas as soluções geradas com a SRV (Figura 26(a)) e com a CRV (Figura 26(b)). Uma vez que a solução inicial atingida é a mesma para ambas as técnicas PRIM e GRASP, o procedimento BT/PR obteve o mesmo comportamento.

Figura 26 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 14 barras com configuração inicial obtida pelo PRIM e GRASP - (a) SRV - (b) CRV.

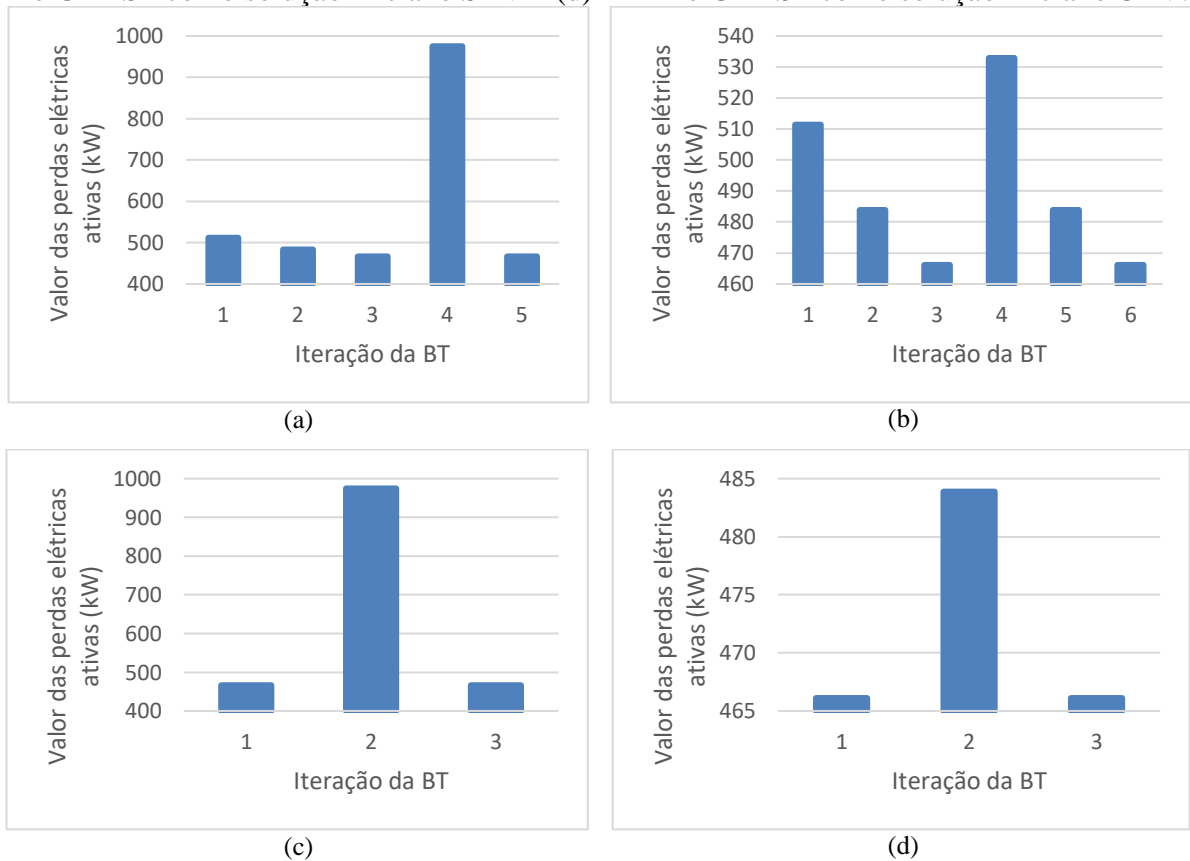


Fonte: Próprio autor

As hipóteses sobre os comportamentos dos gráficos das Figuras 25 e 26 são confirmadas, os mesmos apresentam uma redução na quantidade total de soluções geradas/avaliadas, provando que a técnica de redução de vizinhança alcançou seu objetivo.

Na Figura 27 ilustram-se as melhores soluções encontradas a cada iteração da BT que são repassadas para as próximas iterações como solução corrente. A Figura 27(a)-(b) representam as melhores soluções alcançadas pela BT, com a solução base adotada como configuração inicial, e SRV e CRV, respectivamente. Nas Figuras 27(c)-(d) representam-se as mesmas características que as Figuras 27(a)-(b), mas utilizando a configuração da solução inicial aquelas obtidas pelas técnicas de PRIM e GRASP.

Figura 27 - Melhores soluções obtidas no procedimento de busca local pela BT - (a) Solução base como solução inicial e SRV - (b) Solução base como solução inicial e CRV - (c) PRIM e GRASP como solução inicial e SRV - (d) PRIM e GRASP como solução inicial e CRV.



Fonte: Próprio autor

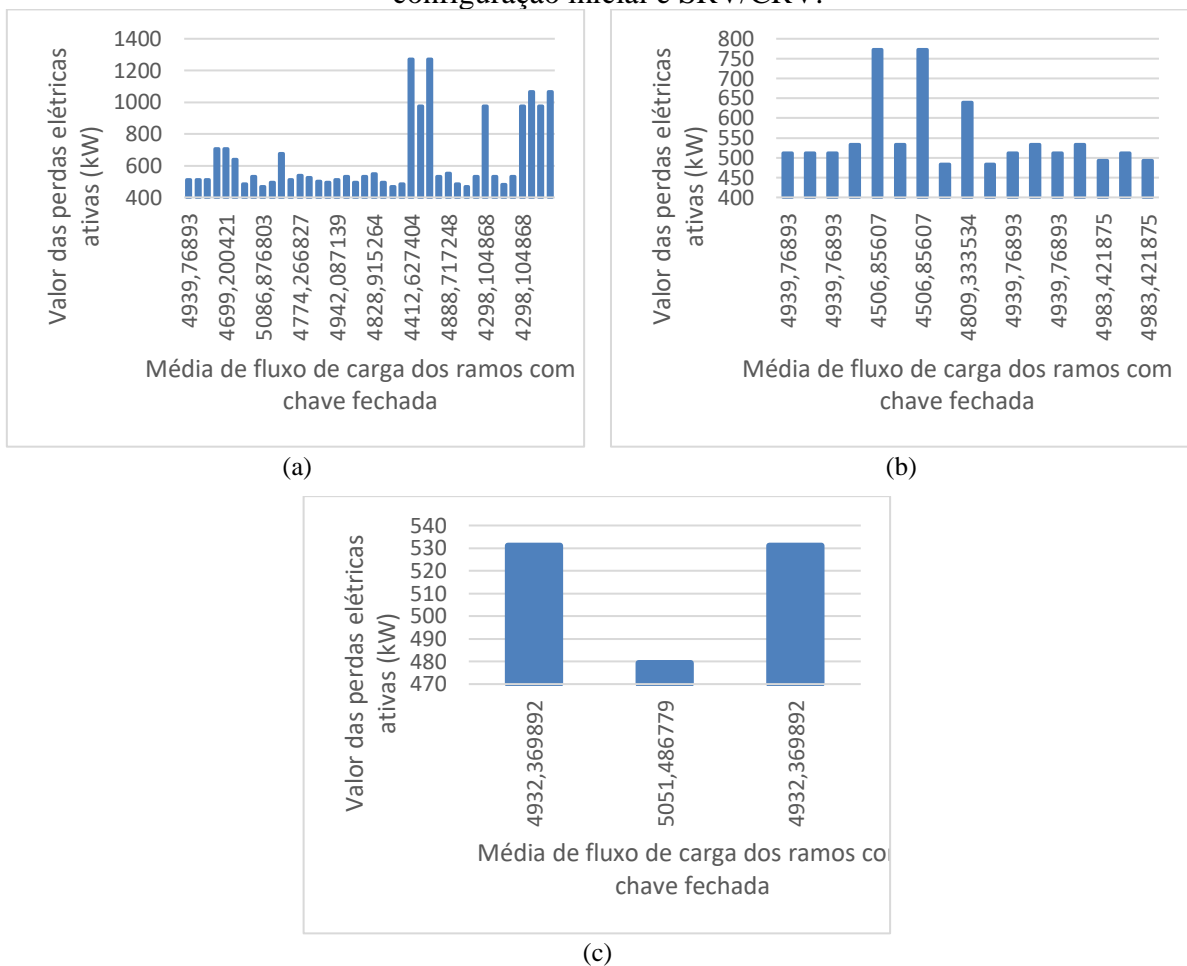
Na Figura 27 verifica-se que as melhores soluções de cada iteração para a BT pertinente a cada tipo de solução inicial escolhida melhoraram quando se aplicou a técnica de redução de vizinhança. Em outras palavras, o caminho das soluções durante o procedimento de busca local encontrou as melhores soluções, convergindo de uma forma mais rápida.

Na Figura 28 apresentam-se as soluções obtidas durante o procedimento de pós-otimização por meio do PR para cada tipo de estratégia de obtenção de uma solução inicial, e

com a SRV e CRV. As Figuras 28(a)-(b) representam os gráficos onde a solução base é adotada como configuração inicial e SRV e CRV, respectivamente. Na Figura 28(c) apresentam-se as mesmas características, mas com as configurações iniciais obtidas pelo PRIM e GRASP.

O caminho de soluções geradas quando aplicada a técnica de redução de vizinhança teve seu percurso por soluções melhores, alcançando em diversos momentos a melhor solução já encontrada. A Figura 28(c) demonstra que as soluções geradas foram as mesmas tanto com a utilização da técnica de redução de vizinhança, quanto sem o uso da mesma. Isso pode ser explicado pelos gráficos das Figuras 27(c)-(d) que geram poucas soluções de boa qualidade para que possam entrar na lista de soluções elite, as quais acabaram por serem as mesmas.

Figura 28 - Histórico de soluções do PR – (a) Solução base como configuração inicial e SRV – (b) Solução base como configuração inicial e CRV – (c) PRIM e GRASP como configuração inicial e SRV/CRV.

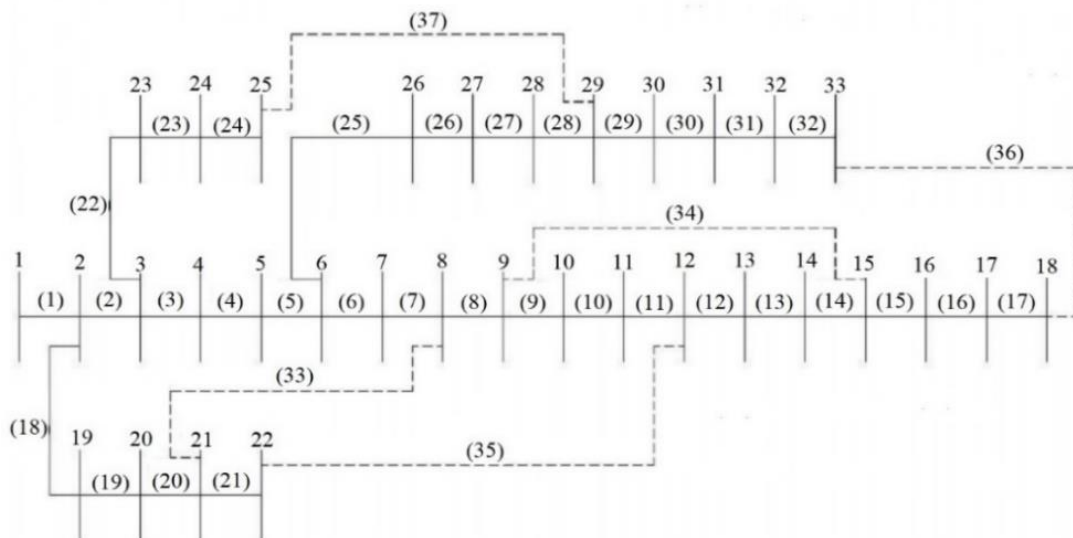


Fonte: Próprio autor

5.3.2 Sistemas de 33 barras

O sistema de 33 barras é apresentado por Baran and Wu (1989). Possui um total de 33 barras e 37 ramos de ligação. A barra configurada como subestação possui índice 1, sua tensão base é de 12,66 kV com 1 p.u., as impedâncias fornecidas são por meio de cálculo percentual, a potência base é de 1.000 kVA e o erro limite de convergência é de 10^{-8} . Na Figura 29 apresenta-se o diagrama do sistema de 33 barras com a configuração de uma solução base provinda do arquivo, contendo cinco ramos de ligação com chaves abertas, cujos índices são 33, 34, 35, 36 e 37.

Figura 29 - Sistemas de 33 barras.



Fonte : (MARINHO, 2017)

Na Tabela 12 apresentam-se as estratégias utilizadas para obtenção de uma solução inicial para o problema RSDEE com 33 barras, juntamente com a técnica comparativa. São apresentadas as chaves abertas das melhores configurações obtidas, suas devidas perdas elétricas ativas e seus tempos de execução. As técnicas propostas estão indicadas pela referência T0, e a técnica comparativa pela referência T4 (ISABELLA F. FEITEIRA; LEONARDO H. MACEDO; RUBÉN ROMERO, 2018). Vê-se novamente um ganho significativo na obtenção de uma solução inicial pela técnica GRASP aqui implementada. O alto tempo de CPU consumido para as dez soluções obtidas pelo GRASP será explicado ao final deste capítulo.

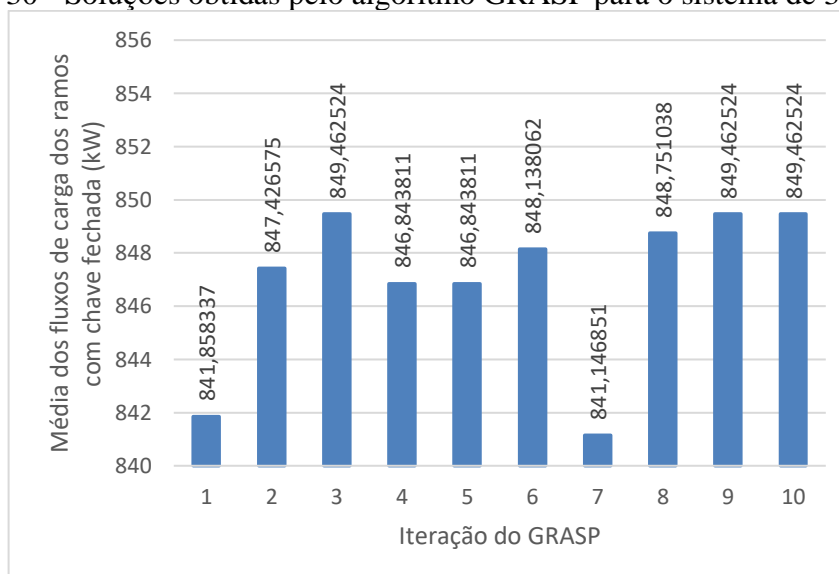
Tabela 12 - Estratégias de obtenção de solução inicial para o sistema de 33 barras.

Solução Inicial			
Configuração	Chaves abertas	Perdas Ativas (kW)	Tempo(s)
Base – T0	33, 34, 35, 36, 37	202,677086	$< 10^{-3}$
PRIM – T0	7, 10, 14, 28, 32	140,705839	$< 10^{-3}$
GRASP – Fase Construtiva – 1 solução - T0	7,10, 14, 28, 32	140,705839	0,003
GRASP – Fase Construtiva – 10 soluções - T0	7,10, 14, 28, 32	140,705839	3,035
GRASP – Fase Construtiva – 1 solução – T4	5, 7, 10, 14, 36	180,920200	0,035

Fonte: Próprio autor

Na Figura 30 apresentam-se as configurações encontradas durante dez iterações da fase construtiva do algoritmo GRASP para o problema de RSDEE com 33 barras. A dimensão desse problema proporcionou uma maior diversidade das soluções geradas. Um total de dez soluções foram geradas e avaliadas suas médias do somatório de fluxo de potência nos ramos com chaves fechadas. A solução com o maior valor desta média é transferida como solução inicial do procedimento de BT.

Figura 30 - Soluções obtidas pelo algoritmo GRASP para o sistema de 33 barras.



Fonte: Próprio autor

Na Tabela 13 apresentam-se os resultados obtidos pela BT e PR para cada estratégia de obtenção de solução inicial utilizada. Para a técnica proposta serão apresentados os dados obtidos com SRV e CRV adotada. Na Tabela 14 apresentam-se as quantidades de soluções

geradas pelas técnicas BT e PR e seus devidos tempos de execução em separado, o que faz com que se estude o rendimento de cada técnica em separado.

Tabela 13 - Solução final obtida para o sistema de 33 barras.

Solução Final Obtida				
Configuração		Chaves abertas	Perdas Ativas (kW)	Tempo(s)
SRV	Base/BT/PR	7, 9, 14, 32, 37	139,551342	107,164
	PRIM/BT/PR			15,483
	GRASP/BT/PR			29,060
CRV	Base/BT/PR			149,648
	PRIM/BT/PR			80,770
	GRASP/BT/PR			83,890
GRASP – T3			139,551300	0,178
PRIM/BVNS – T4				0,861
PRIM/VND – T4				0,564

Fonte: Próprio autor

Tabela 14 - Quantidades de soluções geradas/avaliadas, e tempos de execução das técnicas de BT e PR.

Técnicas Adotadas			Soluções geradas	Tempos de Execução (s)
	Solução Inicial	Heurísticas		
SRV	Base	BT	439	0,031
		PR	173	107,133
	Prim/Grasp	BT	258	0,021
		PR	25	15,454
CRV	Base	BT	229	0,010
		PR	241	149,618
	Prim/Grasp	BT	85	0,010
		PR	134	80,754

Fonte: Próprio autor

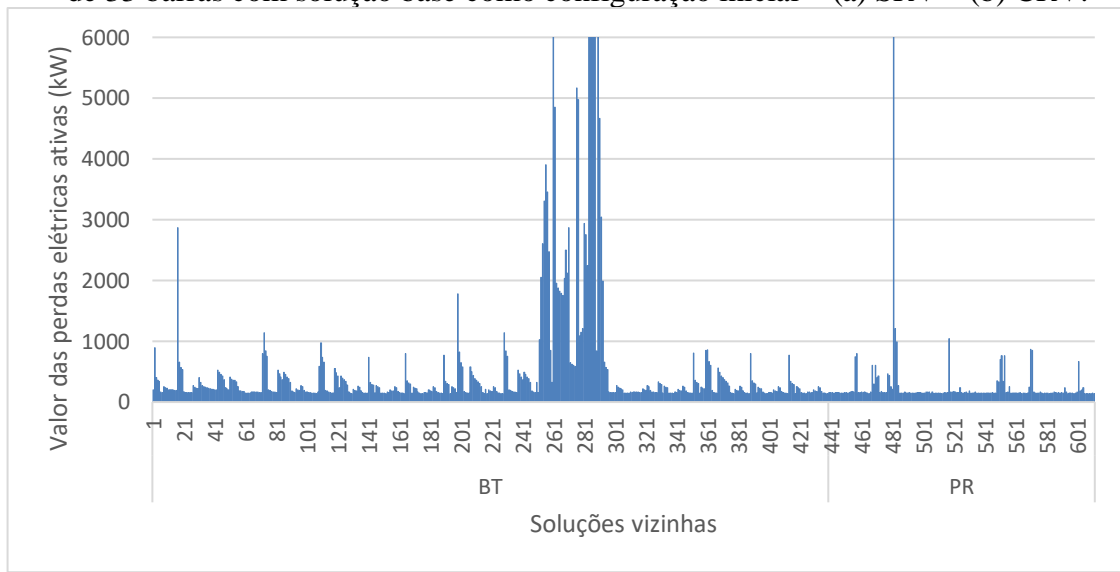
O ocorrido no sistema de 14 barras a BT se repete, possui um rendimento muito superior ao PR, mas diferentemente do sistema de 14 barras, a BT em alguns testes possui quantidades equivalentes de soluções ou bem menos soluções do que as avaliadas pelo PR. Como supracitado, isso vale para as características de cada meta-heurística e a forma de implementação adotada. Ao final deste capítulo essa discussão será tratada com maior profundidade.

As configurações encontradas neste trabalho são as mesmas encontradas em outros trabalhos. Nota-se que o tempo de execução da técnica implementada neste trabalho é bem

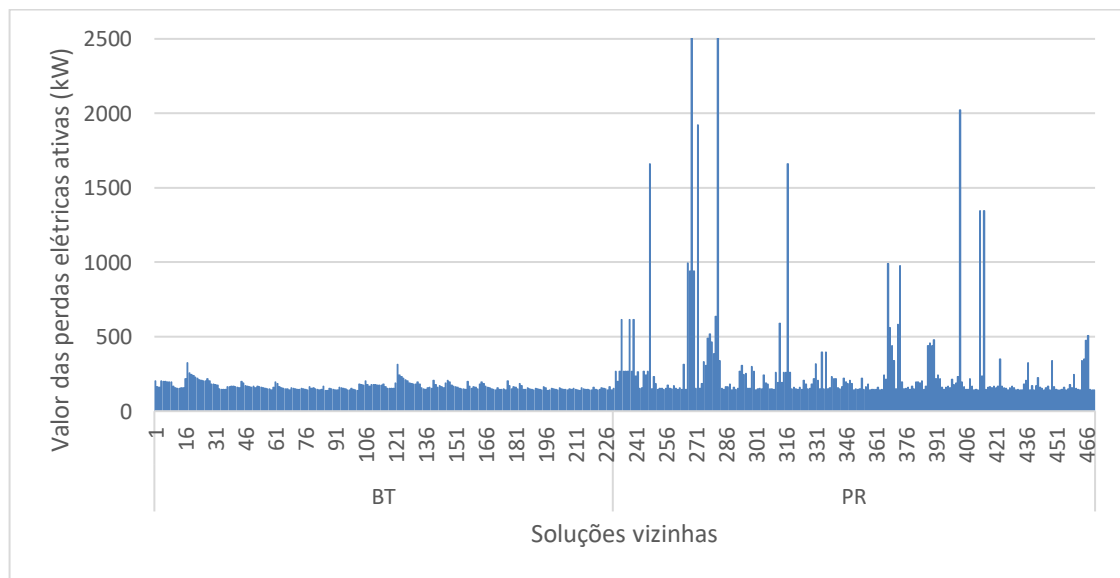
maior do que os tempos comparados de outros trabalhos. Uma melhor explicação da razão de altos tempos de execução do algoritmo proposto será apresentada em um subcapítulo mais adiante informando as vantagens e desvantagens da técnica proposta.

Na Figura 31 apresentam-se gráficos que representam todas as soluções geradas e avaliadas pelo procedimento BT/PR implementado utilizando a solução base como configuração inicial. Na Figura 31(a) apresentam-se os resultados com SRV, enquanto a Figura 31(b) já traz os resultados com CRV.

Figura 31 – Histórico de solução geradas/avaliadas pelo procedimento BT/PR para o sistema de 33 barras com solução base como configuração inicial – (a) SRV – (b) CRV.



(a)

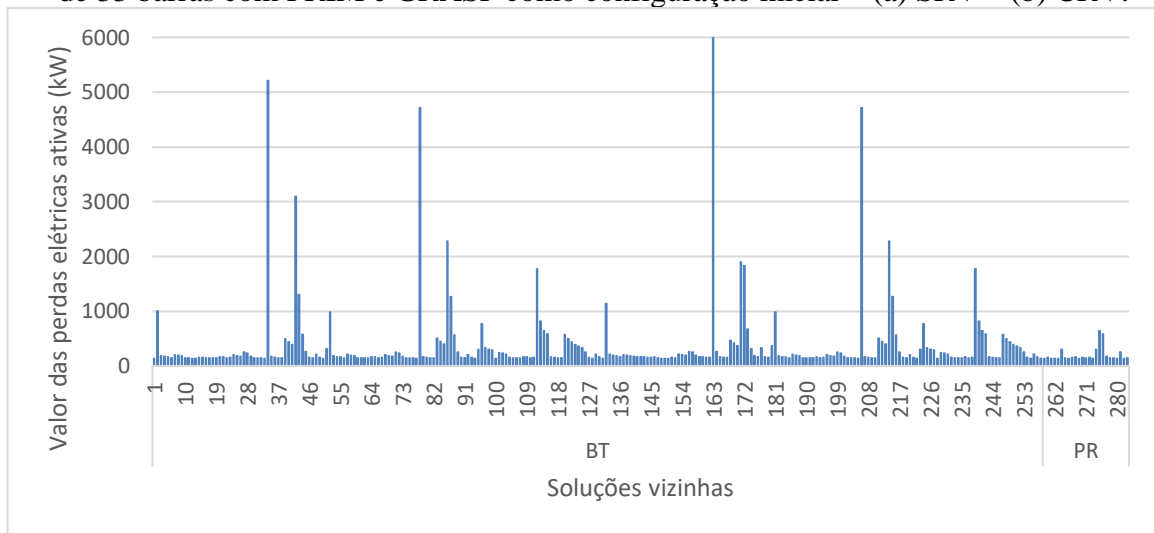


(b)

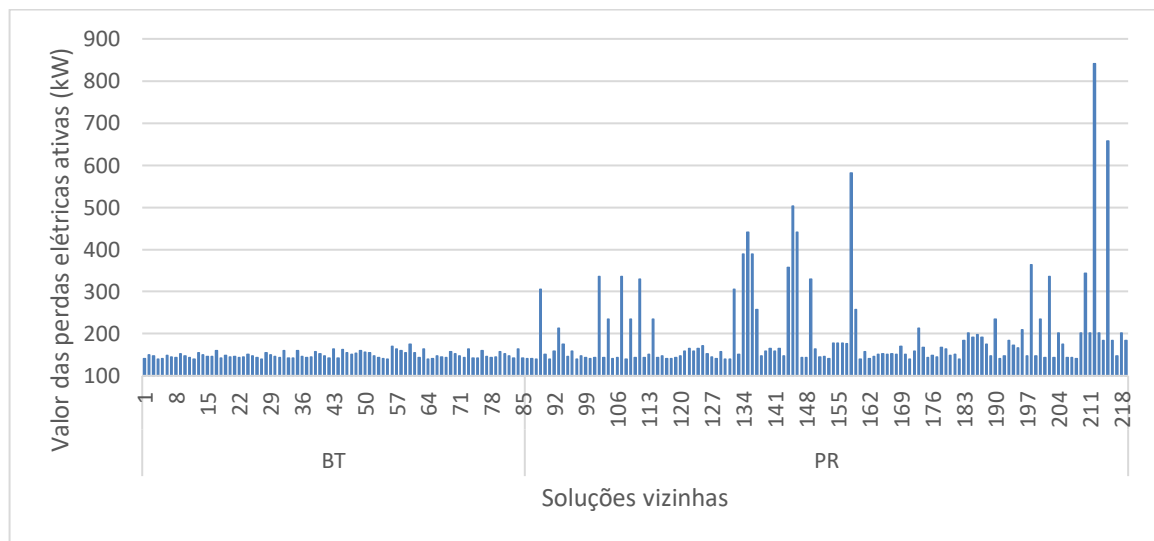
Fonte: Próprio autor

Na Figura 32 representam-se os conjuntos de soluções geradas e avaliadas partindo-se da solução inicial obtida pelos algoritmos PRIM e GRASP. Uma vez que a melhor solução obtida pelos métodos PRIM e GRASP fornecem a mesma solução o conjunto de soluções geradas/avaliadas pela busca local foi o mesmo, produzindo conjuntos semelhantes de soluções elite. Assim, como a Figura 31, a Figura 32 possui suas representações com SRV (Figura 32(a)) e CRV (Figura 32(b)).

Figura 32 - Histórico de solução geradas/avaliadas pelo procedimento BT/PR para o sistema de 33 barras com PRIM e GRASP como configuração inicial – (a) SRV – (b) CRV.



(a)



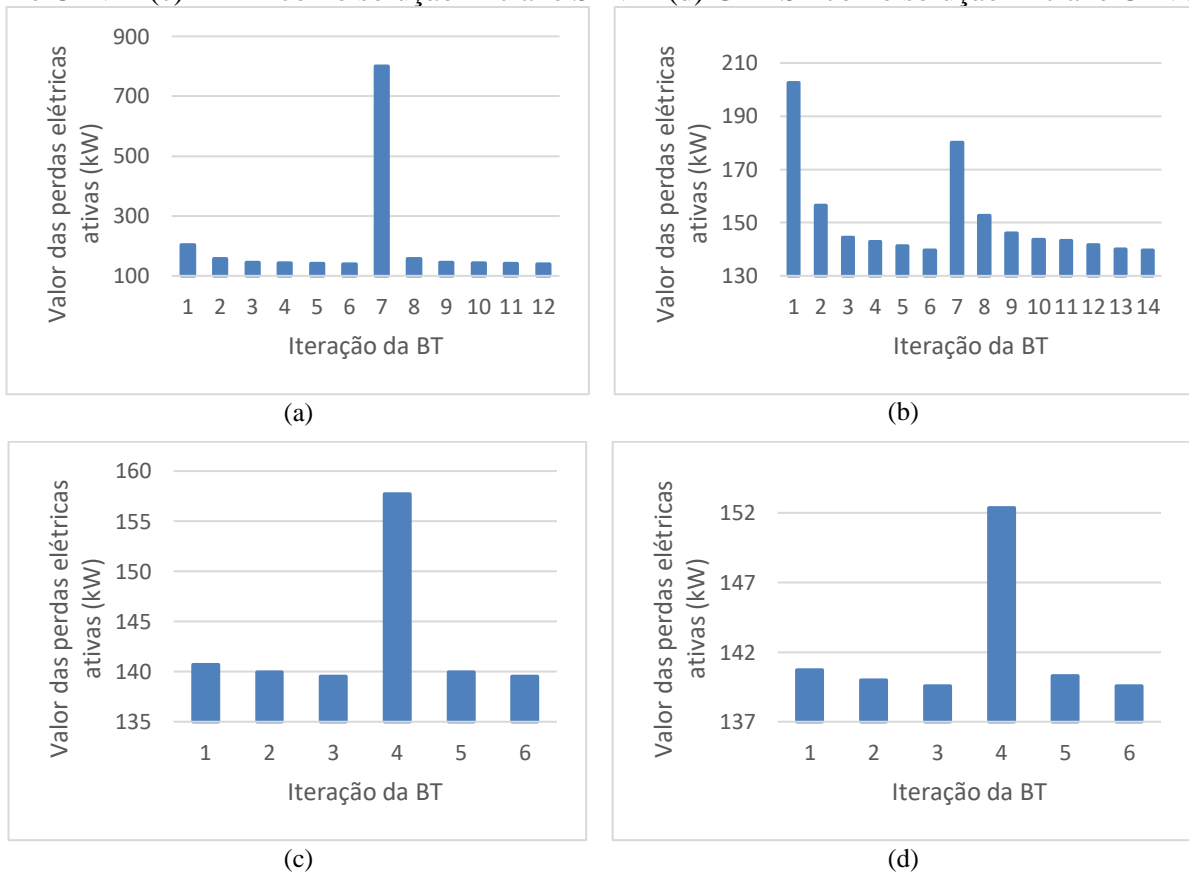
(b)

Fonte: Próprio autor

A solução final obtida pela metodologia proposta obteve o mesmo valor das perdas elétricas e configuração dos trabalhos da literatura. A melhoria obtida entre a solução final obtida e a solução inicial do arquivo foi de 31,1459%, e em relação às soluções obtidas pelas heurísticas PRIM/GRASP foi de 0,8205%. As grandes diferenças entre os tempos obtidos entre as técnicas propostas e as técnicas comparativas são devido os tipos de cada meta-heurística utilizada como comparação. As naturezas de comportamento de cada uma diferem na forma com que se percorre os espaços de soluções. O objetivo foi o de se verificar não somente o tempo de execução, mas também a eficácia e acurácia da técnica proposta.

Na Figura 33 ilustram-se as melhores soluções encontradas a cada iteração da BT. As Figuras 33(a)-(b) representam as melhores soluções obtidas pela BT, com a solução base adotada como configuração inicial, e SRV/CRV, respectivamente. Já as Figuras 33(c)-(d) representam as mesmas características que as Figuras 33(a)-(b), mas utilizando a configuração da solução inicial aquelas obtidas pelas técnicas de PRIM e GRASP.

Figura 33 - Melhores soluções da BT a cada iteração, para o sistema de 33 barras – (a) Solução base como configuração inicial e SRV – (b) Solução base como configuração inicial e CRV – (c) PRIM como solução inicial e SRV – (d) GRASP como solução inicial e CRV.

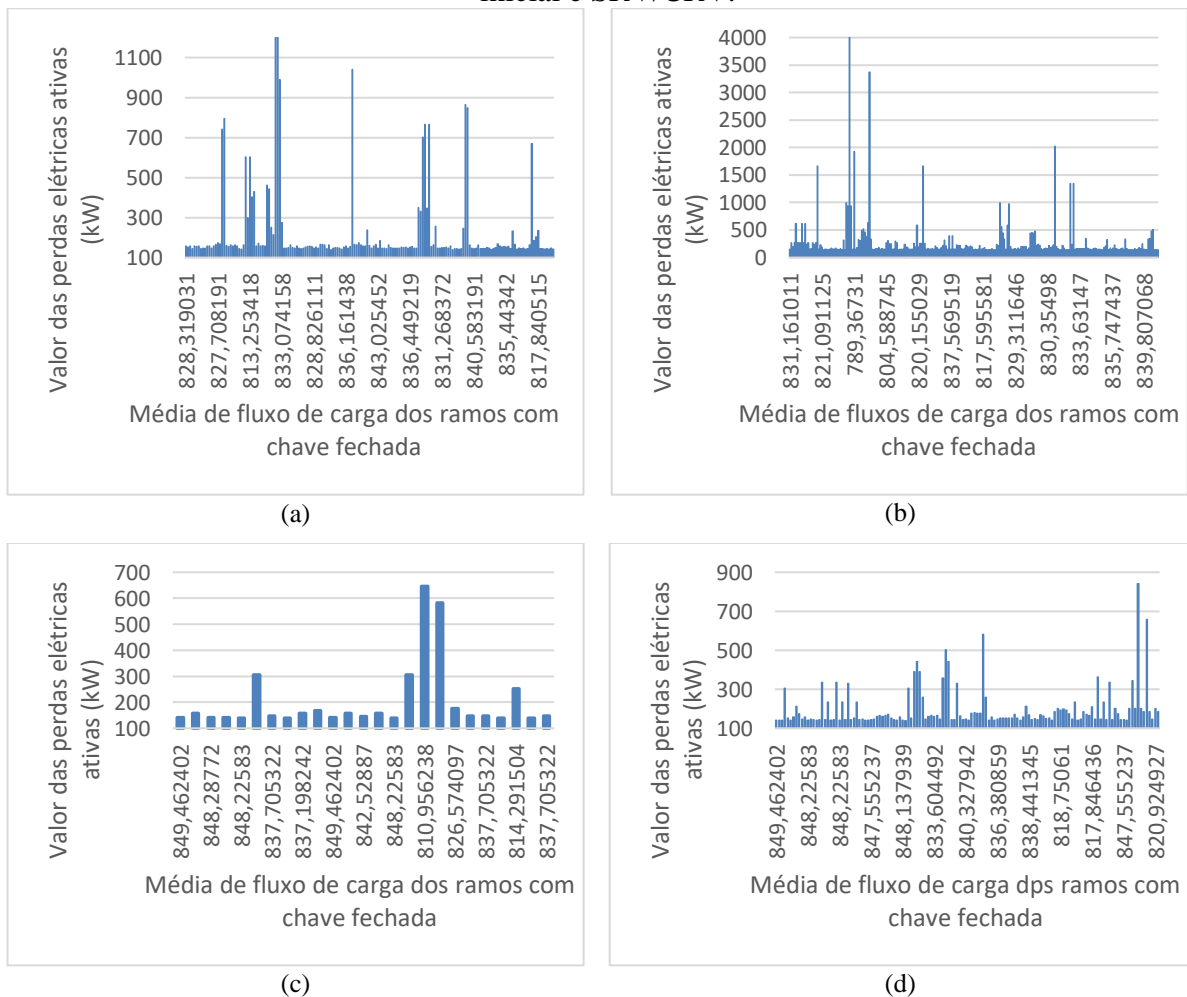


Fonte: Próprio autor

Da Figura 33 verifica-se que as melhores soluções de cada iteração para a BT pertinente a cada tipo de solução inicial escolhida melhoraram quando se aplicou a técnica de redução de vizinhança, isto é, o caminho das soluções durante o procedimento de busca local encontrou as melhores soluções, convergindo de uma forma mais rápida.

Na Figura 34 apresentam-se as soluções obtidas durante o procedimento de pós-otimização por meio do PR para cada tipo de estratégia de obtenção de uma solução inicial, e SRV e CRV. Nas Figuras 34(a)-(b) representam-se os gráficos onde a solução base é adotada como configuração inicial SRV e CRV, respectivamente. Nas Figuras 34(c)-(d) apresentam-se as mesmas características, mas com as configurações iniciais obtidas pelo PRIM e GRASP.

Figura 34 - Histórico de soluções do PR - (a) Solução base como configuração inicial e SRV - (b) Solução base como configuração inicial e CRV - (c) PRIM e GRASP como configuração inicial e SRV/CRV.



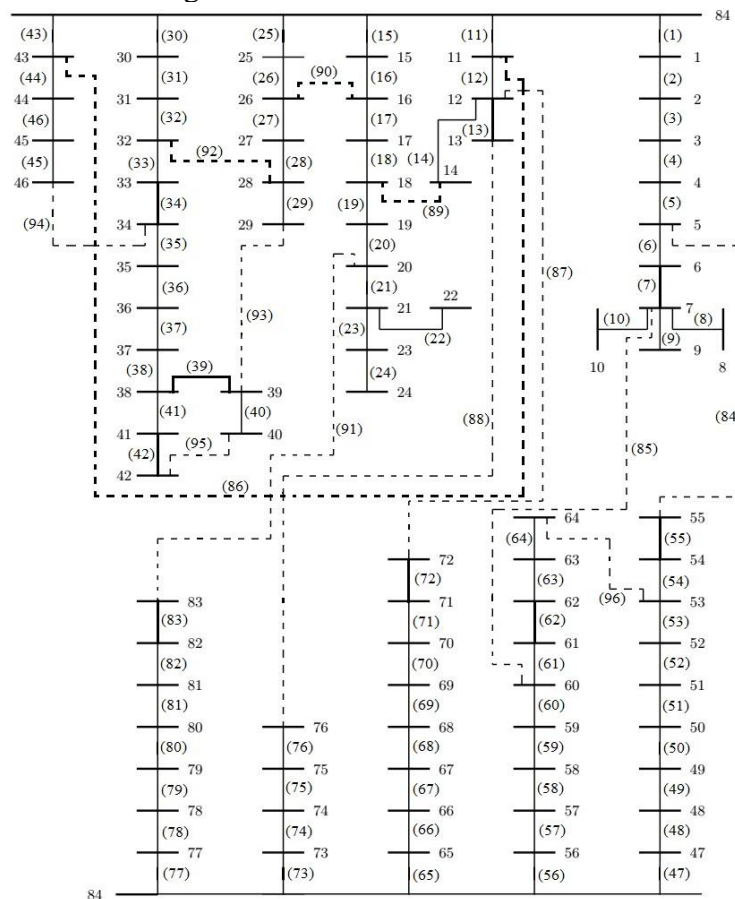
Fonte: Próprio autor

Para os gráficos da Figura 34, a aplicação da técnica de redução de vizinhança acabou por não trabalhar com soluções de melhor qualidade, diferente do que aconteceu com o sistema de 14 barras. Mesmo assim, já era esperado esse tipo de comportamento para um sistema com maior capacidade combinatória, pois a proposta principal do PR é a diversidade das soluções.

5.3.3 Sistemas de 84 barras

O sistema de 84 barras de Chiou *et al.* (2005) contabiliza 84 barras, 96 ramos de ligação, subestação caracterizada pela barra de índice 84, tensão da subestação de 11,4 kV, potência base de 10.000 kVA e um erro limite de convergência de 10^{-5} . Na Figura 28 apresenta-se o diagrama de 84 barras, perfazendo a configuração inicial provinda do arquivo de dados indicando os ramos de ligação com chave aberta cujos índices são de 84 a 96.

Figura 35 - Sistemas de 84 barras.



Fonte: Adaptada de Marinho (2017) e Possagnolo (2015)

Na Tabela 15 apresentam-se as estratégias utilizadas para obtenção de uma solução inicial para o problema RSDEE para este sistema. São apresentadas as chaves abertas das melhores configurações obtidas, as perdas elétricas ativas e os tempos de execução. Verifica-se que os resultados obtidos pelas técnicas propostas alcançaram os mesmos resultados ou resultados muito próximos quando adotada a comparação entre as heurísticas PRIM/GRASP implementadas contra as técnicas dos outros trabalhos mencionados.

Tabela 15 - Estratégias de obtenção de solução inicial para o sistema de 84 barras.

Solução Inicial			
Configuração	Chaves abertas	Perdas Ativas (kW)	Tempo(s)
Base – T0	84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96	531,994477	$< 10^{-3}$
PRIM – T0	7, 33, 39, 42, 63, 72, 82, 84, 86, 88, 89, 90, 92	471,726656	$< 10^{-3}$
GRASP – Fase Construtiva – 1 solução - T0	7, 33, 39, 42, 63, 72, 83, 84, 86, 88, 89, 90, 92	471,198708	0,007
GRASP – Fase Construtiva – 10 soluções - T0			3,074
GRASP – Fase Construtiva – 1 solução – T4	7, 26, 33, 39, 42, 63, 72, 76, 82, 84, 86, 89, 92	512,160400	0,039

Fonte: Próprio autor

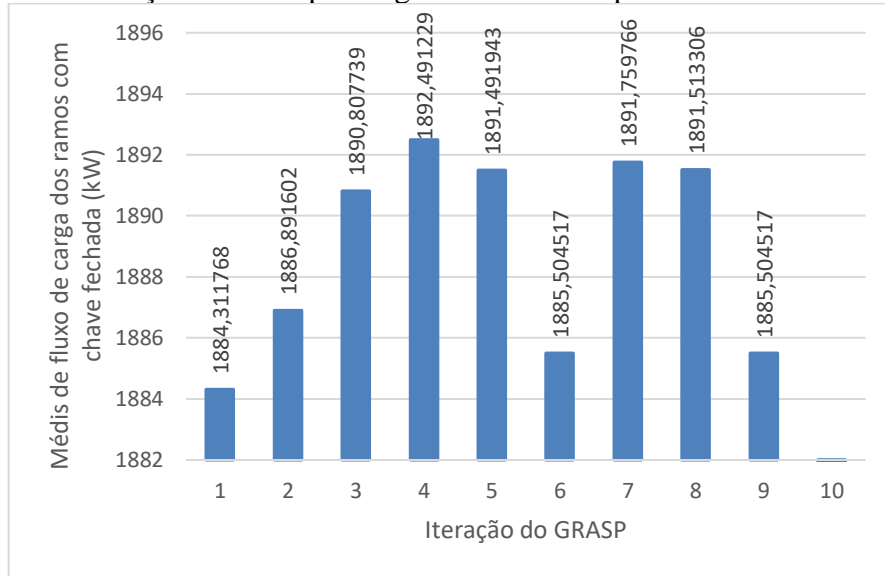
Com um sistema de médio porte como este sistema de 84 barras, o poder combinatório é muito maior fazendo com que a técnica GRASP seja mais eficiente em se encontrar uma melhor solução inicial, como ilustrado na Tabela 15. Diferente do PRIM, o GRASP possui a característica da aleatoriedade na escolha do próximo ramo, dentre uma lista dos possíveis melhores ramos a serem escolhidos. Tal característica faz com que seja possível obter melhores soluções.

Em relação à solução inicial obtida pela solução base, a solução encontrada pela técnica PRIM neste trabalho alcançou uma melhoria de 11,3287% e o GRASP alcançou uma melhoria de 11,4279%. Explicações quanto ao alto tempo de execução da técnica GRASP para dez soluções serão apresentadas ao final deste capítulo.

Na Figura 36 apresentam-se as configurações encontradas durante dez iterações da fase construtiva do algoritmo GRASP para o problema de RSDEE com 84 barras. A maior

dimensão do problema em avaliação proporcionou uma maior variedade de soluções geradas de forma que nenhuma solução gerada foi idêntica em sua configuração, e ainda proporcionou a obtenção de uma solução de melhor qualidade do que a obtida pelo algoritmo PRIM.

Figura 36 - Soluções obtidas pelo algoritmo GRASP para o sistema de 84 barras.



Fonte: Próprio autor

Na Tabela 16 apresentam-se os resultados obtidos pela BT e PR implementadas para cada estratégia de obtenção de solução inicial utilizada. Para a técnica proposta neste trabalho serão apresentados os dados obtidos SRV e CRV. Na Tabela 17 apresentam-se as quantidades de soluções geradas pelas técnicas BT e PR e seus devidos tempos de execução em separado, fazendo com que se visualize melhor o rendimento de cada técnica em separado.

Tabela 16 - Solução final obtida para o sistema de 84 barras.

Solução Final Obtida				
Configuração		Chaves abertas	Perdas Ativas (kW)	Tempo(s)
SRV	Base/BT/PR	7, 13, 34, 39, 42, 55, 62, 72, 83, 86, 89, 90, 92	469,877534	154,366
	PRIM/BT/PR			122,772
	GRASP/BT/PR			102,087
CRV	Base/BT/PR			148,373
	PRIM/BT/PR			117,037
	GRASP/BT/PR			104,010
GRASP – T3			469,879900	1,496
PRIM/BVNS – T4		1,552		
PRIM/VND – T4		1,452		

Fonte: Próprio autor

As pequenas variações entre as soluções finais obtidas pelas técnicas neste trabalho e as técnicas comparativas derivam-se do tipo de linguagem de programação utilizada, IDE adotada e precisão numérica de convergência. Todas as variações da técnica proposta neste trabalho alcançaram a melhor configuração já conhecida, e em relação à solução inicial configurada como configuração inicial (Tabela 15) ocorreu uma melhoria de 11,6768%, e entre a melhor configuração inicial obtida pelo GRASP implementado a melhoria foi de 0,2804%. Isso indica que as soluções obtidas pelo GRASP se encontram mais próximas às melhores soluções de um determinado sistema.

Na Tabela 17 verifica-se que a técnica de redução de vizinhança continuou apresentando resultados satisfatórios quanto à redução do número total de soluções geradas/avaliadas no procedimento de BT, reduzindo em 28% em comparação com as quantidades de soluções geradas/avaliadas sem a técnica de redução de vizinhança. As quantidades de soluções geradas/avaliadas e os tempos de execução no procedimento de PR variam devido às características de sua implementação por escolha de elemento aleatório. Isso será melhor explicado ao final deste capítulo. Mesmo assim, a BT, sozinha, já consegue alcançar a melhor configuração encontrada e com um tempo de execução muito menor do que as técnicas comparativas.

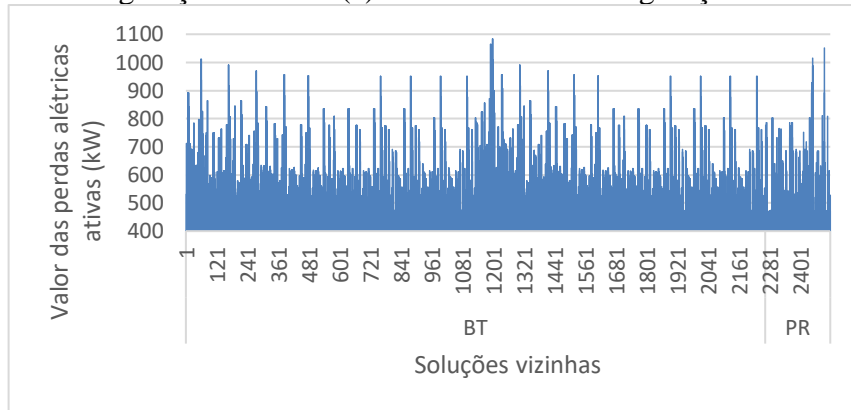
Tabela 17 - Quantidade de soluções geradas/avaliadas, e tempos de execução das técnicas de BT e PR.

Técnicas Adotadas		Soluções geradas	Tempos de Execução (s)
Solução Inicial	Heurísticas		
SRV	Base	BT	0,868
		PR	153,483
	PRIM	BT	0,588
		PR	122,173
	GRASP	BT	0,511
		PR	98,490
CRV	Base	BT	0,303
		PR	148,061
	PRIM	BT	0,221
		PR	116,797
	GRASP	BT	0,182
		PR	100,772

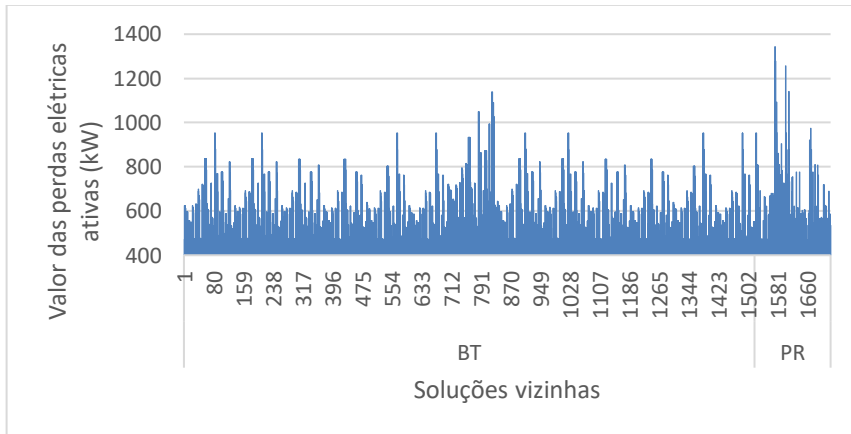
Fonte: Próprio autor

Na Figura 37 apresentam-se os gráficos que representam todas as soluções geradas e avaliadas pelo procedimento BT/PR implementado utilizando cada estratégia de solução inicial obtida e SRV.

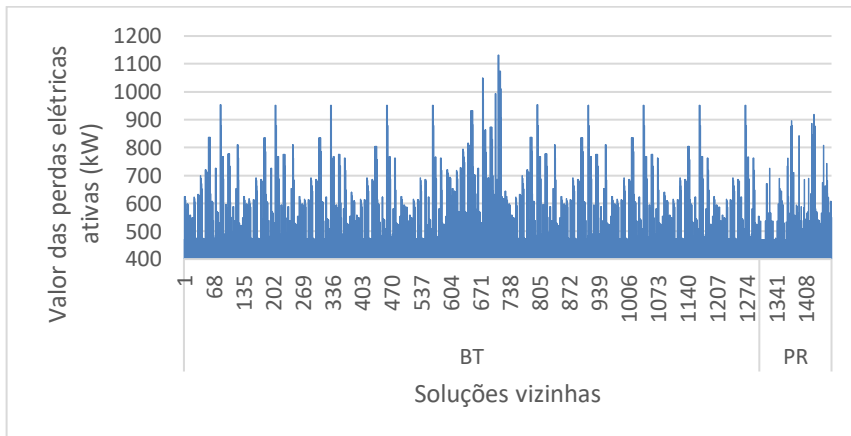
Figura 37 – Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 84 barras e SRV - (a) Solução base como configuração inicial – (b) PRIM como configuração inicial – (c) GRASP como configuração inicial.



(a)



(b)

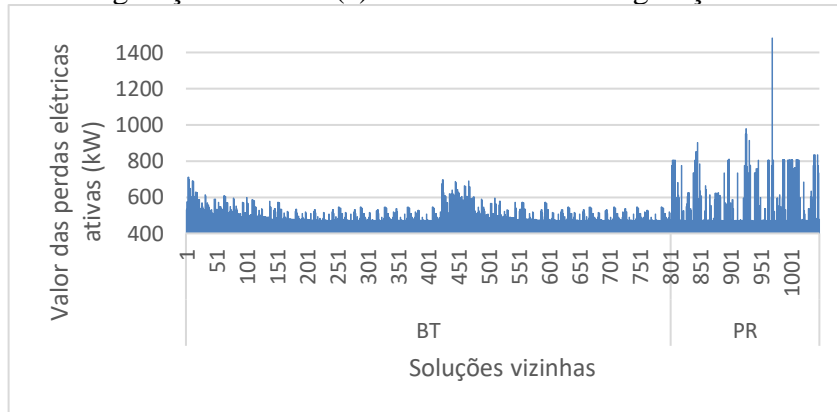


(c)

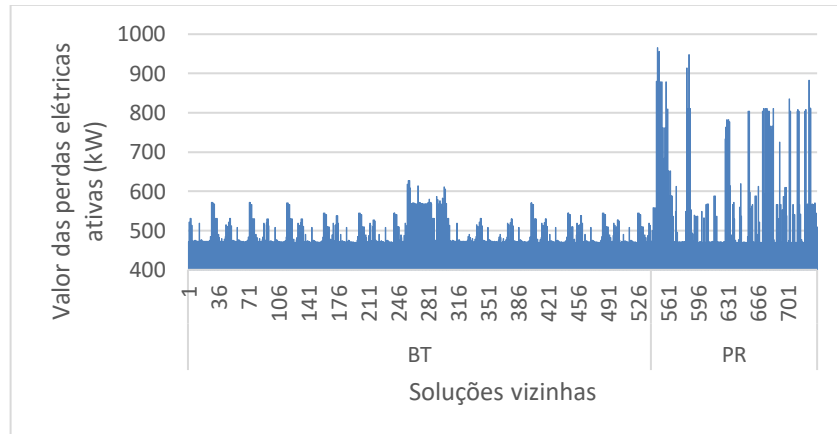
Fonte: Próprio autor

Nas Figuras 37(a)-(b)-(c) representam-se os históricos de soluções geradas/avaliadas tomando como configuração inicial a solução base, PRIM e GRASP. Na Figura 38 apresentam-se as mesmas características da Figura 37, mas adotando-se CRV.

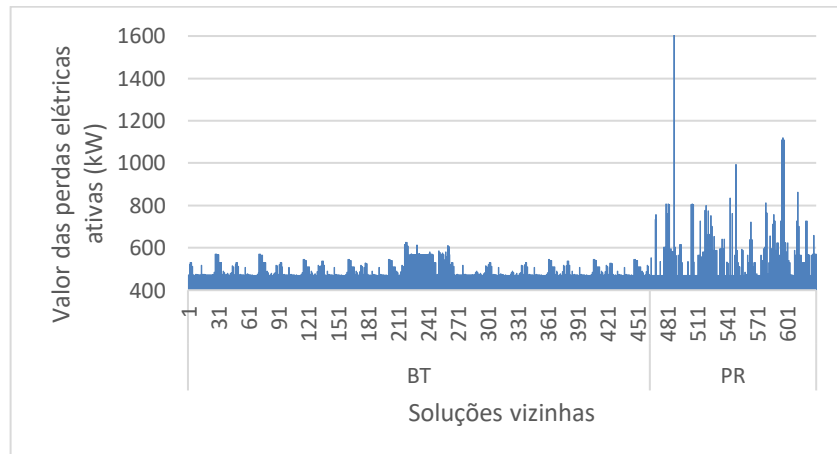
Figura 38 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 84 barras e CRV - (a) Solução base como configuração inicial – (b) PRIM como configuração inicial – (c) GRASP como configuração inicial.



(a)



(b)



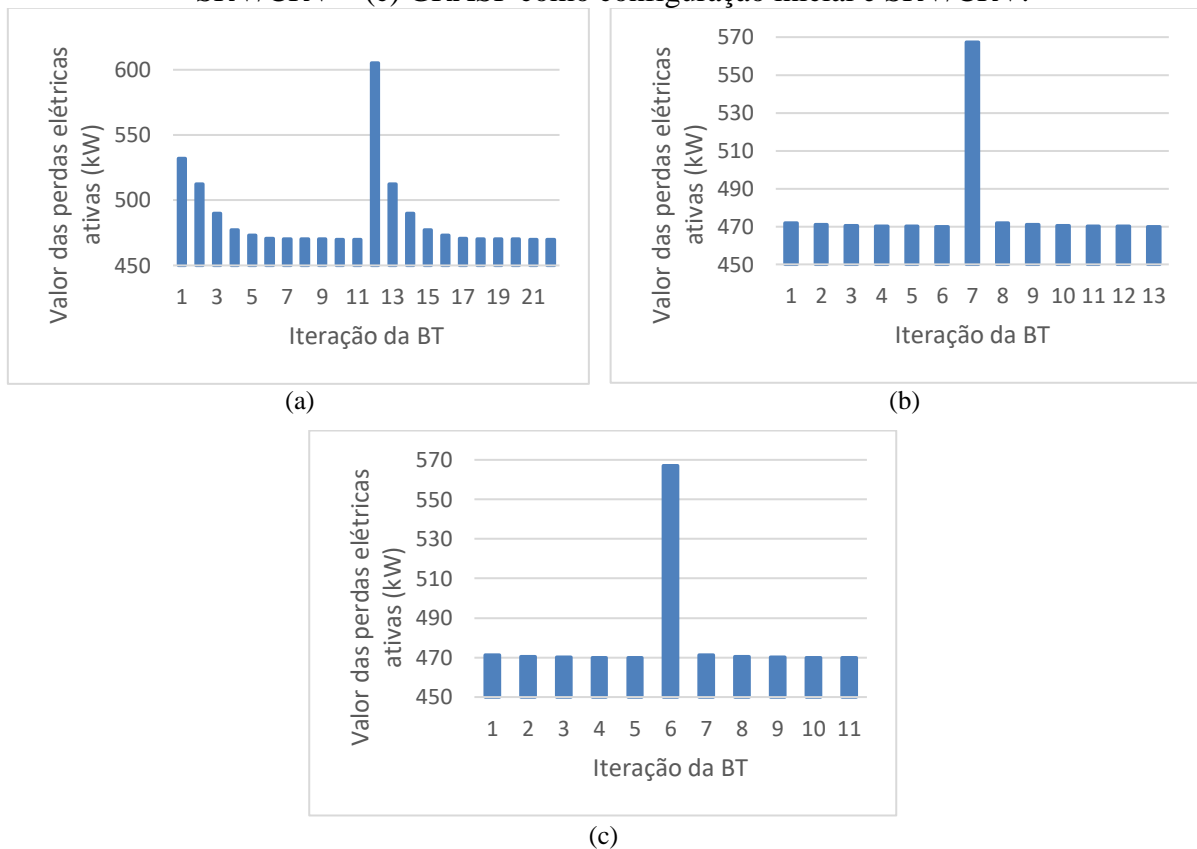
(c)

Fonte: Próprio autor

A quantidade de soluções obtidas quando se parte da solução base provinda do arquivo de dados é bem maior do que a quantidade de soluções obtidas quando se utiliza os métodos PRIM e GRASP. Isso pode ser explicado devido ao método GRASP ter encontrado uma configuração de melhor qualidade para o início do procedimento meta-heurístico BT. A quantidade de soluções geradas e avaliadas pelo procedimento GRASP para gerar uma solução inicial tem praticamente a metade da quantidade avaliada em comparação com as soluções geradas e avaliadas adotando-se como configuração inicial a solução base. O PR tem sua variação dependendo do conjunto de soluções elite e dos procedimentos aleatórios da técnica.

Na Figura 39 apresentam-se as melhores soluções encontradas a cada iteração da BT que são repassadas para as próximas iterações como solução corrente quando adotada como solução inicial a solução provinda do arquivo de dados do sistema.

Figura 39 - Melhores soluções da BT a cada iteração, para o sistema de 84 barras – (a) Solução base como configuração inicial e SRV/CRV – (b) PRIM como configuração inicial e SRV/CRV – (c) GRASP como configuração inicial e SRV/CRV.



(c)
Fonte: Próprio autor

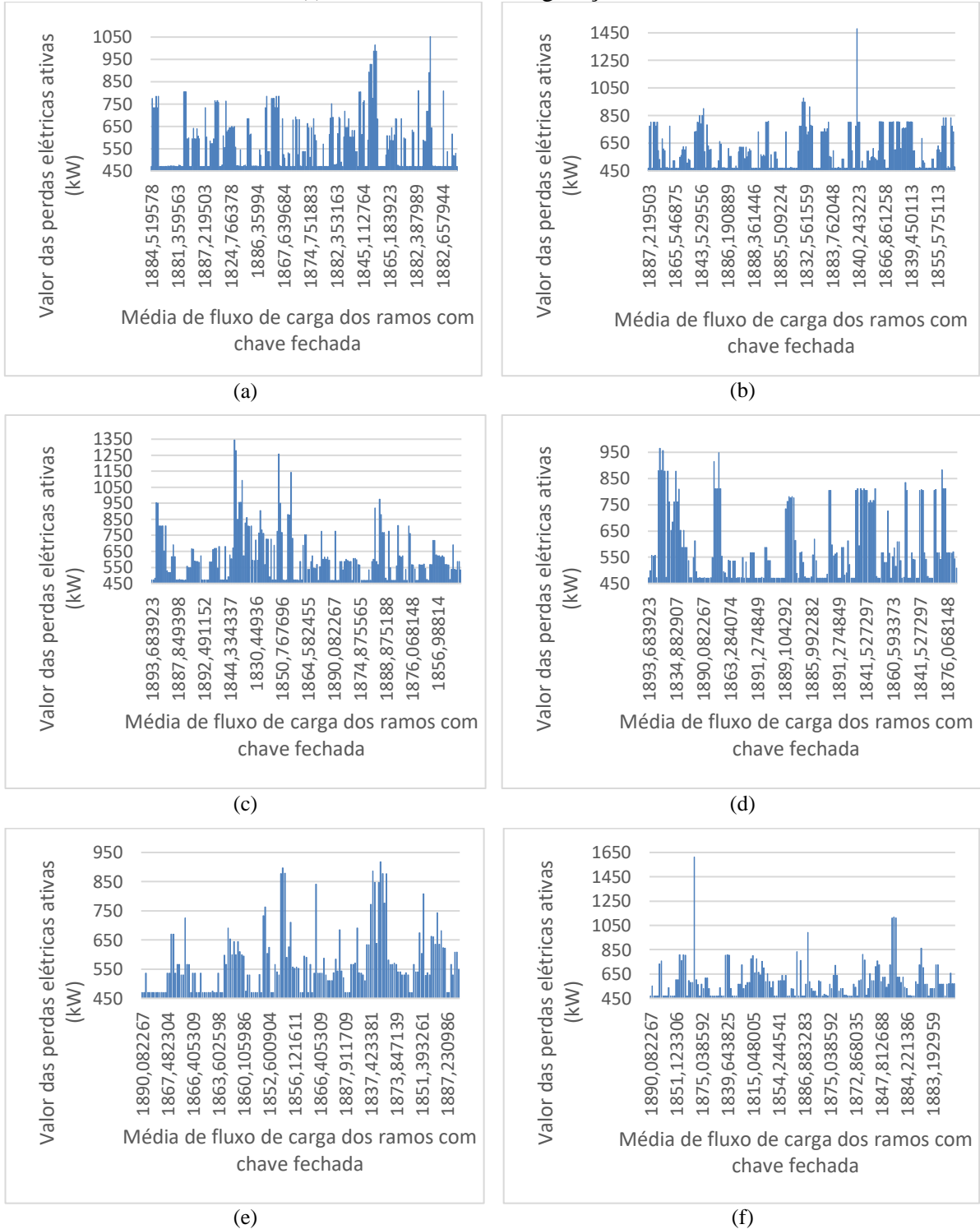
Na Figura 39(a) representam-se as melhores soluções alcançadas partindo da solução base como configuração inicial SRV e CRV; as Figuras 39(b)-(c) possuem a mesma característica, mas com as configurações iniciais obtidas pelo PRIM e o GRASP, respectivamente. Isto mostra que a técnica de redução de vizinhança não alterou as melhores configurações encontradas a cada iteração da BT, mas reduziu o espaço de soluções (Tabela 17).

Na Figura 40 apresentam-se o histórico de soluções obtidas para o procedimento de PR partindo-se de cada estratégia de obtenção de uma configuração inicial SRV e CRV. As Figuras 40(a) e 40(b) possuem como configuração inicial a solução base SRV e CRV, respectivamente. O mesmo ocorre para as Figuras 40(c) e 40 (d) e Figuras 40(e) e 40(f), mas cada uma das figuras parte de configurações iniciais obtidas pelo PRIM e pelo GRASP, respectivamente.

A diversidade das soluções construídas pelo PR é a grande característica desta técnica, buscando por espaços de busca não visitados que podem vir a produzir soluções ainda não visitadas com grande possibilidade de se encontrar melhores soluções. Neste trabalho a melhor solução já foi adotado no procedimento de BT, e nenhuma melhor solução foi obtida pelo procedimento de PR.

A utilização de SRV e CRV não acarreta uma grande diferença para o PR uma vez que somente as dez melhores configurações encontradas na BT que constroem o conjunto de soluções elite, e pelas Figuras 39(b)-(c) as melhores soluções obtidas na BT não sofreram modificação seja pela utilização do PRIM ou GRASP, ou seja pela utilização da SRV e CRV. As soluções intermediárias diferem entre as Figuras 40(a) e 40(b), Figuras 40(c) e 40(d) e Figuras 40(e) e 40(f) devido à característica aleatória de escolha de elementos para a geração de novas soluções intermediárias. Assim o percurso de soluções intermediárias geradas, para um mesmo conjunto de soluções elite, pode ser diferente. A escolha da utilização dessa característica aleatória de escolha de elementos vem exatamente a favor da construção de soluções intermediárias diferentes entre execuções contínuas do algoritmo. A utilização de uma escolha pré-definida de um elemento para compor uma próxima solução intermediária traz a característica determinística da técnicas, o que se tentou evitar, pois esta característica vem a produzir um mesmo conjunto de soluções intermediárias quando se parte de um mesmo conjunto de soluções elite.

Figura 40 – Histórico de soluções do PR – (a) Solução base como configuração inicial e SRV – (b) Solução base como configuração inicial e CRV – (c) PRIM como configuração inicial e SRV – (d) PRIM como configuração inicial e CRV – (e) GRASP como configuração inicial e SRV – (f) GRASP como configuração inicial e CRV.

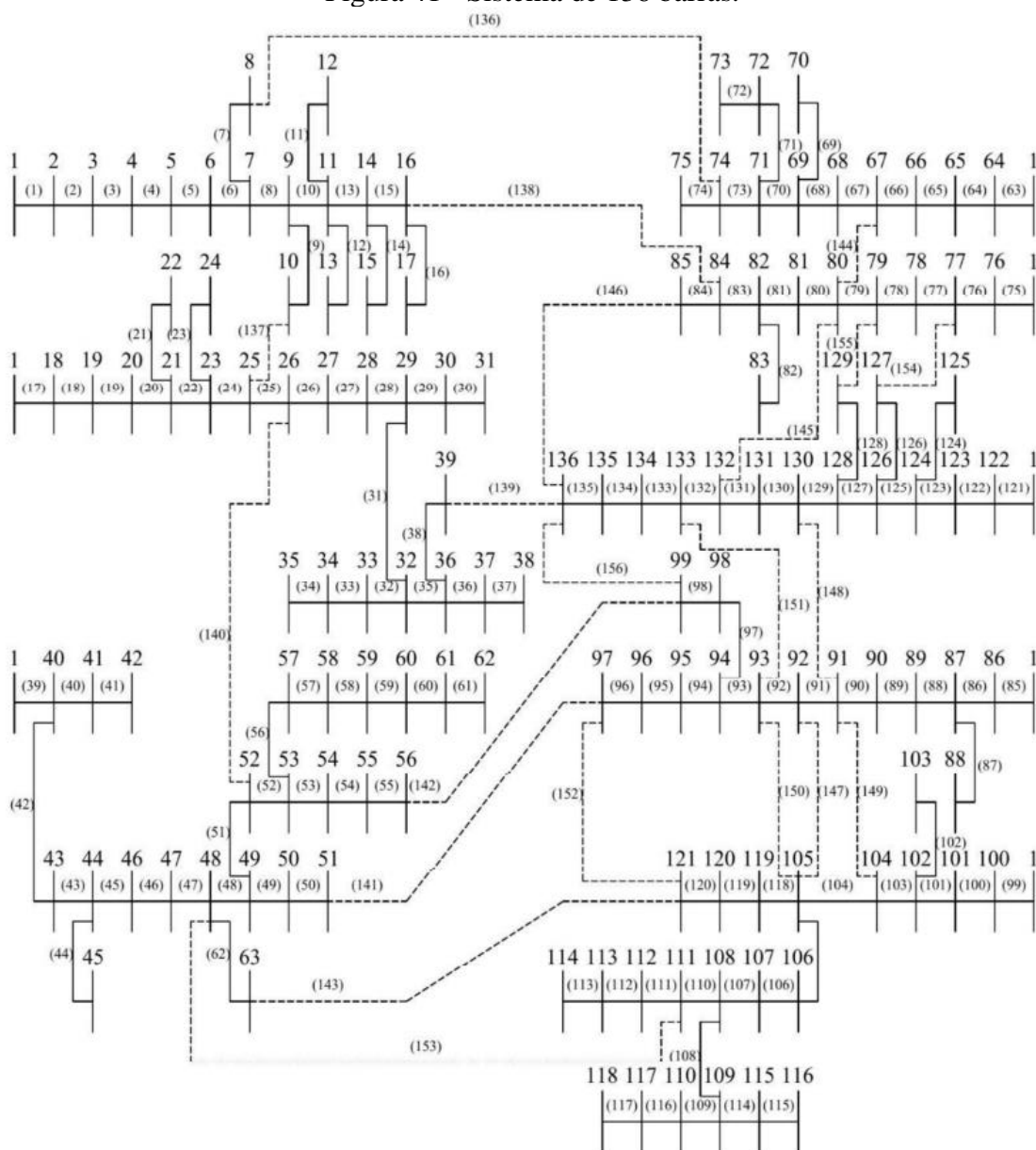


Fonte: Próprio autor

5.3.4 Sistemas de 136 barras

O sistema de 136 barras de Mantovani *et al.* (2000) perfaz um total de 136 barras, 156 ramos, a subestação é caracterizada pela barra de índice 136, a tensão da subestação é de 13,8 kV, a potência base é de 100.000 kVA e o erro limite de convergência de 10^{-8} . Um total de 21 ramos de ligação devem estar com as chaves desligadas (Figura 41, ramos 136-156) para configurar uma configuração radial de uma solução para o sistema de 136 barras. Na Figura 41 apresenta-se o diagrama para o sistema de 136 barras.

Figura 41 - Sistema de 136 barras.



Fonte: Adaptada de Marinho, (2017)

Na Tabela 18 apresentam-se as estratégias utilizadas para obtenção de uma solução inicial para o problema RSDEE com 136 barras, juntamente com a técnica comparativa. São apresentadas as chaves abertas das melhores configurações obtidas, suas devidas perdas elétricas ativas e seus tempos de execução. As técnicas propostas neste trabalho estão referenciadas por T0 e a técnica comparativa por T4 (ISABELLA F. FEITEIRA; LEONARDO H. MACEDO; RUBÉN ROMERO, 2018).

N a Tabela 18 verifica-se que a técnica GRASP proposta neste trabalho supera em qualidade a configuração inicial encontrada. Conforme a dimensão do problema aumenta, mais elementos podem vir a participar da RCL, elementos estes que acabam não sendo tão impactantes. Em contrapartida existe uma maior quantidade de configurações possíveis que podem ser avaliadas. A escolha do valor do indicador de sensibilidade do GRASP afeta diretamente as soluções obtidas.

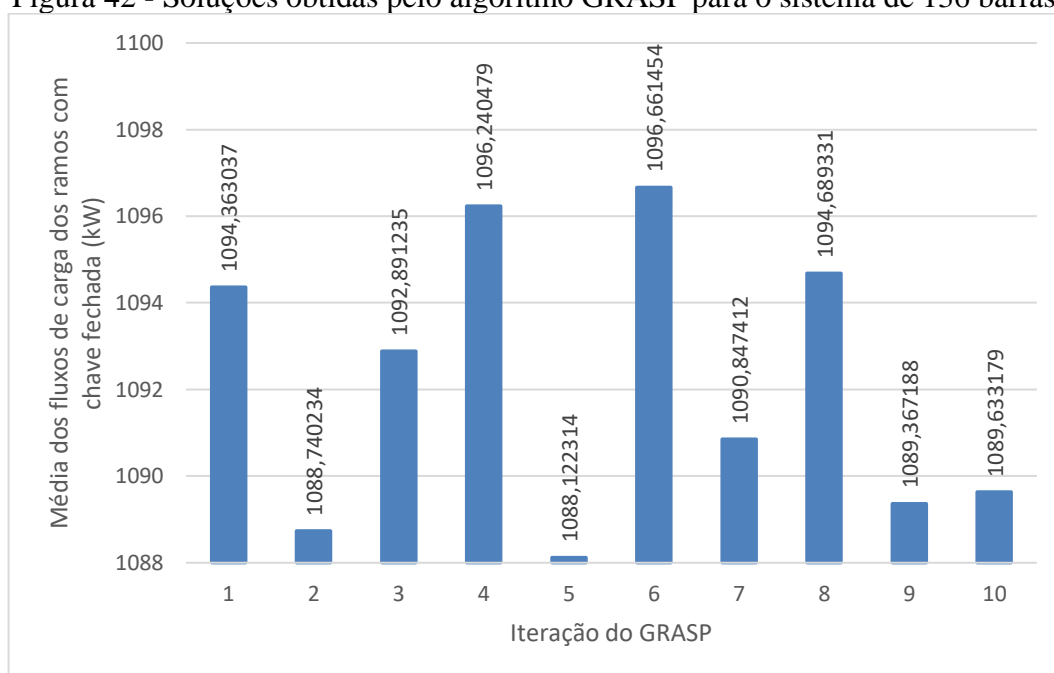
Tabela 18 - Estratégias de obtenção de solução inicial para o sistema de 136 barras.

Solução Inicial			
Configuração	Chaves abertas	Perdas Ativas (kW)	Tempo(s)
Base – T0	136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156	320,364420	$< 10^{-3}$
PRIM – T0	9, 35, 50, 51, 54, 84, 90, 96, 106, 126, 135, 136, 138, 143, 144, 145, 147, 148, 150, 151, 155	292,925630	0,003
GRASP – Fase Construtiva – 1 solução – T0	9, 35, 49, 51, 54, 90, 96, 106, 126, 135, 136, 138, 143, 144, 145, 146, 147, 148, 150, 151, 155	289,400015	0,015
GRASP – Fase Construtiva – 10 soluções – T0			3,1510
GRASP – Fase Construtiva – 1 solução – T4	9, 35, 50, 51, 54, 90, 92, 96, 107, 126, 135, 136, 138, 143, 144, 145, 146, 147, 148, 150, 155	303,580000	0,059

Fonte: Próprio autor

Na Figura 42 apresenta-se o percurso das configurações encontradas durante 10 iterações da fase construtiva do algoritmo GRASP para o problema de RSDEE com 136 barras.

Figura 42 - Soluções obtidas pelo algoritmo GRASP para o sistema de 136 barras.



Fonte: Próprio autor

A dimensão deste sistema proporcionou, ainda, uma maior diversidade das soluções geradas. Um total de dez soluções foram geradas e avaliadas e foram calculadas suas médias do somatório de fluxo de potência dos ramos com chave fechada. A solução com o maior valor desta média é transferida como solução inicial do procedimento de BT. Pela grande dimensão, nenhuma solução gerada foi equivalente.

Na Tabela 19 apresentam-se os resultados obtidos pela BT e PR implementadas para cada estratégia de obtenção de solução inicial utilizada. Para a técnica proposta serão apresentados os dados obtidos SRV e CRV. Na Tabela 20 apresentam-se as quantidades de soluções geradas e avaliadas pelas técnicas BT e PR e seus devidos tempos de execução em separado, podendo se verificar o comportamento de cada técnica em separado.

Analisando-se a Tabela 19, verifica-se que a melhoria obtida pela melhor solução alcançada nesta proposta foi pelo conjunto GRASP/BT/PR com CRV. Em comparação com a solução inicial obtida pela solução base (Tabela 17) ocorreu uma melhoria de 12,5393%, e em comparação com a melhor solução inicial alcançada, também pelo GRASP implementado neste trabalho, a melhoria foi de 3,1815%. Pela Tabela 20 verifica-se que a convergência foi muito mais rápida dada a melhor qualidade da solução inicial obtida. Nos testes realizados deste trabalho, durante o procedimento de busca local pela BT já se alcançava a melhor solução já encontrada e conhecida. Dito isto, os tempos de execução desta proposta com a

CRV para a BT são muito pequenos, dada a quantidade de soluções avaliadas, mostrando a robustez da proposta.

Tabela 19 - Solução final obtida para o sistema de 136 barras.

Solução Final Obtida				
Configuração		Chaves abertas	Perdas Ativas (kW)	Tempo(s)
SRV	Base/BT/PR	7, 38, 51, 53, 90, 96, 106, 118, 126, 137, 138, 141, 144, 145, 146, 147, 148, 150, 151, 155, 156	280,298176	241,354
	PRIM/BT/PR	7, 35, 51, 90, 96, 106, 118, 126, 135, 137, 138, 141, 142, 144, 145, 146, 147, 148, 150, 151, 155	280,193007	176,110
	GRASP/BT/PR			202,053
CRV	Base/BT/PR	7, 38, 51, 53, 90, 96, 106, 118, 126, 137, 138, 141, 144, 145, 146, 147, 148, 150, 151, 155, 156	280,298176	241,215
	PRIM/BT/PR	7, 35, 51, 90, 96, 106, 118, 126, 135, 137, 138, 141, 142, 144, 145, 146, 147, 148, 150, 151, 155	280,193007	184,246
	GRASP/BT/PR			197,191
GRASP – T3		7, 35, 51, 90, 96, 106, 118, 126, 135, 137, 138, 141, 142, 144, 145, 146, 147, 148, 150, 151, 155	280,190300	6,709
PRIM/BVNS – T4				5,290
PRIM/VND – T4				6,070

Fonte: Próprio autor

Tabela 20 - Quantidades de soluções geradas/avaliadas e tempos de execução das técnicas BT e PR.

	Técnicas Adotadas		Soluções geradas	Tempos de Execução (s)
	Solução Inicial	Heurísticas		
SRV	Base	BT	5139	7,378
		PR	388	233,962
	PRIM	BT	3967	5,482
		PR	282	170,611
	GRASP	BT	3467	4,847
		PR	315	194,055
CRV	Base	BT	2661	3,621
		PR	393	237,585
	PRIM	BT	1924	2,798
		PR	300	181,431
	GRASP	BT	1718	2,372
		PR	316	191,694

Fonte: Próprio autor

Na Tabela 20 verifica-se que SRV e CRV a solução inicial utilizada afeta diretamente o total de soluções avaliadas no procedimento de busca local realizado pela BT. Este comportamento já era esperado uma vez que soluções de boa qualidade encontram-se em espaços de busca que, possivelmente, contém soluções de melhor qualidade. Isso não é uma premissa, visto que ótimos locais podem vir a requerer grandes esforços computacionais para que novos espaços de busca sejam avaliados na procura dos ótimos globais.

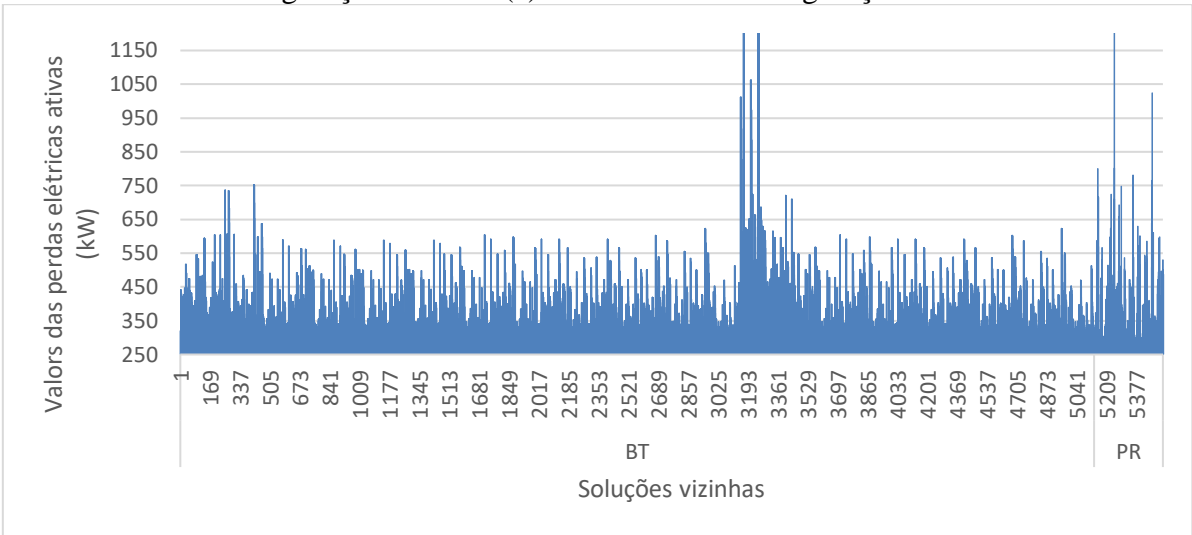
Na Figura 43 apresentam-se os gráficos que representam todas as soluções geradas e avaliadas pelo procedimento BT/PR implementado utilizando cada estratégia de solução inicial obtida SRV. Nas Figuras 43(a)-(b)-(c) representam-se, respectivamente, os históricos de soluções geradas e avaliadas tomando como configuração inicial a solução base, o PRIM e o GRASP. Na Figura 44 apresentam-se as mesmas características da Figura 43, mas CRV.

A quantidade de soluções obtidas durante o procedimento de BT quando utilizada a solução base como configuração inicial é muito maior do que quando utilizado um procedimento mais sofisticado para a obtenção da mesma. Novamente se percebe que quando se escolhe um procedimento heurístico para gerar soluções iniciais de boa qualidade, a quantidade de soluções para o procedimento de busca local ou atuação de um procedimento meta-heurístico, reduz consideravelmente obtendo menores tempos computacionais (Figura 43(b)-(c)). Analisando-se a Figura 43, a redução da quantidade de soluções geradas/avaliadas entre a solução inicial obtida pela solução base (Figura 43(a)) e a solução inicial obtida pelo GRASP (Figura 43(c)) SRV chegou aos 33%.

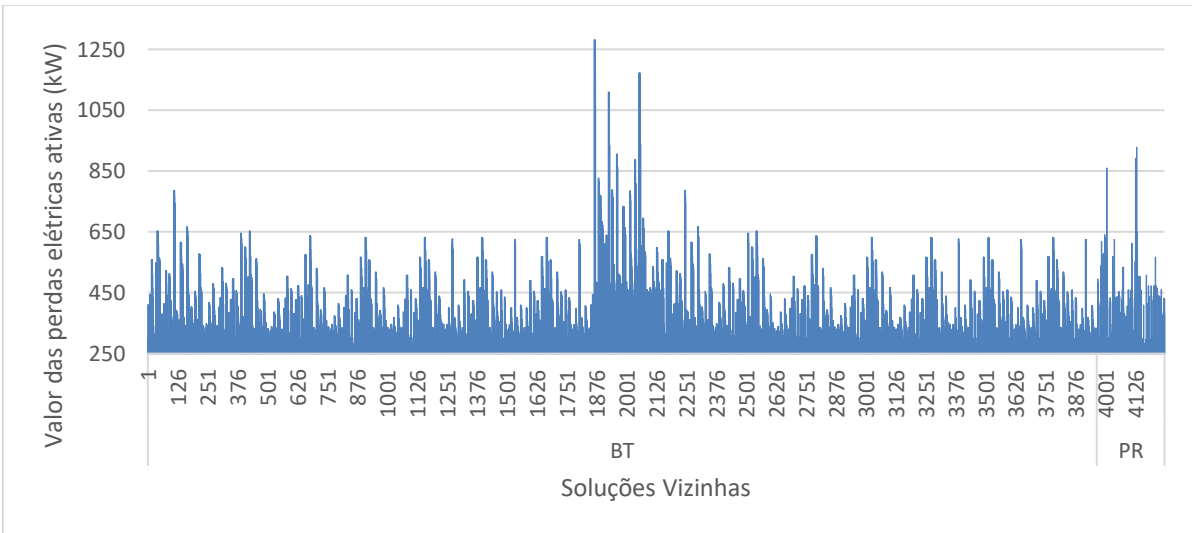
A aplicação da técnica de redução de vizinhança (Figura 44) reduziu em praticamente 50% as soluções geradas no procedimento de BT em comparação com a Figura 43, sem a redução de vizinhança. Essa redução não ocorre no PR devido ao não uso da técnica porque não se avalia um espaço de soluções e sim somente a troca de um elemento (ramo de ligação), escolhido aleatoriamente, para que seja trocada sua chave e gere uma nova solução.

Durante o procedimento de busca local com a BT, de acordo com o algoritmo apresentado na Figura 13, somente uma vez, quando a BT em uma iteração não encontra uma solução melhor do que a solução corrente, ela passa como a solução corrente a pior solução encontrada até o momento e reinicia o processo. Para todas as figuras que apresentam o comportamento da BT, vê-se que existem picos próximos à metade da execução do referido procedimento. É nesses picos que está ocorrendo a troca da solução corrente por uma solução de péssima qualidade e se reiniciando o processo de BT.

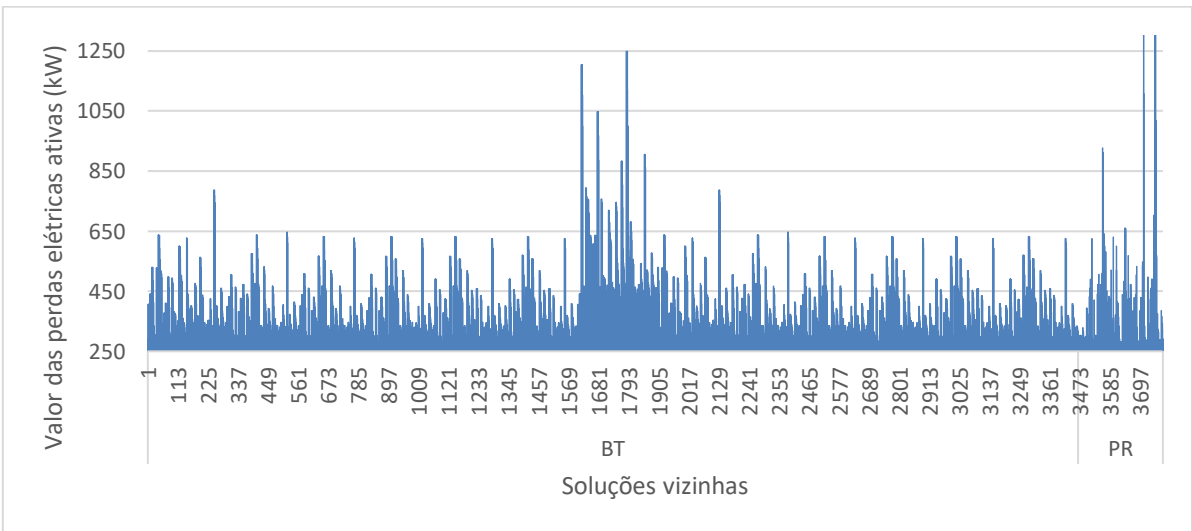
Figura 43 – Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 84 barras SRV – (a) Solução base como configuração inicial – (b) PRIM como configuração inicial – (c) GRASP como configuração inicial.



(a)



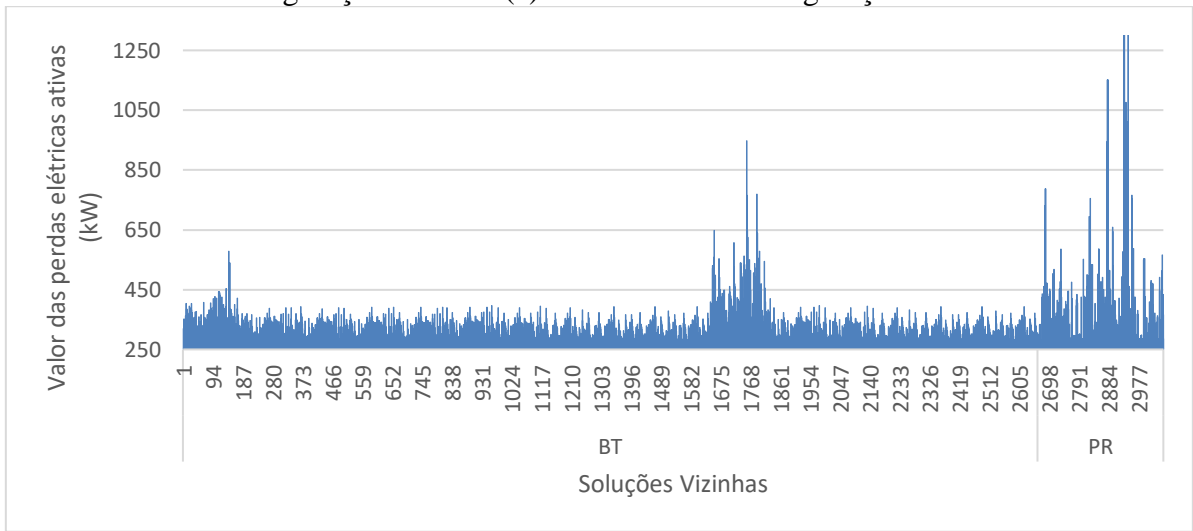
(b)



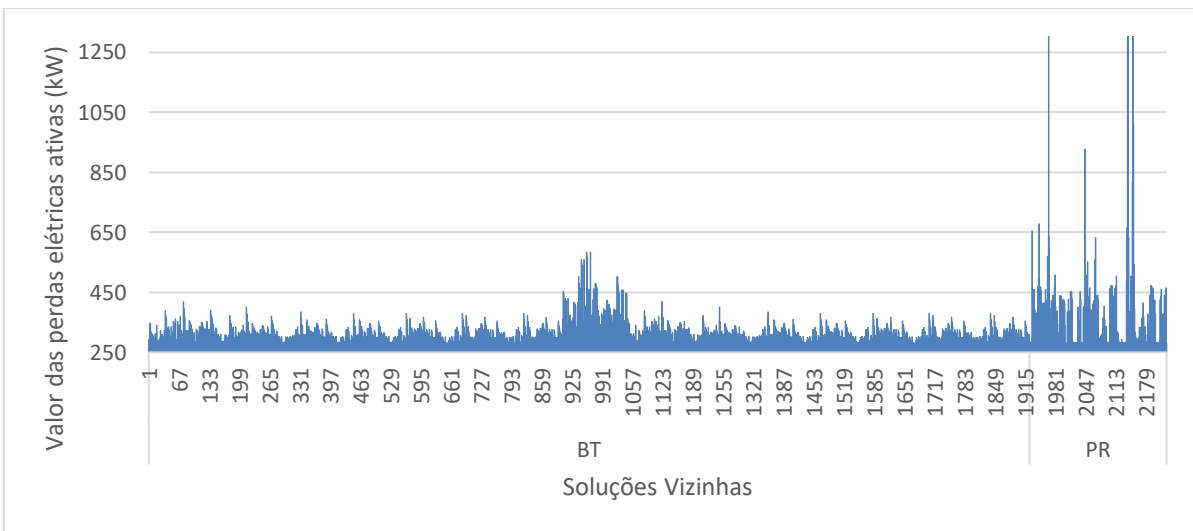
(c)

Fonte: Próprio autor

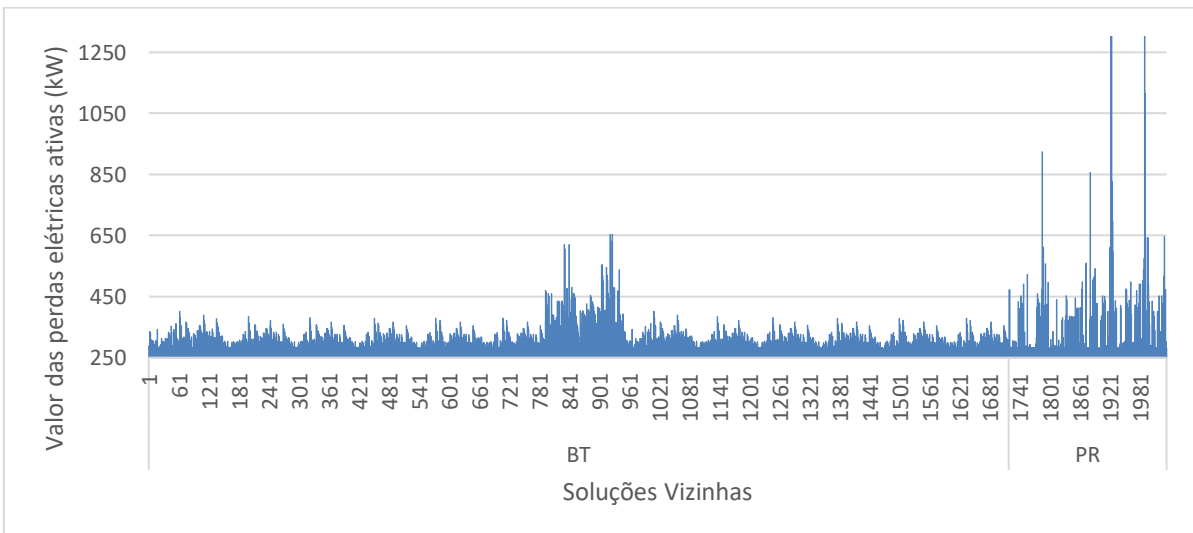
Figura 44 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 136 barras CRV - (a) Solução base como configuração inicial – (b) PRIM como configuração inicial – (c) GRASP como configuração inicial.



(a)



(b)

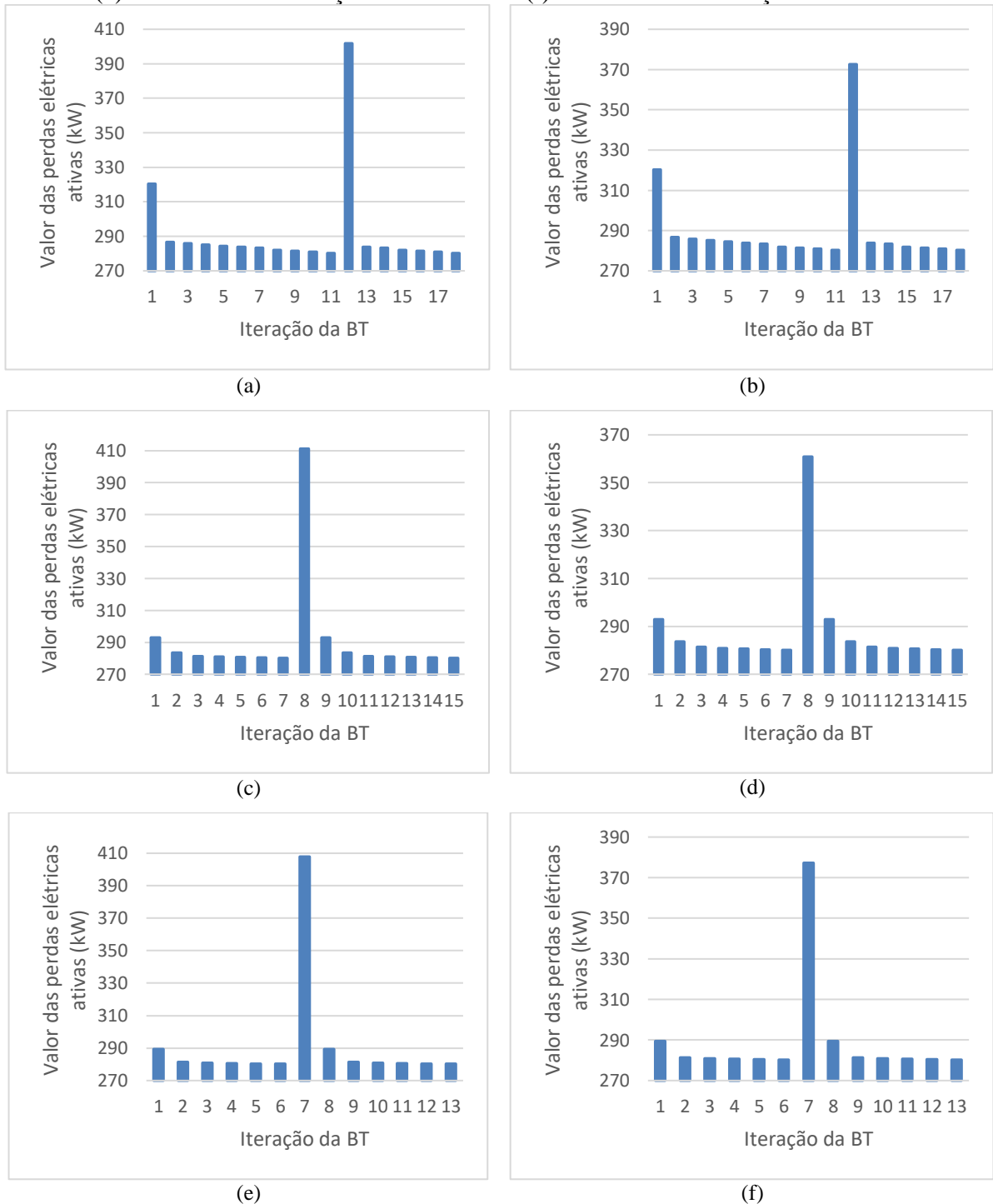


(c)

Fonte: Próprio autor

Na Figura 45 apresentam-se as melhores soluções encontradas a cada iteração da BT que são repassadas para as iterações seguintes como solução corrente quando adotada cada estratégia de solução inicial escolhida SRV/CRV.

Figura 45 – Melhores soluções da BT a cada iteração, para o sistema de 136 barras – (a) Solução base como configuração inicial SRV – (b) Solução base como configuração inicial CRV - (c) PRIM como configuração inicial SRV – (d) PRIM como configuração inicial CRV - (e) GRASP como solução inicial SRV - (f) GRASP como solução inicial CRV.



A Figura 45 apresentou um comportamento sutilmente diferente do que o sistema de 84 barras. Aqui a técnica de redução de vizinhança alterou a solução de má qualidade que é repassada como solução corrente no procedimento de BT quando esta não encontra uma solução melhor do que a solução incumbente em uma iteração. Mas, nota-se que o conjunto de soluções imediatamente antes e imediatamente após, esta solução de má qualidade não apresentou alteração. Isso indica que o espaço de busca é muito amplo para o tipo de vizinhança adotada nesta pesquisa.

A convergência é muito rápida, não no sentido de tempo de processamento, mas de iterações realizadas até obter uma estagnação. Assim como nos outros testes realizados a técnica de redução de vizinhança não alterou as melhores soluções de cada iteração da BT, somente auxiliando na redução do espaço de busca.

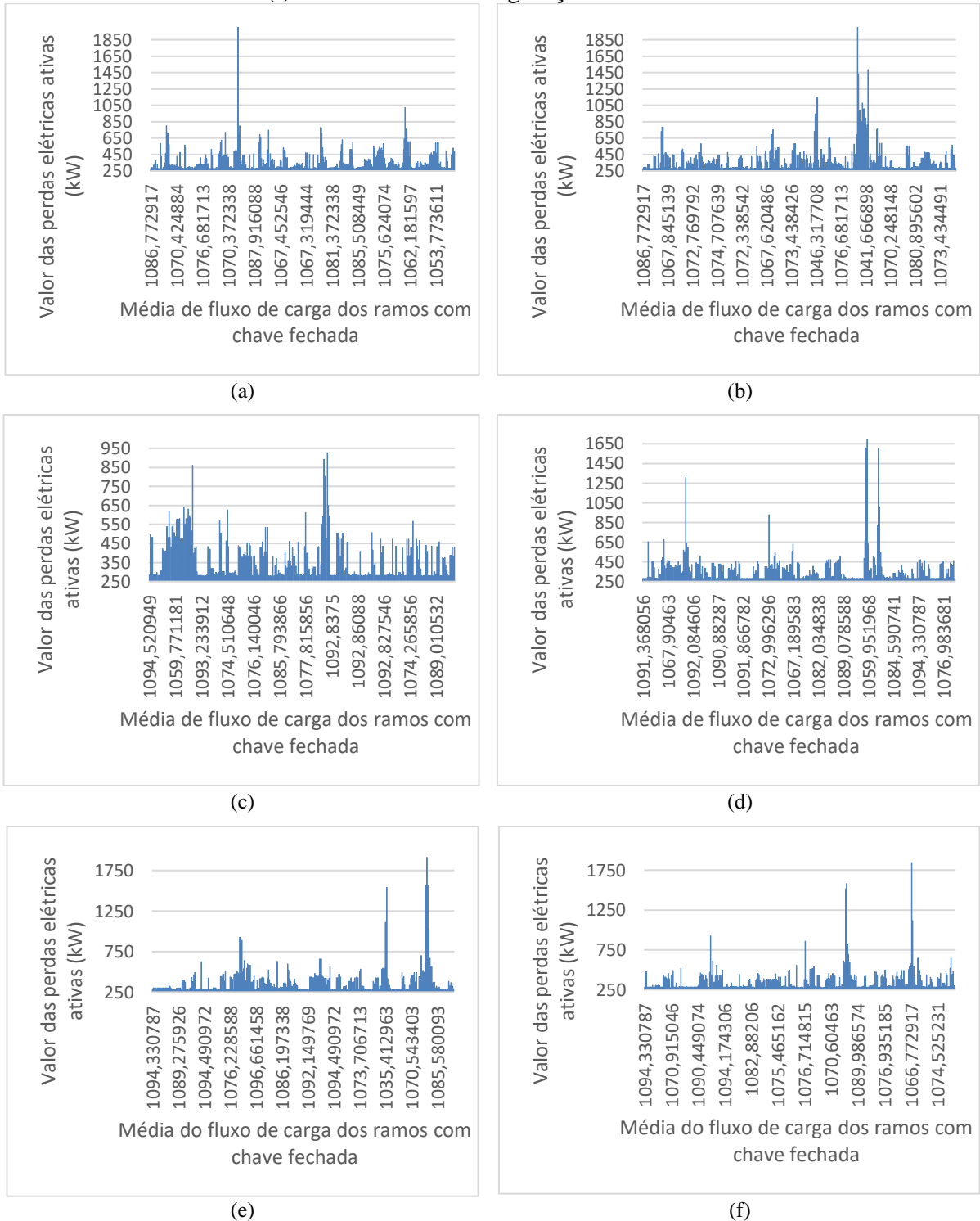
Na Figura 46 apresenta-se o histórico de soluções obtidas pelo procedimento de PR. Na Figura 46 apresentam-se as soluções geradas pelo procedimento de PR adotando cada estratégia de solução inicial utilizada SRV e CRV. As Figuras 46(a) e 46(b) representam as soluções obtidas no procedimento de PR adotando a solução base como configuração inicial e SRV/CRV, respectivamente; já as Figuras 46(c) e 46(d) e as Figuras 46(e) e 46(f) apresentam as mesmas características que as Figuras 46(a) e 46(b) mas para soluções iniciais obtidas pelo PRIM e GRASP, respectivamente.

Como supracitado, a variação das soluções geradas vem da proposta aleatória de escolha do próximo elemento que mudará sua característica, ou seja, os ramos que terão suas chaves abertas e/ou fechadas.

A diferença entre a quantidade de soluções, de cada teste realizado neste trabalho, adotando cada estratégia de solução inicial a ser utilizada SRV e CRV, não muda devido às soluções contidas no conjunto de soluções elite resultante, mas pela característica aleatória de se percorrer da solução inicial à guia. O conjunto de soluções elite é o mesmo SRV e CRV, para cada estratégia de solução inicial adotada, pois a técnica de redução de vizinhança não modificou as melhores soluções visitadas pela BT e, sendo assim, não modificou as soluções que fizeram parte do conjunto de soluções elite.

A técnica de redução de vizinhança fez com que se reduzissem as verificações de cada solução gerada/avaliada para se participar do conjunto de soluções elite, tornando o procedimento de busca local realizado pela BT mais rápido, pois a verificação ocorre na busca local.

Figura 46 - Histórico de soluções do PR – (a) Solução base como configuração inicial SRV – (b) Solução base como configuração inicial CRV – (c) PRIM como configuração inicial SRV – (d) PRIM como configuração inicial CRV – (e) GRASP como configuração inicial SRV – (f) GRASP como configuração inicial CRV.



Fonte: Próprio autor

Durante o procedimento de PR a melhor solução já conhecida, e encontrada pelo procedimento de busca local BT, e soluções muito próximas a esta, foram obtidas diversas vezes para todos os testes executados. Isto se justifica devido às soluções elite estarem entre as dez melhores soluções encontradas no procedimento de busca local, e devido a serem muito próximas umas das outras, fazendo com que a distância entre uma solução e outra passe a gerar poucas soluções. Uma estratégia proposta para prevenir tal comportamento será explicada ao final deste capítulo.

5.3.5 Sistemas de 415 barras

Na Figura 47 ilustra-se o diagrama do sistema de 415 barras em sua configuração radial básica conforme proposto por (I. J. RAMIREZ-ROSADO, 1998). Devido ao tamanho da Figura 47 não foram especificados os índices dos ramos de ligação. O sistema originalmente possui 417 barras, das quais, 3 são barras de subestação, e um total de 473 ramos de ligação. Após reconfigurado o arquivo de dados do sistema, a quantidade de barras passa a ser 415. A barra 1 passa a ser a barra considerada como subestação, com tensão base de 10,00 kV com 1 p.u., as impedâncias fornecidas são por meio de cálculo percentual, a potência base é de 100.000 kVA e o erro limite de convergência é de 10^{-8} .

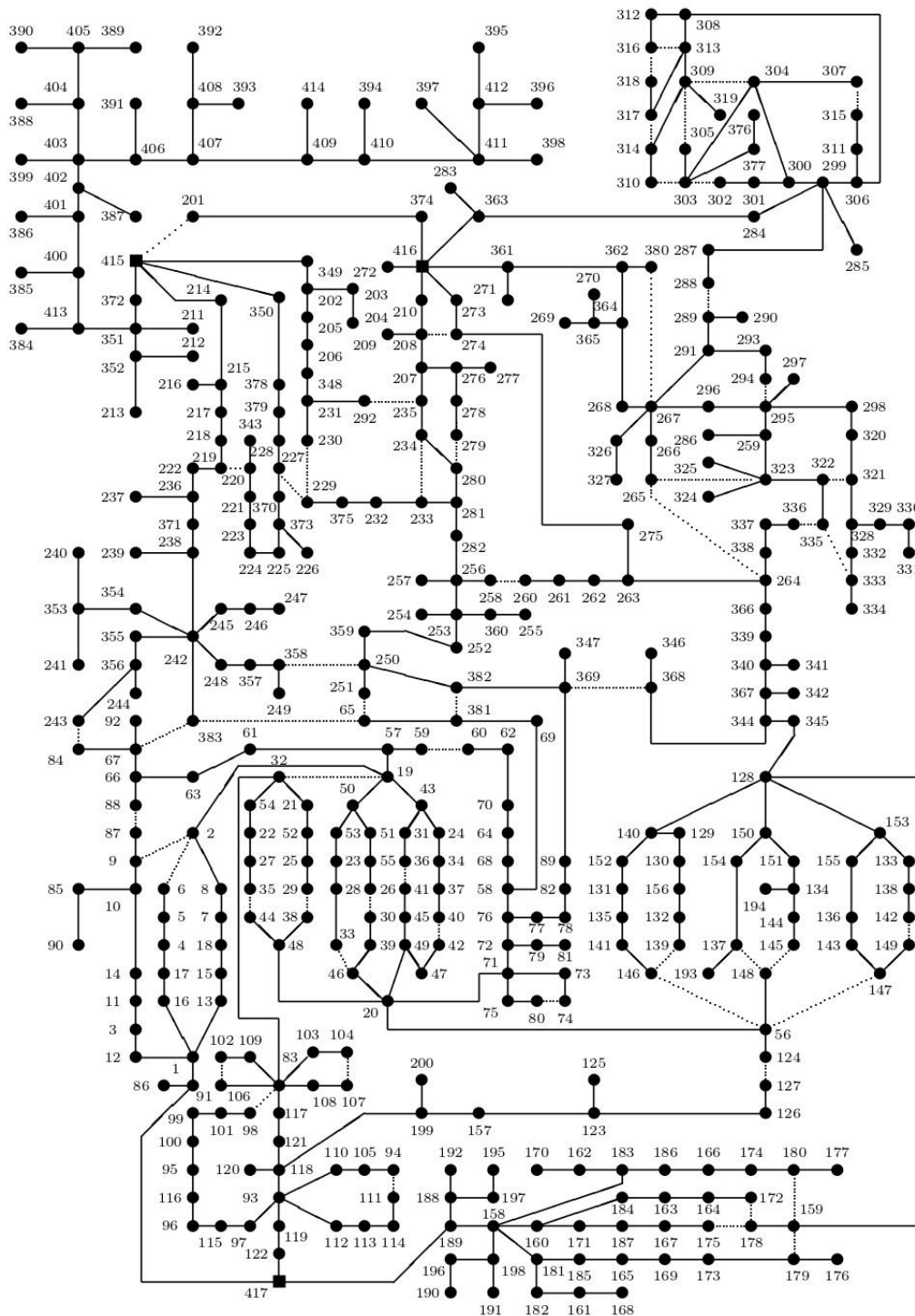
Diferente dos outros sistemas testados, os ramos de ligação que compõem a solução base não se encontram nos últimos índices, como será visto adiante. Como tratado anteriormente, nos testes deste sistema com o GRASP são considerados dois valores para o indicador de sensibilidade, sendo $\alpha = 0,4$, já utilizados nos testes propostos, e $\alpha = 0,1$, permitindo uma maior aleatoriedade na escolha dos próximos elementos a fazerem parte da construção da solução.

Com um sistema de grande porte é mais visível ainda a imensa possibilidade de diferentes configurações possíveis a serem geradas. A explosão combinatória deste sistema é gigantesca, pois são 59 ramos que devem estar com a chave aberta para perfazer uma configuração radial. Isso implica em 59 laços internos independentes a serem identificados, um a um, suas devidas combinações de chaves entre os ramos.

Pode-se inferir que esse é um sistema mal “comportado” por características próprias, onde diversos ramos de ligação possuem fluxo de potência aparente praticamente nulo

(existem muitas barras com demanda nula), o que dificulta o procedimento de chaveamento nas permutações. Isto atrapalha o procedimento do GRASP e da BT porque nem sempre a escolha destes ramos, por terem um fluxo de potência aparente nulo, é a melhor escolha, visto que a partir desta as próximas podem piorar.

Figura 47 - Sistema de 415 barras.



Fonte: Próprio autor

Na Tabela 21 apresentam-se os resultados para a obtenção da configuração inicial pela técnica proposta neste trabalho e pelas técnicas comparativas.

Tabela 21 - Estratégias de obtenção de solução inicial para o sistema de 415 barras.

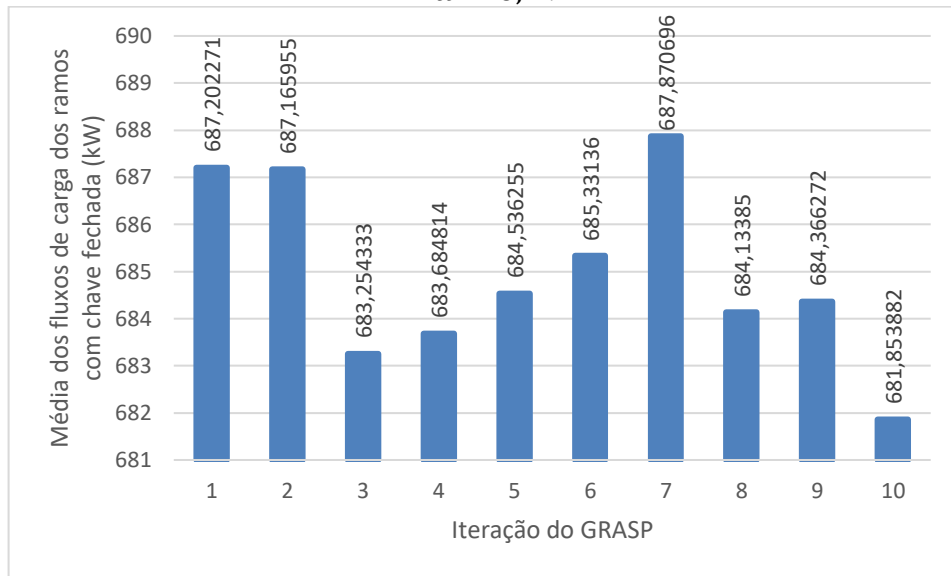
Solução Inicial			
Configuração	Chaves abertas	Perdas Ativas (kW)	Tempo(s)
Base – T0	1, 5, 15, 16, 26, 31, 53, 54, 55, 75, 82, 94, 96, 97, 106, 107, 119, 136, 138, 154, 155, 156, 168, 169, 177, 179, 194, 195, 201, 207, 211, 214, 219, 241, 256, 258, 282, 297, 302, 314, 321, 354, 359, 362, 364, 385, 388, 395, 396, 404, 407, 423, 424, 426, 431, 436, 445, 446, 449	708,941557	0,000
PRIM – T0	1, 2, 5, 15, 16, 21, 26, 29, 31, 40, 50, 75, 82, 94, 96, 97, 110, 111, 119, 136, 142, 154, 155, 156, 163, 168, 169, 179, 194, 201, 209, 211, 214, 229, 256, 282, 297, 302, 314, 321, 354, 362, 372, 385, 392, 395, 396, 403, 404, 423, 424, 426, 431, 436, 437, 446, 449, 466	662,501669	0,079
GRASP – Fase Construtiva – 1 solução - $\alpha = 0,4$ - T0	1, 5, 15, 16, 21, 26, 29, 31, 50, 55, 60, 75, 82, 94, 96, 97, 106, 107, 119, 136, 142, 155, 156, 158, 163, 168, 169, 177, 178, 179, 194, 201, 209, 214, 256, 282, 297, 302, 314, 321, 354, 359, 362, 372, 385, 389, 392, 395, 396, 404, 407, 423, 424, 426, 431, 436, 437, 446, 449, 454	656,057429	0,120
GRASP – Fase Construtiva – 10 soluções - $\alpha = 0,4$ - T0			4,202
GRASP – Fase Construtiva – 1 solução - $\alpha = 0,1$ - T0	4, 5, 11, 16, 26, 31, 44, 45, 50, 55, 65, 75, 90, 94, 96, 99, 106, 121, 128, 142, 155, 156, 163, 166, 169, 177, 179, 181, 201, 209, 211, 214, 219, 241, 256, 282, 293, 302, 314, 321, 362, 372, 385, 392, 395, 396, 403, 404, 423, 424, 426, 431, 433, 436, 437, 446, 449, 466, 467	731,041143	0,110
GRASP – Fase Construtiva – 10 soluções - $\alpha = 0,1$ - T0			4,102
GRASP – Fase Construtiva – 1 solução – T4	1, 5, 15, 16, 26, 29, 31, 47, 50, 60, 75, 82, 94, 96, 97, 103, 110, 117, 119, 132, 142, 155, 156, 158, 168, 169, 173, 179, 195, 201, 209, 214, 225, 245, 256, 282, 293, 302, 313, 314, 333, 362, 364, 385, 389, 392, 395, 396, 400, 402, 403, 423, 424, 431, 436, 437, 446, 449, 466	735,196800	0,126

Fonte: Próprio autor

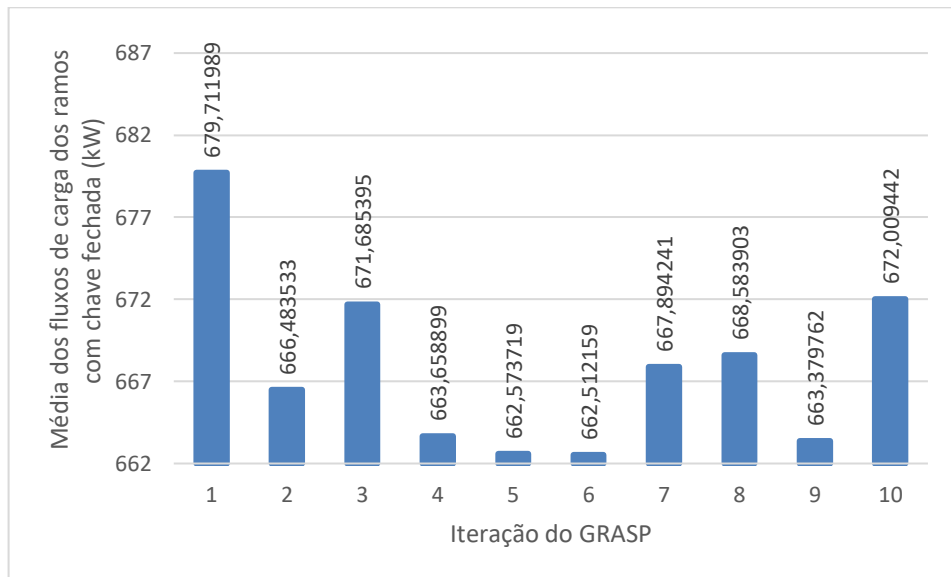
As melhorias obtidas entre as configurações iniciais construídas pelas técnicas de PRIM e GRASP, com o indicador de sensibilidade $\alpha = 0,4$, em relação à solução base foram respectivamente de 6,5506% e 7,4596%. Devido ao tamanho do sistema, e mantendo-se este mesmo indicador de sensibilidade, podemos ter inúmeras propostas de soluções iniciais.

Na Figura 48 apresentam-se as configurações radiais para o sistema elétrico encontradas durante as dez iterações da fase construtiva do GRASP para o problema de RSDEE com 415 barras, para ambos os indicadores de sensibilidade α utilizados.

Figura 48 - Soluções obtidas pelo GRASP para o sistema de 415 barras – (a) $\alpha = 0,4$ – (b) $\alpha = 0,1$.



(a)



(b)

Fonte: Próprio autor

Um total de dez soluções foram geradas e avaliadas suas médias do somatório de fluxo de potência dos ramos com chave fechada. A solução com o maior valor desta média é a solução inicial do procedimento de BT. Na Figura 48(a) utiliza-se o indicador de sensibilidade $\alpha = 0,4$, e na Figura 48(b) o indicador de sensibilidade $\alpha = 0,1$. A principal razão da utilização de dois indicadores diferentes foi a solução final obtida partindo da solução inicial gerada pelo GRASP.

Na Tabela 22 apresentam-se os resultados obtidos pela BT e PR implementadas para cada estratégia de obtenção de solução inicial utilizada, juntamente com as técnicas comparativas. Para a proposta deste trabalho serão apresentados os resultados SRV e CRV.

Tabela 22 - Solução final obtida para o sistema de 415 barras.

Solução Final Obtida				
Configuração		Chaves abertas	Perdas Ativas (kW)	Tempo(s)
SRV	Base/BT/PR	1, 4, 13, 15, 16, 26, 31, 40, 49, 54, 59, 73, 82, 94, 96, 97, 107, 119, 136, 138, 150, 155, 156, 158, 163, 168, 169, 173, 179, 163, 168, 169, 173, 179, 230, 254, 256, 267, 270, 282, 314, 317, 354, 362, 385, 389, 392, 395, 403, 404, 423, 424, 426, 433, 436, 443, 446, 449, 466	586,652663	1460,250
	PRIM/BT/PR	4, 5, 13, 15, 16, 21, 26, 31, 33, 40, 50, 59, 73, 82, 94, 96, 97, 111, 115, 136, 142, 150, 155, 156, 158, 163, 168, 169, 177, 179, 191, 194, 195, 209, 214, 230, 256, 258, 267, 270, 282, 314, 317, 354, 362, 385, 392, 395, 403, 404, 423, 424, 426, 436, 437, 443, 446, 449, 466	586,121297	1081,470
	GRASP/BT/PR ($\alpha = 0,4$)	3, 5, 13, 15, 16, 21, 26, 31, 33, 40, 50, 59, 73, 82, 94, 96, 97, 107, 119, 136, 142, 150, 155, 156, 158, 163, 168, 169, 177, 179, 191, 194, 195, 209, 214, 230, 256, 258, 267, 270, 282, 314, 317, 354, 362, 385, 392, 395, 403, 404, 423, 424, 426, 433, 436, 443, 446, 449, 465	586,927217	1240,970
	GRASP/BT/PR ($\alpha = 0,1$)	1, 4, 13, 15, 16, 30, 31, 40, 49, 50, 59, 73, 82, 94, 96, 97, 107, 119, 136, 142, 150, 155, 156, 158, 163, 168, 169, 173, 179, 188, 191, 194, 195, 209, 214, 230, 256, 258, 267, 282, 302, 314, 317, 362, 385, 389, 392, 395, 403, 404, 423, 424, 426, 433, 436, 443, 446, 449, 466	587,272365	1563,875
CRV	Base/BT/PR	1, 2, 4, 15, 16, 26, 31, 45, 51, 54, 60, 73, 82, 94, 96, 97, 107, 119, 136, 138, 150, 155, 156, 158, 163, 168, 169, 179, 191, 194, 195, 207, 209, 211, 214, 230, 254, 256, 267, 270, 282, 314, 317, 354, 362, 369, 392, 395, 403, 404, 369, 392, 395, 403, 404, 443, 446, 449, 466	587,803265	1249,900
	Continua na página seguinte.			

Continuação da Tabela 22.			
PRIM/BT/PR	4, 5, 13, 15, 16, 21, 26, 31, 33, 40, 50, 59, 73, 82, 94, 96, 97, 111, 115, 136, 142, 150, 155, 156, 158, 163, 168, 169, 177, 179, 191, 194, 195, 209, 214, 230, 256, 258, 267, 270, 282, 314, 317, 354, 362, 385, 392, 395, 403, 404, 423, 424, 426, 436, 437, 443, 446, 449, 466	586,121297	957,379
GRASP/BT/PR ($\alpha = 0,4$)	3, 5, 13, 15, 16, 21, 26, 31, 33, 40, 50, 59, 73, 82, 94, 96, 97, 107, 119, 136, 142, 150, 155, 156, 158, 163, 168, 169, 177, 179, 191, 194, 195, 209, 214, 230, 256, 258, 267, 270, 282, 314, 317, 354, 362, 385, 392, 395, 403, 404, 423, 424, 426, 433, 436, 443, 446, 449, 465	596,927217	1139,290
GRASP/BT/PR ($\alpha = 0,1$)	1, 4, 13, 15, 16, 30, 31, 40, 49, 50, 59, 73, 82, 94, 96, 97, 107, 119, 136, 142, 150, 155, 156, 158, 163, 168, 169, 173, 179, 188, 191, 194, 195, 209, 214, 230, 256, 258, 267, 282, 302, 314, 317, 362, 385, 389, 392, 395, 403, 404, 423, 424, 426, 433, 436, 443, 446, 449, 466	583,560718	1459,674
GRASP – T3	4, 5, 13, 15, 16, 21, 26, 31, 33, 40, 50, 59, 73, 82, 94, 96, 97, 107, 119, 136, 142, 150, 155, 156, 158, 163, 168, 169, 177, 179, 191, 195, 209, 214, 225, 230, 254, 256, 267, 270, 282, 314, 317, 325, 362, 385, 392, 395, 403, 404, 423, 424, 426, 433, 436, 443, 446, 449, 466	586,902000	289,357
PRIM/BVNS – T4	1, 2, 13, 15, 16, 30, 31, 37, 40, 50, 59, 73, 75, 82, 94, 96, 97, 107, 119, 136, 142, 150, 155, 156, 158, 163, 168, 169, 178, 179, 191, 195, 213, 214, 225, 230, 256, 258, 270, 294, 314, 317, 325, 362, 385, 389, 392, 395, 403, 404, 423, 424, 426, 433, 436, 439, 446, 449, 466	583,590500	53,0747
PRIM/VND – T4	2, 4, 5, 13, 15, 16, 21, 26, 31, 33, 40, 50, 59, 73, 82, 94, 96, 97, 111, 115, 136, 142, 150, 154, 155, 156, 163, 168, 169, 179, 194, 195, 209, 214, 230, 254, 256, 267, 270, 282, 314, 321, 354, 362, 385, 392, 395, 396, 403, 404, 423, 424, 426, 436, 437, 443, 446, 449, 466	586,156700	90,040

Fonte: Próprio autor

A melhor solução final fornecida pela técnica proposta neste trabalho em relação às soluções iniciais base e a obtida pelo GRASP com o indicador de sensibilidade $\alpha = 0,4$ alcançou melhorias de 17,6857% e 11,0504%, respectivamente. A proposta implementada neste trabalho não conseguiu obter a melhor solução conhecida, que possui perdas elétricas de 581,549400 kW (POSSAGNOLO, 2015), mas a melhor solução alcançada possui um erro de somente 0,3447% em comparação com a melhor solução conhecida para este sistema. Propostas quanto ao possível alcance dessa solução serão discutidas posteriormente.

Ainda assim, a melhor solução alcançada neste trabalho, para esse sistema, só foi possível de ser obtida por meio da utilização de um indicador de aleatoriedade menor para o

GRASP, $\alpha = 0,1$. Isso fez com que se partisse de um espaço de solução ruim, perfazendo, então, um caminho de soluções no procedimento de BT que, piorando a solução durante tal procedimento, alcançou a solução indicada na Tabela 22.

Na Tabela 23 apresentam-se as quantidades de soluções geradas e avaliadas pelas técnicas de BT e PR e seus devidos tempos de execução em separado, com a SRV e CRV, o que faz com que se visualize melhor o rendimento de cada técnica.

Tabela 23 - Quantidades de soluções geradas/avaliadas, e tempos de execução das técnicas de BT e PR.

Técnicas Adotadas			Soluções geradas	Tempos de Execução (s)
	Solução Inicial	Heurísticas		
SRV	Base	BT	43861	1265,800
		PR	301	194,288
	PRIM	BT	29807	872,171
		PR	314	209,162
	GRASP ($\alpha = 0,4$)	BT	34707	1005,190
		PR	356	235,698
	GRASP ($\alpha = 0,1$)	BT	47135	1384,820
		PR	271	174,865
CRV	Base	BT	37596	1081,930
		PR	251	167,895
	PRIM	BT	26362	758,406
		PR	309	198,823
	GRASP ($\alpha = 0,4$)	BT	30810	929,966
		PR	317	205,060
	GRASP ($\alpha = 0,1$)	BT	44110	1269,270
		PR	289	186,302

Fonte: Próprio autor

Pela Tabela 23 a técnica de redução de vizinhança não foi tão efetiva quanto vista nos outros sistemas testados. Aqui a redução alcançou patamares de 10% a 15,66% entre os procedimentos de BT executados SRV e CRV. Um maior ajuste na técnica de redução de vizinhança pode vir a trazer uma maior redução de soluções, mas pode vir a não avaliar soluções que, futuramente, pudessem trazer melhores soluções.

Os tempos obtidos pela técnica de PR estão de acordo com as quantidades de soluções obtidas. Podemos ver isso por meio da comparação com os tempos e quantidade de soluções obtidas com os outros sistemas testados. Isso mostra que mesmo que tal técnica possua

tempos altos para uma quantidade muito menor de soluções avaliadas, possui um comportamento estável.

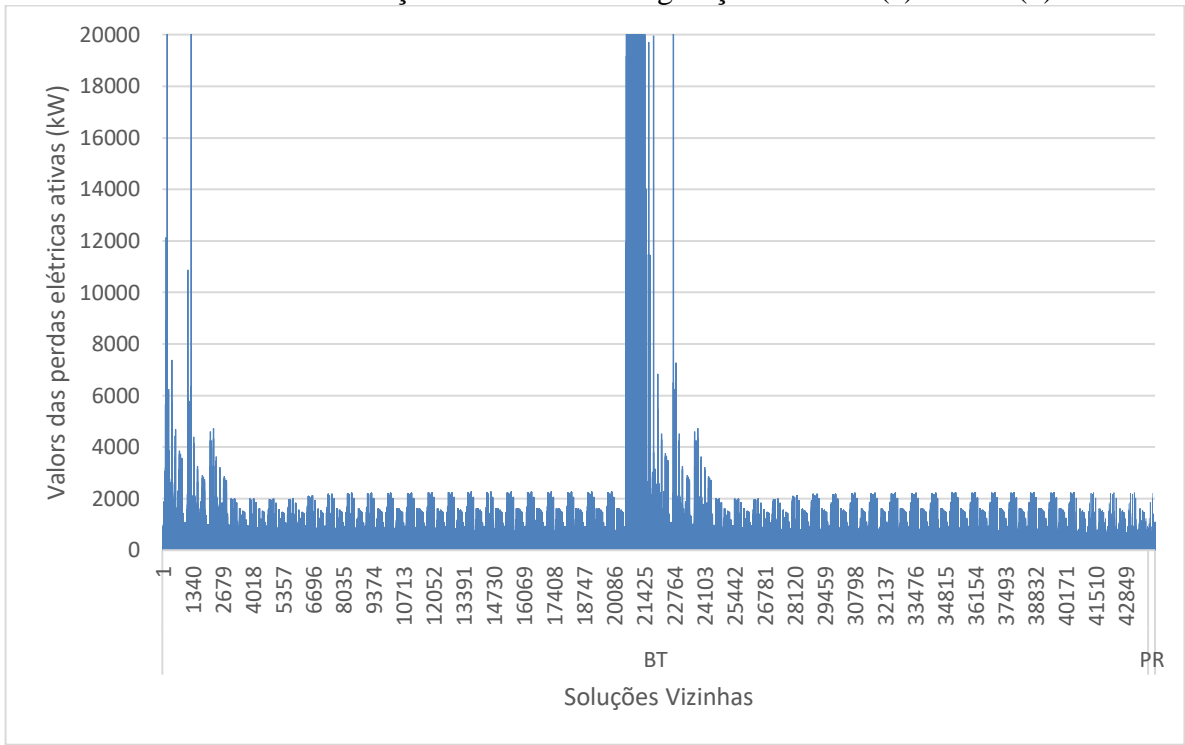
Os tempos elevados de execução do procedimento de busca local realizado pela técnica BT, no entanto, podem ser explicados tanto pela quantidade de soluções geradas/avaliadas quanto pelo excesso de procedimentos recursivos executados. Isto foi discutido nos resultados obtidos nas Tabelas 7 e 8 que competem ao procedimento de construção e destruição da árvore dinâmica que armazena uma configuração radial, e para o procedimento de cálculo de fluxo de potência em uma árvore dinâmica, respectivamente.

Ainda, não é somente o comportamento dos procedimentos das Tabelas 7 e 8 (tempos de execução da árvore dinâmica e do cálculo do fluxo de potência, respectivamente) que fazem este tempo de execução ser alto. Têm-se diversos outros elementos que corroboram para tal fator. Alguns deles são a forma de implementação; o programa estar em funções, ou seja, não são somente as chamadas de funções para a construção e destruição da árvore dinâmica e o cálculo do fluxo de potência que possuem chamadas recursivas; a avaliação de cada solução para adentrar no conjunto de soluções elite; a atualização do conjunto de soluções elite; condições no pipeline de execução, devido à estruturas condicionais; atualização do vetor de ramos bloqueados; e cópias de configurações encontradas para se reservar como a melhor ou a pior visitada.

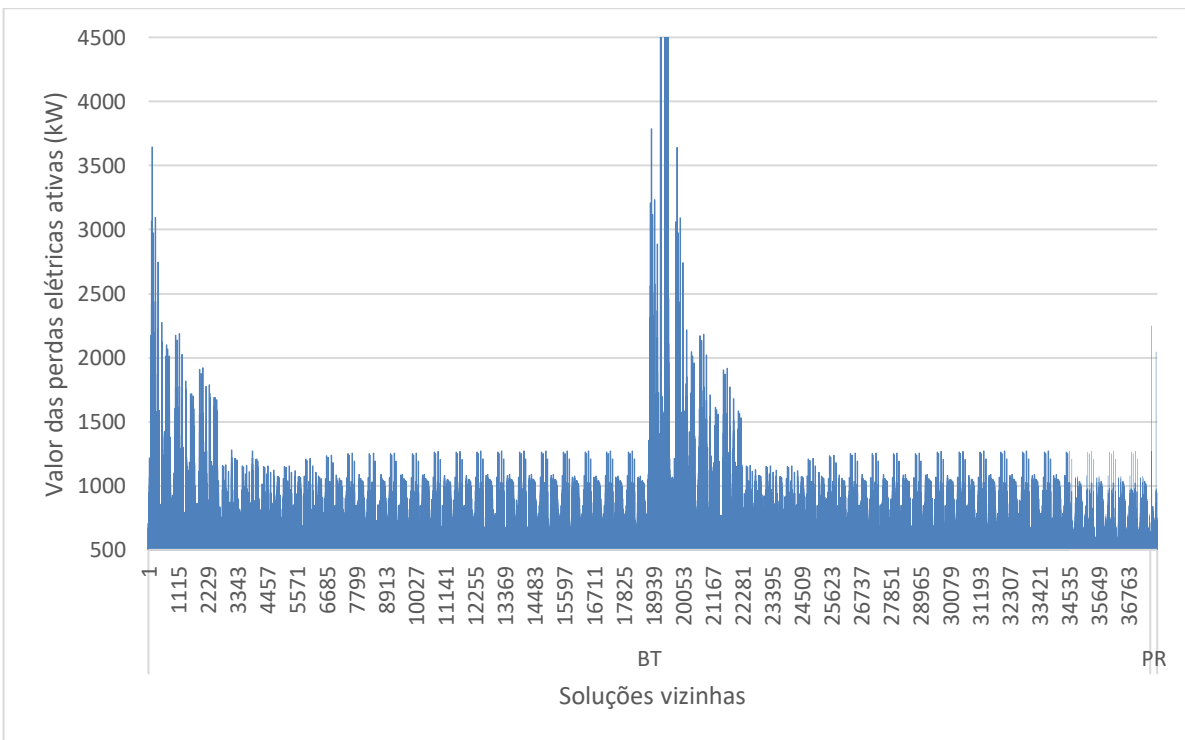
Na Figura 49 apresentam-se os gráficos que representam todas as soluções geradas e avaliadas pelo procedimento BT/PR, utilizando a solução base como configuração inicial. Na Figura 49(a) apresenta-se SRV e na Figura 49(b) apresenta-se CRV. As Figuras 50, 51 e 52 apresentam os gráficos com as mesmas características que as da Figura 49, mas para as diferentes estratégias de soluções iniciais adotadas, sendo o PRIM, GRASP com indicador de sensibilidade $\alpha = 0,4$ e GRASP com o indicador de sensibilidade $\alpha = 0,1$, respectivamente.

As Figuras 49 e 52, assim como as mesmas figuras que mostram as características para com os outros testes avaliados, evidenciaram que, dependendo do sistema adotado, a técnica de redução de vizinhança além de reduzir consideravelmente a quantidade de soluções geradas/avaliadas, também fez com que os espaços de busca percorridos, no procedimento de busca local realizado pela BT, fossem por meio de soluções de melhor qualidade.

Figura 49 – Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 415 barras com a solução base como configuração inicial – (a) SRV – (b) CRV.

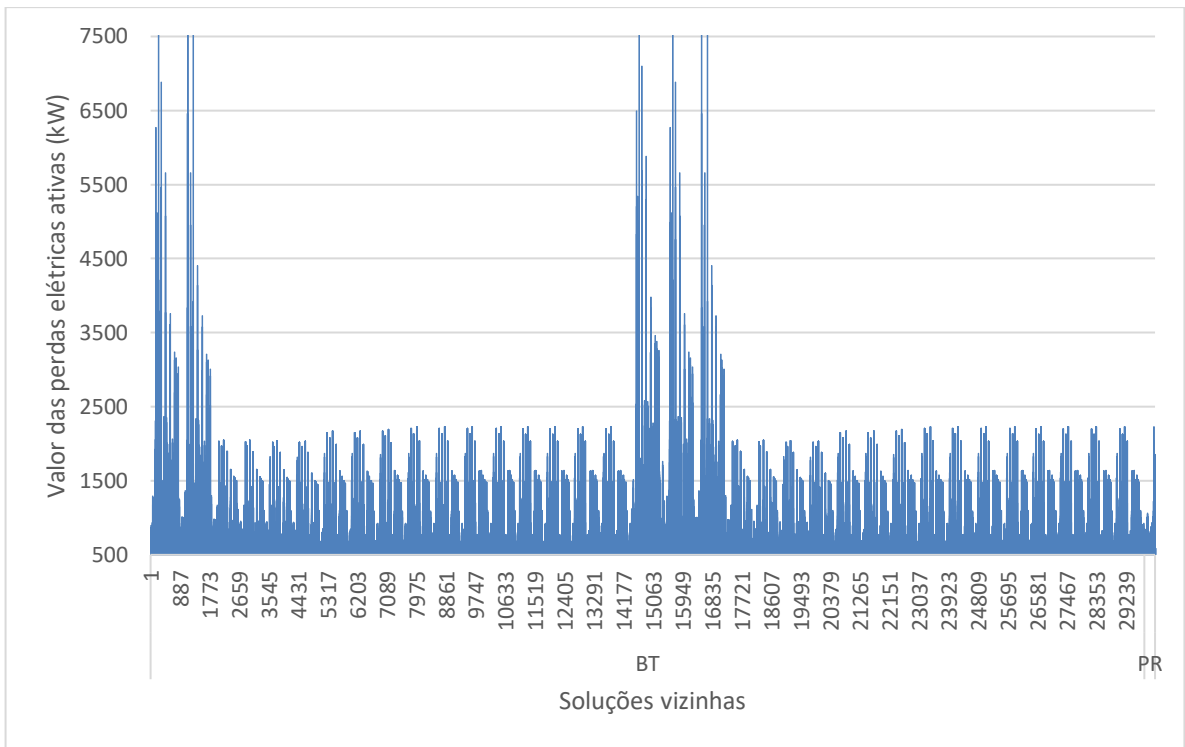


(a)

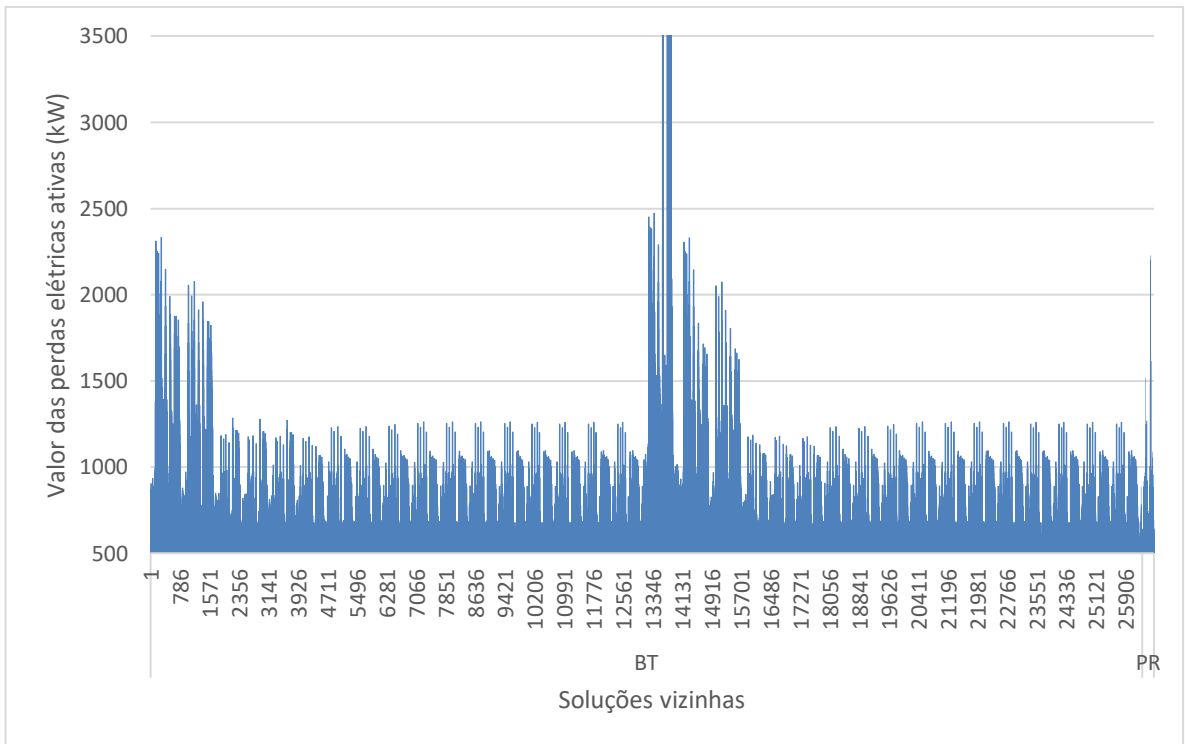


Fonte: Próprio autor

Figura 50 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 415 barras com o PRIM como configuração inicial – (a) SRV – (b) CRV.



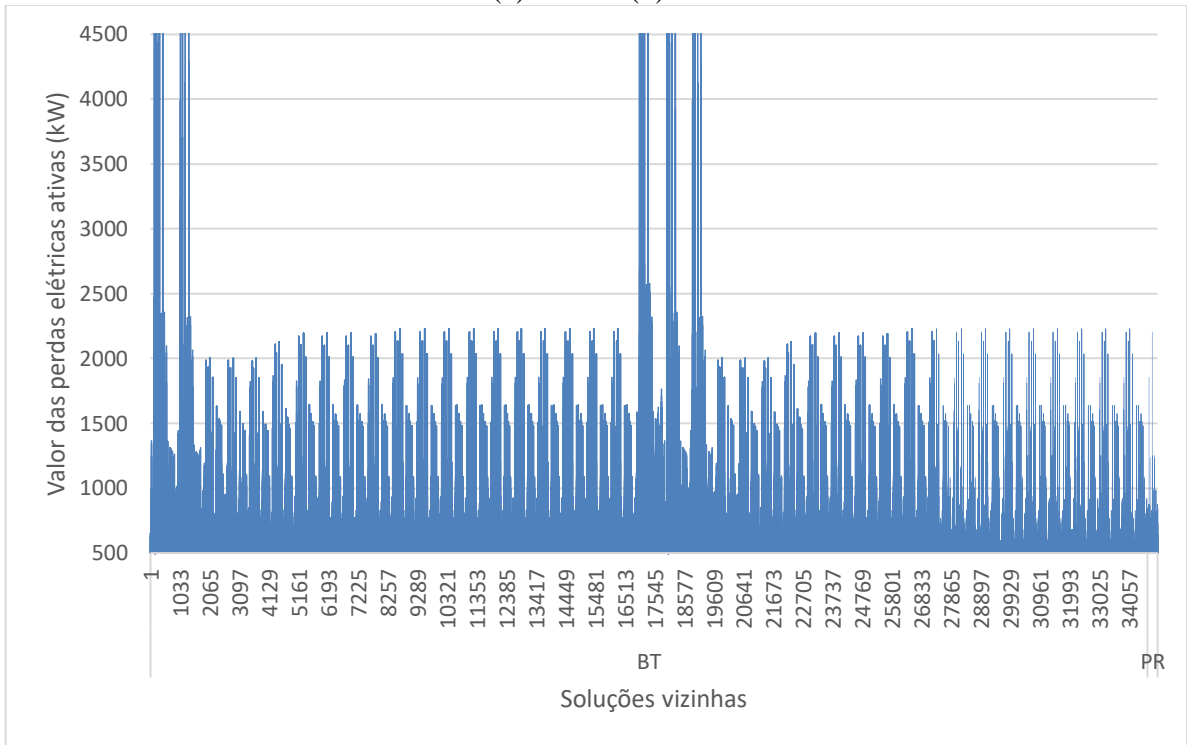
(a)



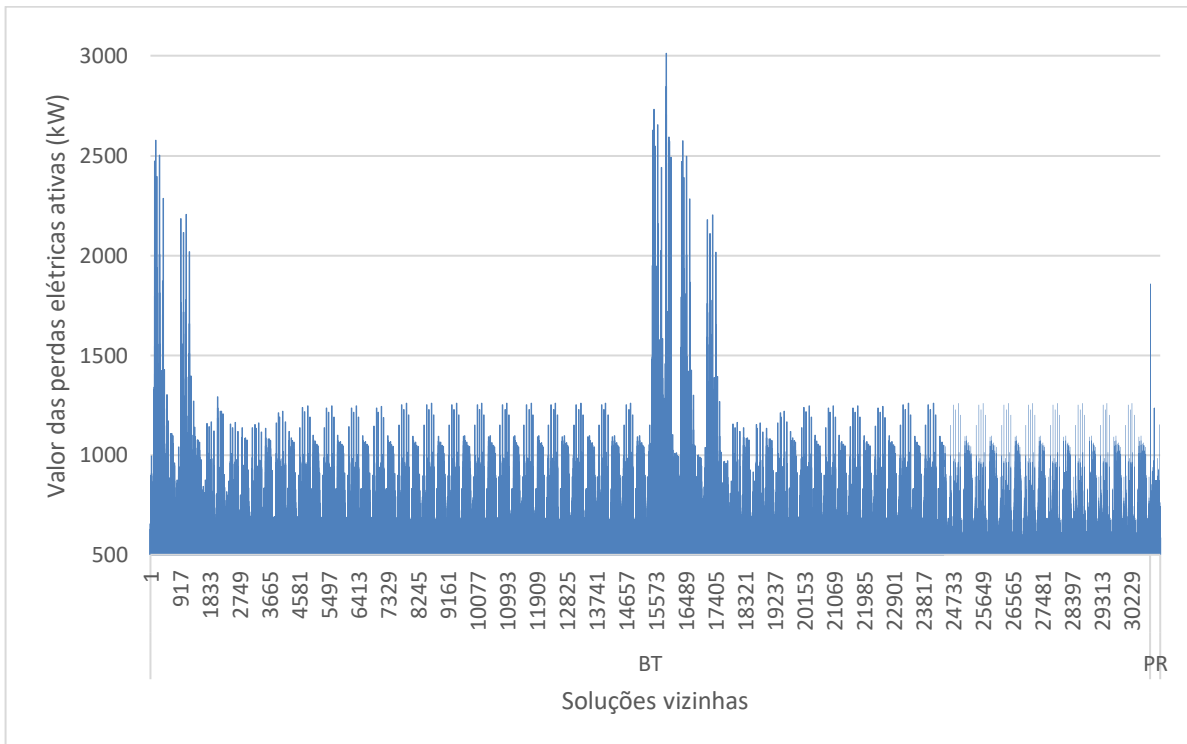
(b)

Fonte: Próprio autor

Figura 51 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 415 barras com o GRASP como configuração inicial e indicador de sensibilidade $\alpha = 0,4$ – (a) SRV – (b) CRV.



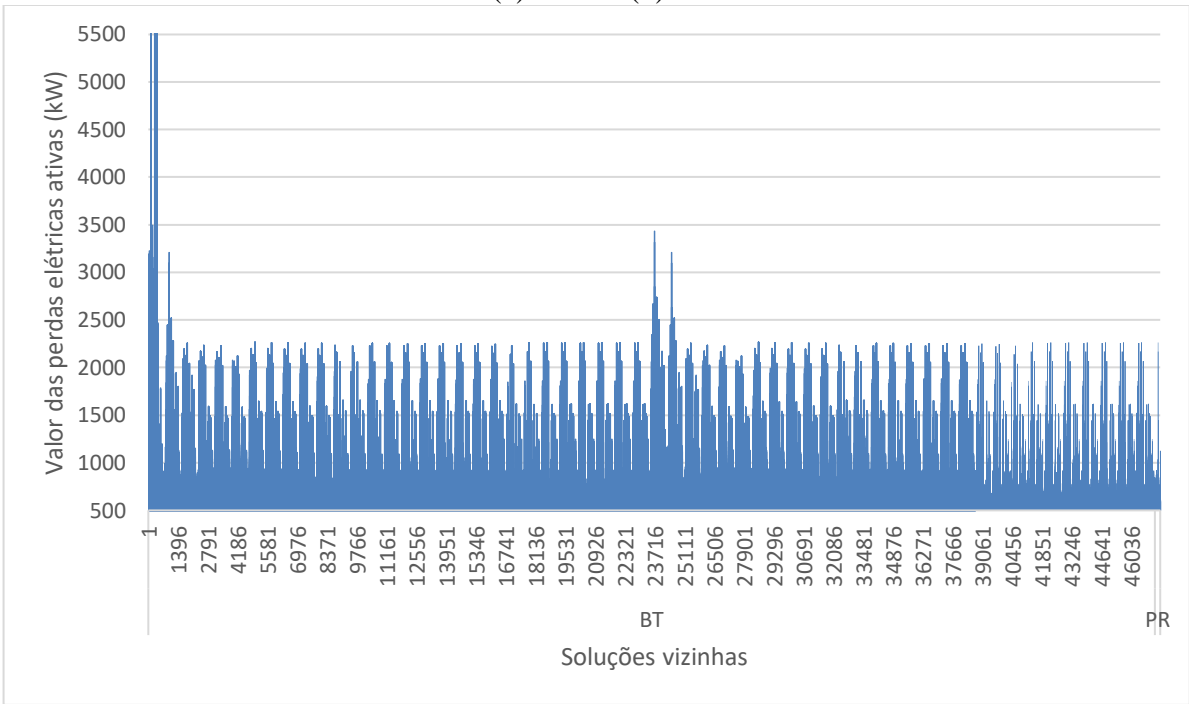
(a)



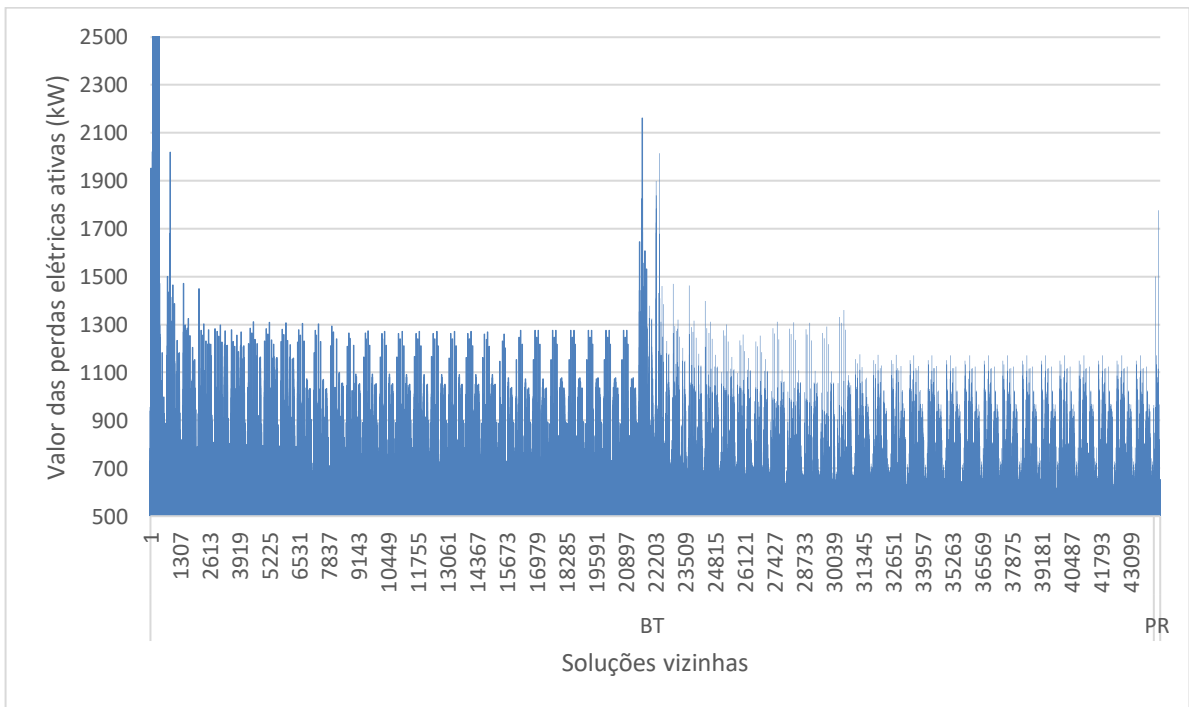
(b)

Fonte: Próprio autor

Figura 52 - Histórico de soluções geradas/avaliadas pelo procedimento BT/PR para o sistema de 415 barras com o GRASP como configuração inicial e indicador de sensibilidade $\alpha = 0,1$ – (a) SRV – (b) CRV.



(a)

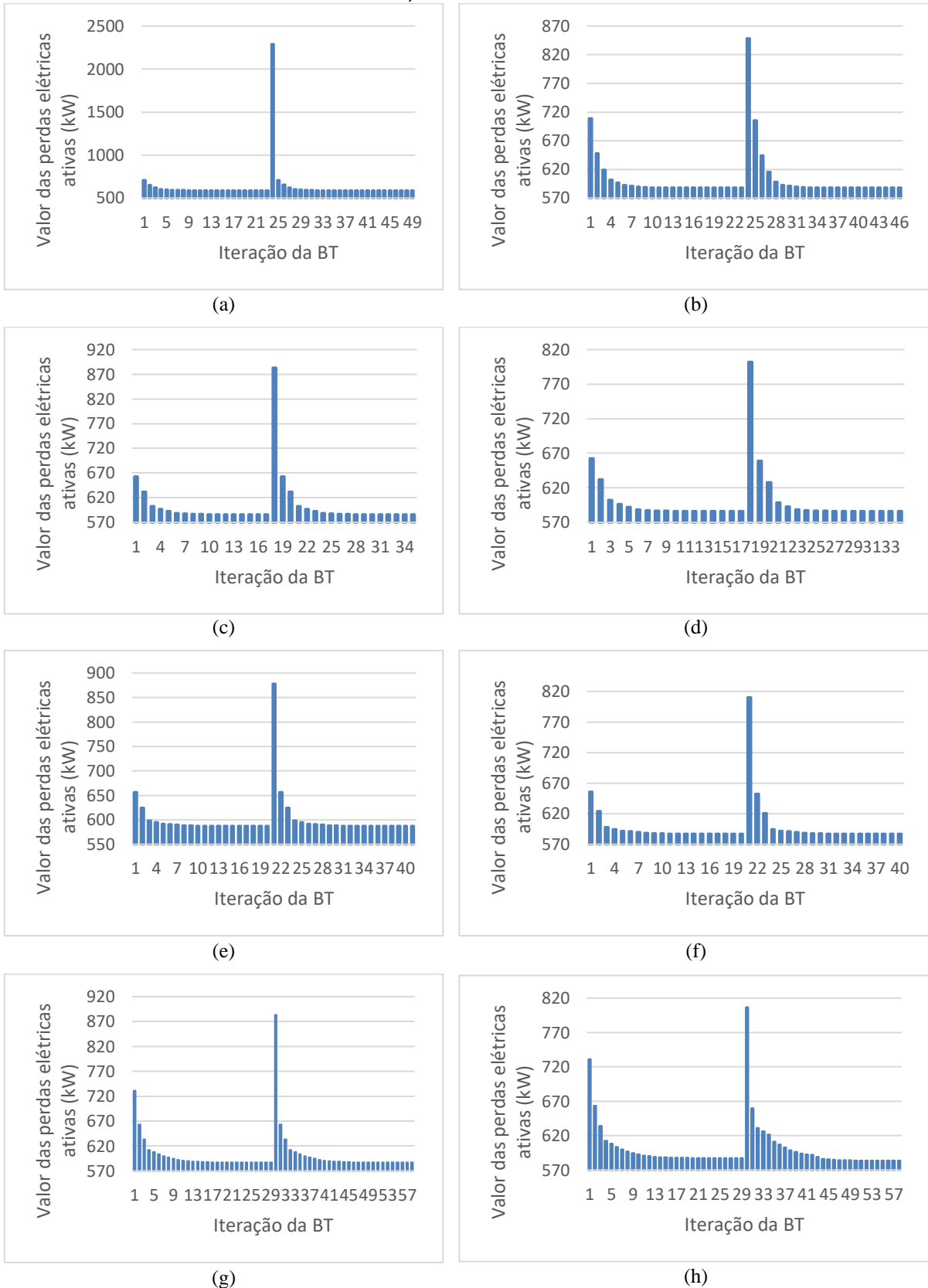


(b)

Fonte: Próprio autor

Na Figura 53 apresentam-se os gráficos referentes às melhores soluções obtidas durante o procedimento de busca local realizado pela BT para cada estratégia de obtenção de uma configuração inicial, variando-se o indicador de sensibilidade do GRASP SRV e CRV.

Figura 53 - Melhores soluções da BT – (a)-(b) Solução base como configuração inicial e SRV/CRV – (c)-(d) PRIM como configuração inicial SRV/CRV - (e)-(f) GRASP como configuração inicial, $\alpha = 0,4$ SRV/CRV – (g)-(h) GRASP como configuração inicial, $\alpha = 0,1$ SRV/CRV.



Fonte: Próprio autor

Verifica-se que para todas as figuras de todos os testes pertinentes aos históricos de soluções geradas/avaliadas pelo procedimento BT/PR encontra-se os momentos de cada iteração do procedimento de BT. Existe muita semelhança, nos gráficos, nas soluções obtidas entre uma iteração e outra. Esse comportamento é mais evidente à medida que se aumenta a dimensão do sistema.

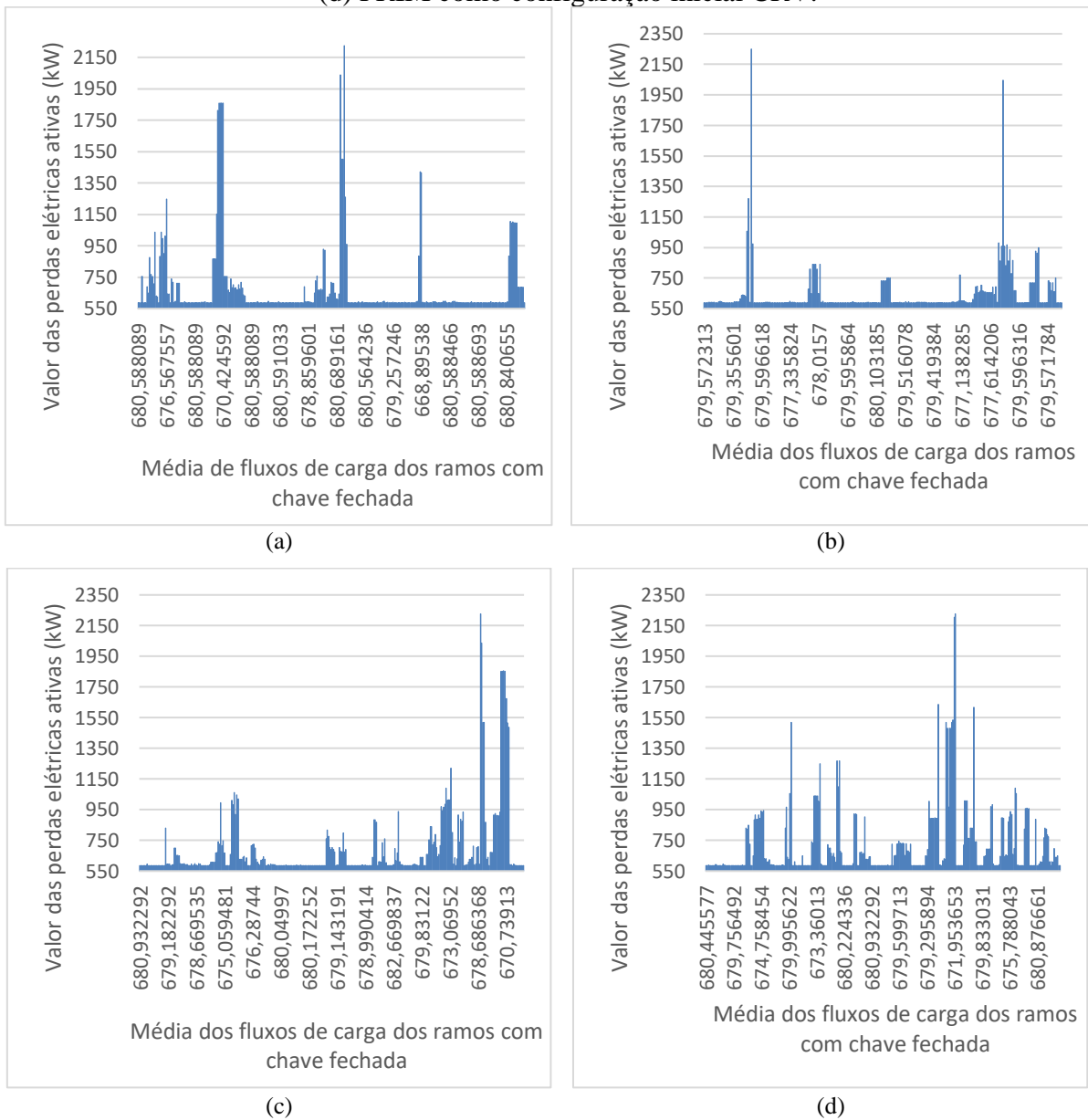
Durante todos os testes realizados verificou-se que, conforme se aumenta a dimensão do sistema, soluções muito próximas, ou seja, soluções com somente dois ou mais status de chaves diferentes nos ramos, mas com perdas elétricas ativas idênticas ou próximas na casa de 10^{-2} a 10^{-4} . É certo que para várias iterações da BT muitas soluções idênticas serão avaliadas, uma vez que a estratégia escolhida para se gerar a vizinhança possui um movimento muito sutil, mas eficaz, e que por meio dele, geram-se soluções idênticas entre iterações diferentes da BT.

Pode-se verificar um comportamento semelhante nos testes realizados para o sistema de 136 barras. Não há grandes mudanças nas soluções geradas/avaliadas no procedimento de busca local, somente ao piorar a solução durante o procedimento de BT. O conjunto de soluções antes e depois de se piorar a solução corrente muda sutilmente para o sistema para cada teste realizado. Nas Figuras 53(a) e 53(d) verifica-se que há uma redução, muito pequena, na quantidade de iterações totais da BT, enquanto que para as Figuras 53(e) e 53(h) o comportamento não ocorre. Isso é devido aos espaços de busca visitados durante o procedimento de BT conterem soluções de boa qualidade. Para estes espaços não ocorreram mudanças nas soluções correntes de cada iteração da BT.

Visualiza-se, também, a rapidez de convergência da proposta nas cinco primeiras iterações do procedimento de BT, tanto no início das iterações quanto logo após se piorar a solução corrente. O mesmo comportamento pode ser verificado em todos os testes realizados neste trabalho. Isso mostra que o procedimento de vizinhança é eficaz. Como o sistema possui uma capacidade combinatória muito grande, após uma convergência rápida, o caminho de soluções geradas/avaliadas pelos espaços de busca, torna-se muito lento porque as mudanças na função objetivo entre cada solução corrente de cada iteração tornam-se muito próximas, da casa de 10^{-2} a 10^{-5} , até a estagnação, que ocorre quando em um determinado espaço de busca não se encontra uma solução melhor do que a solução corrente. Este comportamento pode ser visualizado em todos os gráficos, de todos os testes realizados, para o procedimento de BT.

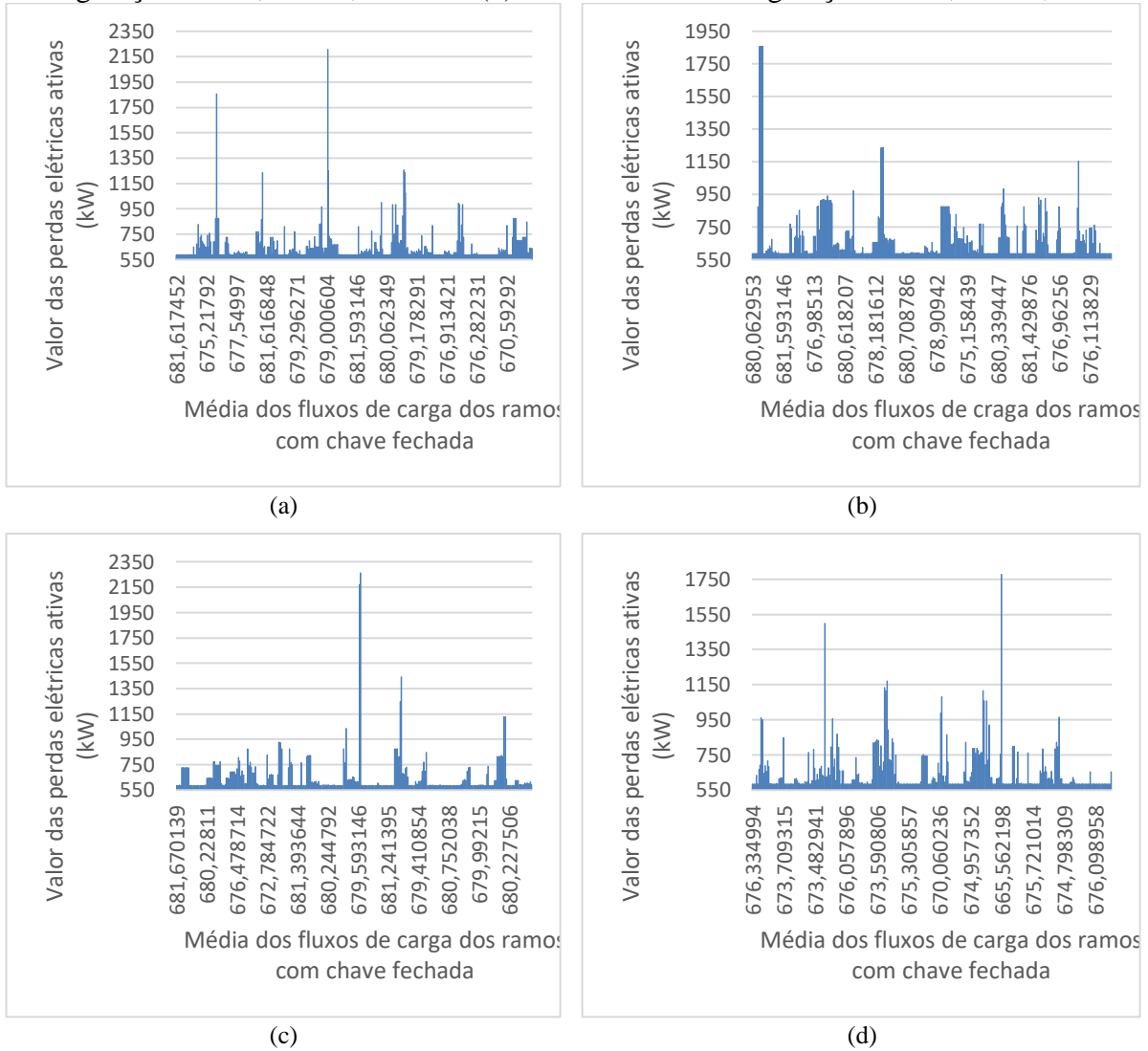
Nas Figuras 54 e 55 apresentam-se os históricos de soluções obtidas durante o procedimento de PR, partindo-se de cada estratégia de obtenção de uma configuração inicial com a SRV e CRV. As Figuras 54(a) e 54(b) possuem como configuração inicial a solução base; as Figuras 54(c) e 54(d) possuem como configuração inicial o PRIM; e as Figuras 55(a) e 55(d) possuem como configuração inicial o GRASP com os dois indicadores de sensibilidade utilizados, $\alpha = 0,4$ e $\alpha = 0,1$.

Figura 54 - Histórico de soluções do PR – (a) Solução base como configuração inicial SRV – (b) Solução base como configuração inicial CRV – (c) PRIM como configuração inicial SRV – (d) PRIM como configuração inicial CRV.



Fonte: Próprio autor

Figura 55 - Histórico de soluções do PR – (a) GRASP como configuração inicial, $\alpha = 0,4$ SRV – (b) GRASP como configuração inicial, $\alpha = 0,4$ CRV – (c) GRASP como configuração inicial, $\alpha = 0,1$ SRV – (d) GRASP como configuração inicial, $\alpha = 0,1$ CRV.



Fonte: Próprio autor

Verifica-se na Figura 54 e na Figura 55, que para todos os testes executados, com a técnica de PR as soluções se mantiveram, em sua grande parte, muito próximas à melhor solução, ou soluções muito próximas a esta. O mesmo comportamento pode ser visualizado em todos os testes deste trabalho para o histórico de soluções geradas pelo PR. Isso se justifica pela forma como é construído o conjunto de soluções elite. Conjunto este que é formado por até as dez melhores soluções encontradas durante todo o procedimento de BT. Assim, o caminho de soluções geradas/avaliadas entre o conjunto de soluções inicial e guia se mantém, em sua grande maioria, em soluções de boa qualidade.

Não foi obtida uma melhor solução do que a encontrada na literatura pela técnica proposta. Estratégias de como o PR podem vir a apresentar um melhor rendimento bem como poder alcançar a melhor solução já encontrada na literatura, será explicado ao final deste capítulo e nos estudos futuros.

5.4 REDUÇÃO DE VIZINHANÇA

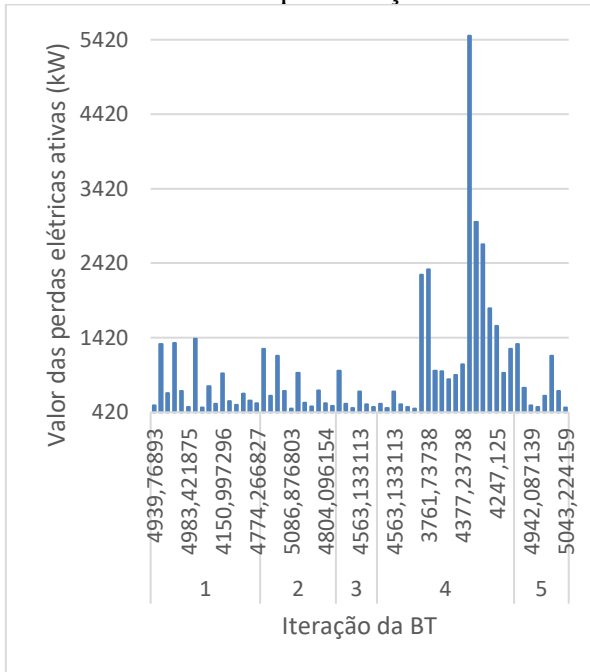
A estrutura de vizinhança adotada para este trabalho, como explicada anteriormente, trabalha com a média dos fluxos de potência aparente nos ramos com chave fechada em uma rede malhada. Esta mesma estratégia é utilizada na construção de soluções no GRASP. Esta estratégia indica que soluções com altas médias perfazem soluções de boa qualidade. Os gráficos mostrados adiante ilustram esse comportamento.

Pela Figura 56(b) e Figura 56(d) se assume que as melhores configurações radiais se encontram em soluções que possuem altas médias de somatório do fluxo de carga nos ramos com chave fechada em rede malhada. Com essas confirmações foi implementada neste trabalho a técnica de redução de vizinhança. Na Figura 57 verifica-se estes mesmo comportamentos para o sistema de 33 barras.

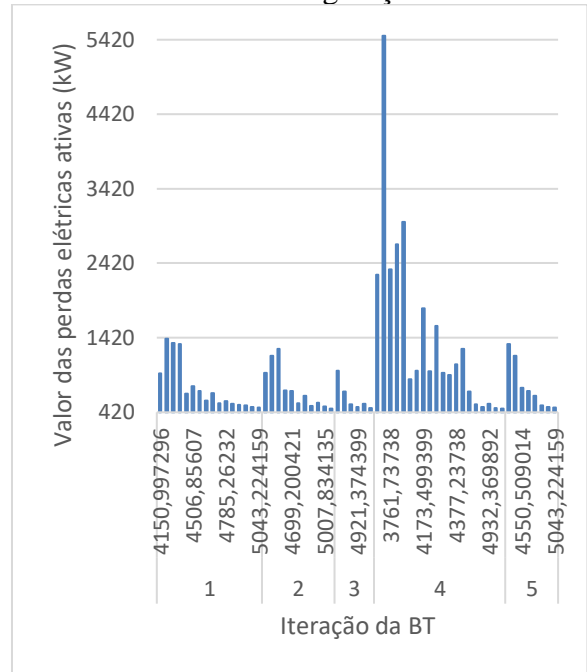
Contra preceitos do problema de RSDEE uma solução com uma média alta não é certo de que ela sempre possuirá baixas perdas, mesmo que os gráficos da Figura 56 e da Figura 57 tenham comprovado isto, pois para um ramo com determinado fluxo de potência, no momento do cálculo das perdas elétricas a tensão e a corrente mudam constantemente até que o processo alcance seu limite de convergência (estratégia *backward/forward*). Sendo assim, não é certo esperar que as perdas sejam derivadas desta média. Mesmo assim os testes executados comprovaram que esse comportamento pode ser esperado de certos sistemas. Na Figuras 58 apresentam-se os gráficos pertinentes para o sistema de 84 barras.

Como pode ser notado, o comportamento continua o mesmo não importando o sistema adotado. Privar-se-á de ilustrar os dados pertinentes aos sistemas de 136 e 415 barras porque estes apresentam comportamentos semelhantes aos apresentados pelos sistemas de 14, 33 e 84 barras.

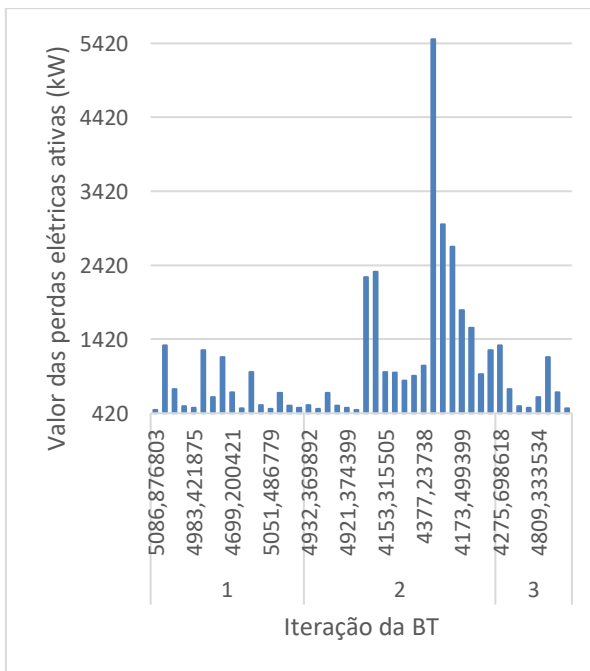
Figura 56 - Histórico de soluções do procedimento de BT para o sistema de 14 barras – (a) Soluções geradas por iterações consecutivas com a solução base como configuração inicial – (b) Soluções ordenadas em ordem decrescente de média dos fluxos de carga dos ramos com chave fechada por iteração da BT com a solução base como configuração inicial – (c) Soluções geradas por iterações consecutivas com PRIM/GRASP como configuração inicial – (d) Soluções ordenadas em ordem decrescente de média dos fluxos de carga dos ramos com chave fechada por iteração da BT com PRIM/GRASP como configuração inicial.



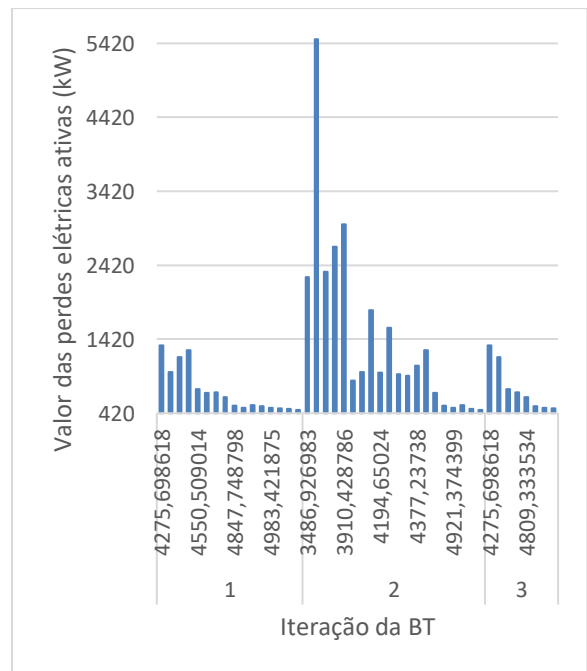
(a)



(b)



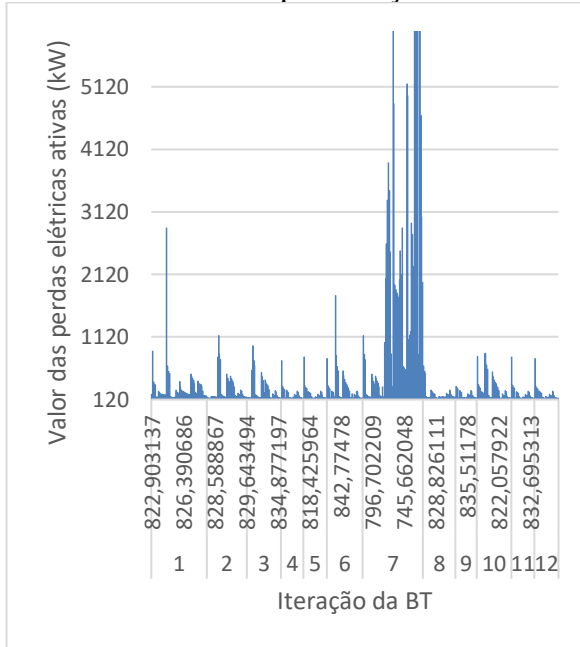
(c)



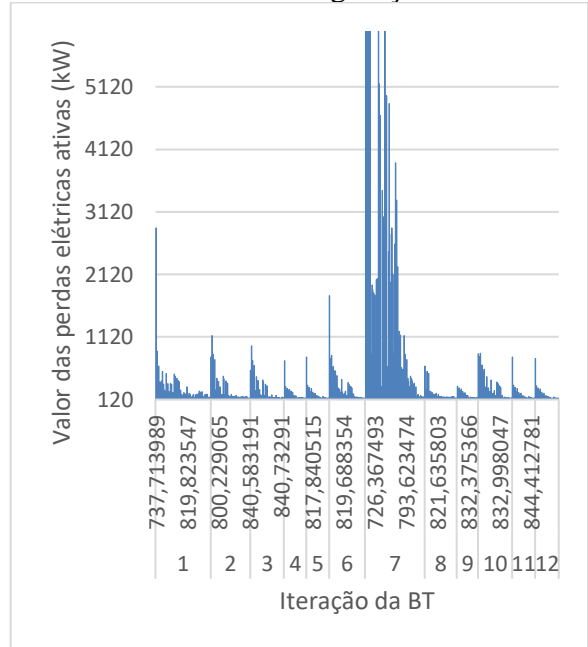
(d)

Fonte: Próprio autor

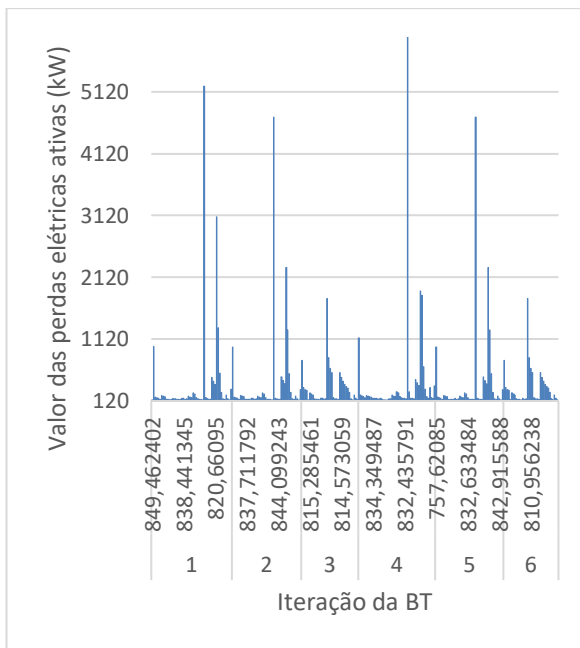
Figura 57 - Histórico de soluções do procedimento de BT para o sistema de 33 barras – (a) Soluções geradas por iterações consecutivas com a solução base como configuração inicial – (b) Soluções ordenadas em ordem decrescente de média dos fluxos de carga dos ramos com chave fechada por iteração da BT com a solução base como configuração inicial – (c) Soluções geradas por iterações consecutivas com PRIM/GRASP como configuração inicial – (d) Soluções ordenadas em ordem decrescente de média dos fluxos de carga dos ramos com chave fechada por iteração da BT com PRIM/GRASP como configuração inicial.



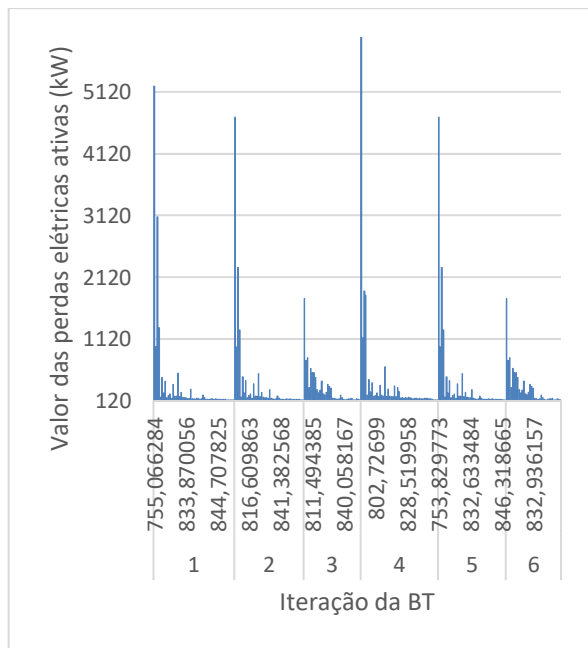
(a)



(b)



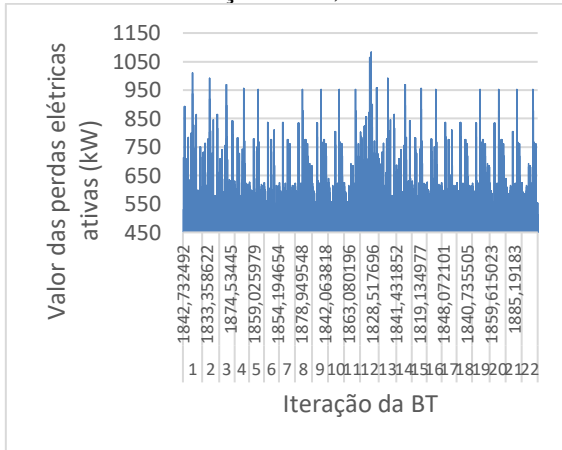
(c)



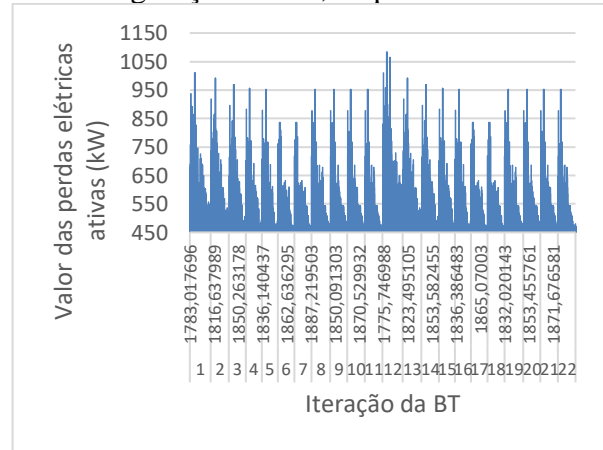
(d)

Fonte: Próprio autor

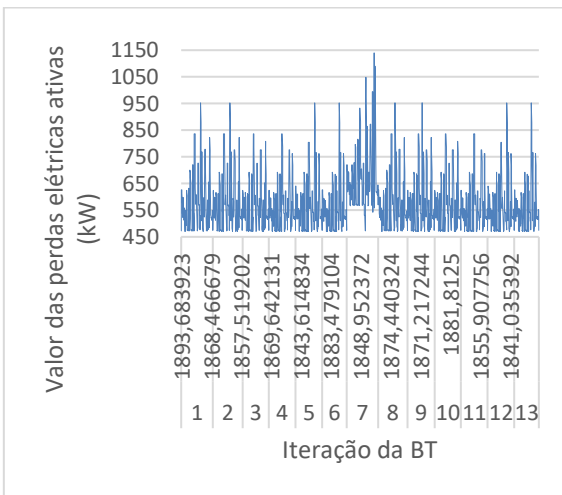
Figura 58 - Histórico de soluções do procedimento de BT para o sistema de 84 barras – (a)-(e) Soluções geradas por iterações consecutivas com a solução base, PRIM e GRASP como configuração inicial, respectivamente – (b)-(d)-(f) Soluções ordenadas em ordem decrescente de média dos fluxos de carga dos ramos com chave fechada por iteração da BT com a solução base, PRIM e GRASP como configuração inicial, respectivamente.



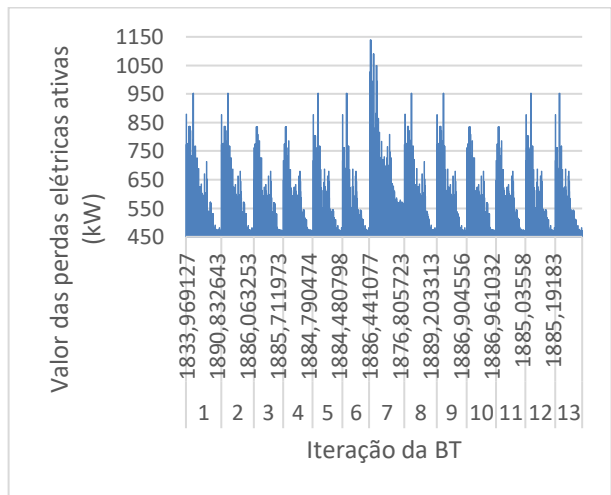
(a)



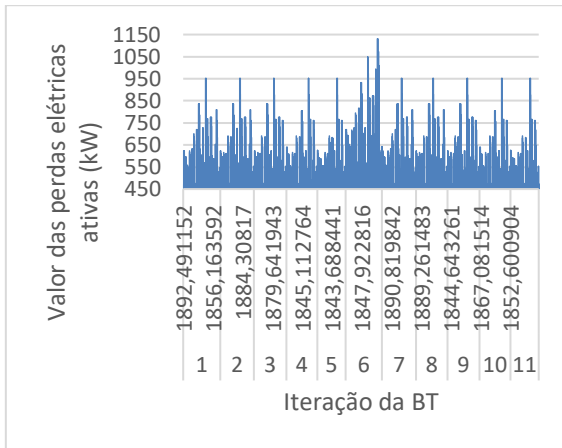
(b)



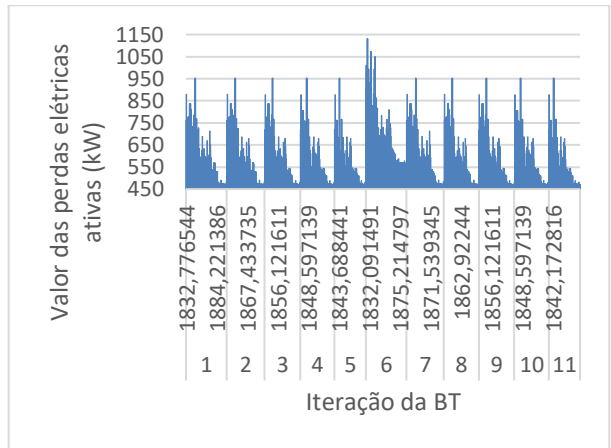
(c)



(d)



(e)



(f)

Fonte: Próprio autor

5.5 DISCUSSÕES DOS RESULTADOS

Inicialmente, os resultados obtidos através da implementação computacional desta proposta mostraram-se válidos e coerentes, de acordo com propostas já apresentadas e validadas (ISABELLA F. FEITEIRA; LEONARDO H. MACEDO; RUBÉN ROMERO, 2018; POSSAGNOLO, 2015; RENATO CESPEDES G., 1990; SHIRMOHAMMADI, 1988). A metodologia implementada traz algumas vantagens quanto aos algoritmos clássicos de tratamento de um problema RSDEE e cálculo de fluxo de potência, bem como algumas desvantagens. A seguir se relatam alguns aspectos observados sobre o problema de RSDEE:

- Com a utilização da árvore dinâmica não há mais a necessidade do procedimento de renumeração das barras a cada configuração diferente de uma solução encontrada, pois são os procedimentos de percurso da árvore que regem a forma como as barras e ramos serão percorridos;
- O cálculo das perdas de uma solução pelo procedimento do cálculo do fluxo de potência realiza seus métodos *backward/forward* lendo a árvore somente uma vez por método, diferentes do procedimento clássico calculado em um vetor ou matriz estáticos;
- A árvore dinâmica criada representa fielmente, na memória do computador, uma distribuição espacial do problema de RSDEE;
- Os tempos computacionais altos obtidos derivam-se da estratégia de se piorar a solução durante o procedimento de BT, que reinicia seu procedimento; a estratégia de implementação do PR e, para sistemas de grande porte, a sobrecarga de procedimentos recursivos;
- O algoritmo encontra-se implementado por meio do paradigma estruturado e está devidamente dividido em funções facilitando a inserção de métodos de paralelização, o que reduziria os tempos computacionais encontrados;
- É apresentado um conjunto de diferentes estruturas de dados para trabalho do problema de RSDEE;

Para os procedimentos de construção e destruição da árvore dinâmica e para o cálculo do fluxo de potência de uma configuração radial em árvore dinâmica os tempos de execução altos são devido aos procedimentos recursivos implementados, como já tratado acima. Os tempos de execução para ambas as estratégias mencionadas quando aplicadas ao sistema de

136 barras já se apresenta muito próximo em relação aos tempos das técnicas comparativas de RENATO CESPEDES G., 1990 e SHIRMOHAMMADI, 1988, mostrando que a característica exponencial da técnica implementada começa a perder rendimento a partir de sistemas de médio porte possuindo por volta de 140 barras, 25 chaves abertas e 164 ramos.

A sobrecarga de procedimentos recursivos para ambas as técnicas pode ser aliviada por meio de uma reestruturação de código onde as funções podem ser construídas sem o princípio da recursividade, tornando-as iterativas. A implementação recursiva necessita de registrar a situação do processamento de forma que a implementação possa continuar a partir do ponto que parou após o término de cada nova rotina gerada pelo procedimento recursivo. Essa tarefa consome tempo de processamento e memória, pois para cada chamada recursiva todo um subconjunto de dados é novamente alocado. Assim, a implementação iterativa tende a ser mais rápida do que a recursiva por não necessitar de construir novos elementos na memória e não precisar controlar as chamadas recursivas e controlar as situações do processamento de cada chamada. Esta reestruturação de código não tende a otimizar o tempo de processamento somente dos procedimentos de construção e de destruição da árvore dinâmica e do cálculo de fluxo de potência, mas da proposta deste trabalho como um todo.

O procedimento de obtenção de uma configuração inicial por meio do GRASP apresentou tempos de execução incompatíveis quando executado uma e dez vezes. A execução de uma iteração mostrou ser muito rápida, mesmo para sistemas de grande porte, mas quando iterada dez vezes o tempo não aumentou em uma mesma proporção. Esta característica aparece devido à estratégia aleatória da meta-heurística e de como a mesma foi implementada. A linguagem C possui a funcionalidade *rand()* que produz números aleatórios, mas não com uma grande eficiência. Em execuções sucessivas os mesmos valores são obtidos, a não ser que se passe uma *semente* diferente a cada geração de um novo valor aleatório. A semente utilizada foi o próprio *clock* do computador.

Ainda assim, o tempo entre a geração de uma solução e outra para o GRASP era muito rápido, e os elementos se repetiam, trazendo de três a cinco soluções idênticas dentro de dez iterações, indicando que as soluções idênticas geradas compartilhavam de um mesmo *clock*. A solução proposta foi que a cada iteração do GRASP o sistema suspendesse a execução do código durante milissegundos e continuasse, então, para a próxima iteração. Isso fez com que a função de aleatoriedade do GRASP pudesse ter tempo de atualizar sua semente com um novo *clock* e construísse diferentes elementos para sua RCL. Desta forma os tempos de execução do GRASP para dez iterações aumentaram devido a esse estado de “espera” do

programa por um novo *clock*. Estratégias mais eficazes de aleatoriedade podem ser estudadas futuramente.

O procedimento de busca local realizado através da BT obteve as melhores soluções já conhecidas para os sistemas de pequeno e médio porte (14, 33, 84 e 136 barras) antes de piorar a solução pelo método de intensificação. Somente o sistema de grande porte (415 barras) necessitou de tratamentos específicos como partir de soluções iniciais de baixa qualidade e intensificação. Assim, os tempos de execução da técnica de BT poderiam ser considerados praticamente a metade para os sistemas de pequeno e médio porte. A inserção de mais estratégias pertinentes à BT pode vir a ser implementadas para que se melhore tal procedimento meta-heurístico e não seja necessário realizar a intensificação, desta maneira, podendo alcançar de forma mais rápida as melhores configurações.

A técnica de diversificação implementada na BT por meio de bloqueio de movimentos de ramos que tenham acabado de mudar seus status de chave ajudou a reduzir o espaço de busca, uma vez que se têm sempre ramos com status de chave aberta proibidos de realizar movimentos, o que reduz o espaço de soluções do sistema. A técnica de intensificação implementada na BT veio para forçar o espaço de busca, após uma estagnação, a partir para um espaço de soluções de baixa qualidade a fim de que novos espaços de busca pudessem ser analisados na tentativa de se encontrar melhores soluções.

Uma vez que a BT já encontrava as melhores soluções, logo antes de se intensificar o procedimento, para os sistemas de 14, 33, 84 e 136 barras, isso consumia considerável tempo computacional. Para o sistema de 415 barras, essa estratégia mostrou-se válida, uma vez que a melhor solução obtida só foi obtida após a intensificação., o que indica que, para sistemas de grande porte, a técnica traz resultados interessantes.

A técnica de redução de vizinhança trouxe resultados satisfatórios. A utilização da média de fluxos de carga dos ramos com chave fechada em uma rede malhada partiu de sua utilização como estratégia de construção de soluções de boa qualidade no GRASP. Inicialmente o GRASP era somente para construir as soluções por meio dos somatórios dos fluxos de carga aparente de uma rede malhada e se estendeu para ser utilizada a média deste somatório neste trabalho.

Durante o cálculo do fluxo de potência, as tensões e correntes mudam a cada iteração do procedimento *backward/forward* e assim o fluxo de potência. Não seria correto afirmar

que um determinado ramo possui perdas elétricas específicas, visto que os elementos de corrente e tensão mudam constantemente.

Após testes com o GRASP a proposta de utilização da média do somatório dos fluxos de potência aparente nramos com chave fechada em uma rede malhada provou ser coerente, e esta foi utilizada como técnica de redução de vizinhança, uma vez que se verificou que quanto maior a média, menores são as perdas.

A aplicação da técnica de redução de vizinhança é rápida, necessitando-se somente executar o somatório e o média em um vetor estático e linear. A comparação é realizada entre a solução corrente e cada solução vizinha gerada. Novas estruturas de redução de vizinhança podem vir a surtir diferentes efeitos no histórico de soluções geradas/avaliadas (POSSAGNOLO, 2015).

A técnica de PR utilizada neste trabalho teve como principal função um procedimento de pós-otimização por meio de um conjunto de soluções elite que é composta pelas 10 melhores soluções obtidas em todo o procedimento de busca local realizado pela BT. Esta estratégia tem como objetivo encontrar diferentes soluções não visitadas por meio de uma busca por diferentes espaços de busca/soluções para que estas soluções pudessem vir a serem melhores do que a incumbente.

Uma vez encontradas as melhores configurações para os sistemas de 14, 33, 84 e 136 barras, o PR serviu para experimentação de soluções geradas. Mesmo para o sistema de 415 barras a melhor configuração conhecida não foi obtida. Estratégias quanto à entrada de novas soluções no conjunto de soluções elite podem ajudar para de se manter soluções com uma maior diversidade como, por exemplo, se permitir, entre as dez soluções armazenadas, as cinco piores e as cinco melhores soluções geradas/avaliadas.

O baixo rendimento do PR, visto que em determinados sistemas o tempo de execução era muito maior com uma quantidade de soluções avaliadas muito menor, vem da característica aleatória da técnica. Esta utiliza a seleção de elementos aleatórios, também, por meio da função *rand()* e com isto aparecem os mesmos problemas que a técnica GRASP. Para cada geração de uma nova solução no PR são submetidos três procedimentos de escolha aleatória, e para cada escolha o sistema deve suspender a execução do código por milissegundos, a mesma quantidade utilizada no GRASP. A *semente* utilizada foi a mesma do GRASP.

6 CONSIDERAÇÕES FINAIS

Neste trabalho foram apresentadas formulações de algoritmos, baseadas em procedimentos heurísticos e meta-heurísticos para resolver o problema de RSDEE para um nível de demanda com o objetivo de encontrar uma solução que obtivesse as menores perdas elétricas. A estrutura de vizinhança possui uma estratégia de produzir somente soluções topologicamente factíveis (radiais) e o controle da aleatoriedade do algoritmo GRASP possui uma quantidade de parâmetros reduzidos.

Os resultados obtidos pela técnica e metodologia propostas mostraram-se coerentes quanto às configurações finais encontradas. A estratégia adotada para os pesos dos ramos para a aplicação do PRIM e do GRASP mostrou-se viável, pois constrói soluções iniciais de boa qualidade, compatíveis com as soluções encontradas na literatura, em tempos computacionais pequenos. Para os sistemas de pequeno porte avaliados (14, 33 e 84 barras), ambas as soluções iniciais (PRIM e GRASP) puderam alcançar a melhor solução.

Possagnolo (2015) utiliza a mesma estratégia de pesos nos ramos de ligação, mas por meio de diferentes fatores de fluxo de potência (aparente, ativa e reativa, tanto saindo de uma barra quanto chegando a uma barra) não obtendo resultados expressivos entre as diferentes aplicações.

Os tempos computacionais elevados em alguns momentos são devido à estratégia de intensificação, que pode ser revista; à estratégia de PR que pode ser reestruturada; e à sobrecarga de procedimentos recursivos a partir de sistemas de médio porte mostrando um baixo rendimento para o sistema de grande porte de 415 barras.

6.1 TRABALHOS FUTUROS

Após os testes realizados verificou-se que a estratégia de implementação adotada, mesmo sendo bem inserida para alguns casos, não foi eficiente para outros, visto a grande sobrecarga de procedimentos recursivos em algumas estratégias utilizando-se estruturas dinâmicas, e o método de implementação adotado.

Assim, para a otimização da proposta implementada certas estratégias de melhoria podem vir a trazer melhores resultados. A seguir encontram-se as estratégias propostas para se continuar o trabalho:

Redução de Vizinhança

A proposta implementada é rápida e eficaz em sistemas de pequeno e médio porte, mas não reduzem, na mesma porcentagem, as soluções geradas/avaliadas em sistemas de grande porte. Em Baran; Wu, (1989) é utilizado um método, o mesmo adotado por Possagnolo (2015), de redução de vizinhança que trabalha com o estudo de ramos vizinhos ao ramo de chave aberta em um determinado laço interno independente. Avalia-se cada configuração radial mudando o status de chave fechada de cada ramo vizinho ao ramo originalmente com status de chave aberta, e se um ramo produz uma configuração radial com perdas maiores, não se continua a percorrer pelos seus futuros vizinhos, e parte para os ramos vizinhos do outro lado do ramo que originalmente tinha status de chave aberta. A proposta mostrou-se pertinente e, portanto, pode ser implementada neste trabalho.

Diversificação e Intensificação

A diversificação mostrou-se válida porque fez com que se reduzisse ainda mais o espaço de busca de soluções. Ajustes quanto à quantidade de elementos proibidos, bem como o tempo de proibição podem ser estudados de forma mais aprofundada.

A intensificação pode ser revista e estudada de outra maneira. Pode se utilizar de uma memória de longo prazo que armazena somente as melhores configurações de cada iteração da BT. Assim, logo após o procedimento de intensificação, no momento em que uma solução corrente for idêntica a uma das soluções armazenadas nesta memória de longo prazo, o processo termina, ou parte a partir de outra solução de baixa qualidade. Isto faz com que, após o procedimento de intensificação não seja necessário percorrer os mesmos espaços de busca, que refletem em uma mesma solução final.

Com isso podem ser armazenadas diversas soluções de baixa qualidade para que o procedimento de intensificação possa ser reiniciado várias vezes, e com a memória de longo prazo sabe-se que, se algum desses procedimentos de intensificação encontrar alguma solução corrente já visitada, o processo termina. Tanto a memória de longo prazo quanto a estrutura de armazenamento das soluções de baixa qualidade serão pequenas e lineares, pois a memória de longo prazo armazenaria somente as melhores soluções de cada iteração da busca local, e

poderiam ser armazenadas, por exemplo, somente dez soluções de baixa qualidade para iniciar o procedimento de intensificação.

Frequência e Recência

Outra técnica que pode estar presente no procedimento de busca local realizado pela BT é a frequência e recência com que elementos participam de configurações de alta ou baixa qualidade.

Conforme um ramo participe mais de soluções de boa qualidade mantendo seu status de chave durante diversas configurações ou iterações realizadas, esse ramo deve manter o estado de sua de chave durante um período de tempo, ou de iterações. Por outro lado, um ramo que persista com seu status de chave em diversas configurações ou iterações realizadas de forma que produza soluções de baixa qualidade, este deve ser forçado a sempre modificar seu status para diferentes configurações.

Esta estratégia indica os melhores comportamentos de elementos de uma solução para determinadas configurações ou espaços de busca. Essa técnica foi utilizada por Marinho, (2017) apresentou resultados com tempos de execução pequenos mostrando o bom rendimento computacional desta técnica.

Restruturação de código

Esta parte é a mais complexa a ser realizada. Trata-se de organizar um estudo no código, a fim de se retirar, inicialmente, diversas características recursivas, tornando-as iterativas, buscando um aumento no rendimento do algoritmo, ou seja, uma redução nos tempos de execução.

A princípio, a proposta deverá ser realizada nos conjuntos de funções que atuam nos procedimentos de construção e destruição de uma árvore dinâmica, bem como no procedimento de cálculo de fluxo de potência de uma configuração radial em árvore dinâmica. Posteriormente, outras funções podem ser aperfeiçoadas por meio desta proposta.

Além do procedimento de reestruturação de código das funções recursivas, um estudo com diferentes técnicas de aleatoriedade pode ser realizado, visto que o rendimento do GRASP e do PR foram baixos devido ao comportamento aleatório utilizado.

PR

Mesmo a técnica mostrando ter um comportamento estável para certa quantidade de soluções avaliadas, seu tempo de execução foi alto. Estudos quanto ao uso de diferentes técnicas de escolha de elementos aleatórios podem ser avaliados para então ser verificado o rendimento da técnica de forma “pura”. Uma vez com a ausência da suspensão de atividade do código pode-se avaliar o rendimento desta técnica mais de perto.

Paralelização

O código foi construído seguindo o paradigma estruturado na linguagem C. A forma como foi projetada a implementação possui a capacidade de uma fácil inserção de uma metodologia de paralelização de código.

Após a reestruturação de código, removendo grande parte da recursividade um próximo, e importante passo, pode ser a paralelização de código. Isso traz ainda mais rendimento para o algoritmo, reduzindo de forma exponencial os tempos de execução encontrados.

REFERÊNCIAS

- ABDELAZIZ, A. Y.; MOHAMED, F. M.; MEKHAMER, S. F.; BADR, M. A. L. Distribution system reconfiguration using a modified Tabu Search algorithm. **Electric Power Systems Research**, Lausanne, v. 80, n. 8, p. 943–953, 2010.
- ABDELAZIZ, M. Distribution network reconfiguration using a genetic algorithm with varying population size. **Electric Power Systems Research**, Lausanne, v. 142, p. 9–11, 2017.
- AFSARI, M.; SINGH, S. P.; RAJU, G. S.; RAO, G. K.; SINGH, S. P.; RAJU, G. S.; FAST, G. K. R. A.; FLOW, P. A Fast Power Flow Solution of Radial Distribution Networks. **Electric Power Components and Systems**, Philadelphia, v. 30, p. 1065–1074, 2010.
- AIEX, R. M.; BINATO, S.; RESENDE, M. G. C. Parallel GRASP with path-relinking for job shop scheduling. **Parallel Computing**, Amsterdam, v. 29, n. 4 SPEC., p. 393–430, 2003.
- ALVAREZ-VALDES, R.; CRESPO, E.; TAMARIT, J. M.; VILLA, F. GRASP and path relinking for project scheduling under partially renewable resources. **European Journal of Operational Research**, Amsterdam, v. 189, n. 3, p. 1153–1170, 2008.
- AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA - ANEEL. Site. Disponível em: <http://www2.aneel.gov.br/area.cfm?idArea=801&idPerfil=4>. Acesso em: 27 jan. 2020.
- ARIN, A.; RABADI, G. Local search versus Path Relinking in metaheuristics: Redesigning Meta-RaPS with application to the multidimensional knapsack problem. **Applied Soft Computing Journal**, [s. l.], v. 46, p. 317–327, 2016.
- ARMENTANO, V. A.; SHIGUEMOTO, A. L.; LØKKETANGEN, A. Tabu search with path relinking for an integrated productiondistribution problem. **Computers and Operations Research**, New York, v. 38, n. 8, p. 1199–1209, 2011.
- AUGUGLIARO, A.; DUSONCHET, L.; FAVUZZA, S.; IPPOLITO, M. G.; SANSEVERINO, E. R. A backward sweep method for power flow solution in distribution networks. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 32, n. 4, p. 271–280, 2010.
- BARAN, M. E.; WU, F. F. Network reconfiguration in distribution systems for loss reduction and load balancing. **IEEE Transactions on Power Delivery**, [s. l.], v. 4, n. 2, p. 1401–1407, 1989.
- BARBALHO, H.; ROSSETI, I.; MARTINS, S. L.; PLASTINO, A. A hybrid data mining GRASP with path-relinking. **Computers and Operations Research**, New York, v. 40, n. 12, p. 3159–3173, 2013.
- BAYAT, A. Uniform voltage distribution based constructive algorithm for optimal reconfiguration of electric distribution networks. **Electric Power Systems Research**, Lausanne, v. 104, p. 146–155, 2013.

BAYAT, A.; BAGHERI, A.; NOROOZIAN, R. Optimal siting and sizing of distributed generation accompanied by reconfiguration of distribution networks for maximum loss reduction by using a new UVDA-based heuristic method. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 77, p. 360–371, 2016.

BENAVIDES, A. J.; RITT, M.; BURIOL, L. S.; FRANÇA, P. M. An iterated sample construction with path relinking method: Application to switch allocation in electrical distribution networks. **Computers and Operations Research**, New York, v. 40, n. 1, p. 24–32, 2013.

BOUDIA, M.; LOULY, M. A. O.; PRINS, C. A reactive GRASP and path relinking for a combined production-distribution problem. **Computers and Operations Research**, New York, v. 34, n. 11, p. 3402–3419, 2007.

CHAOVALITWONGSE, W. A.; OLIVEIRA, C. A. S.; CHIARINI, B.; PARDALOS, P. M.; RESENDE, M. G. C. Revised GRASP with path-relinking for the linear ordering problem. **Journal of Combinatorial Optimization**, [s. l.], v. 22, n. 4, p. 572–593, 2011.

CHENG, C. S.; SHIRMOHAMMADI, D. A three-phase power flow method for real-time distribution system analysis. **IEEE Transactions on Power Systems**, [s. l.], v. 10, n. 2, p. 671–679, 1995.

CHIOU, J. P.; CHANG, C. F.; SU, C. T. Variable scaling hybrid differential evolution for solving network reconfiguration of distribution systems. **IEEE Transactions on Power Systems**, [s. l.], v. 20, n. 2, p. 668–674, 2005.

CIVANLAR, S.; GRAINGER, J. J.; YIN, H.; LEE, S. S. H. Distribution Feeder Reconfiguration for Loss Reduction. **IEEE Transactions on Power Delivery**, [s. l.], v. 3, n. 3, p. 1217–1223, 1988.

CORMEN, T.; LEISERSON, C.; RIVEST, R.; STEIN, C. **Introduction to algorithms**. [s.l.: s.n.]. v. 7

DE OLIVEIRA, E. J.; ROSSETI, G. J.; DE OLIVEIRA, L. W.; GOMES, F. V.; PERES, W. New algorithm for reconfiguration and operating procedures in electric distribution systems. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 57, p. 129–134, 2014a.

DE OLIVEIRA, L. W.; DE OLIVEIRA, E. J.; GOMES, F. V.; SILVA, I. C.; MARCATO, A. L. M.; RESENDE, P. V. C. Artificial Immune Systems applied to the reconfiguration of electrical power distribution networks for energy loss minimization. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 56, p. 64–74, 2014b.

DÍAZ, J. A.; FERNÁNDEZ, E. Hybrid scatter search and path relinking for the capacitated p-median problem. **European Journal of Operational Research**, [s. l.], v. 169, n. 2, p. 570–585, 2006.

DROZDEK, A. **Data Structures and Algorithms in C ++ Fourth Edition**. [s.l.] : CENGAGE Learning, 2012.

DUARTE, A.; MARTÍ, R. Tabu search and GRASP for the maximum diversity problem. **European Journal of Operational Research**, [s. l.], v. 178, n. 1, p. 71–84, 2007.

DUARTE, A.; MARTÍ, R.; GORTAZAR, F. Path relinking for large-scale global optimization. **Soft Computing**, [s. l.], v. 15, n. 11, p. 2257–2273, 2011.

DUKPA, A.; VENKATESH, B.; EL-HAWARY, M. Application of continuation power flow method in radial distribution systems. **Electric Power Systems Research**, Lausanne, v. 79, p. 1503–1510, 2009.

FARAG, H. E.; SHATSHAT, R. El; ZIDAN, A. A generalized power flow analysis for distribution systems with high penetration of distributed generation. **Electric Power Systems Research**, Lausanne, v. 81, n. 7, p. 1499–1506, 2011.

FEO, T. A.; RESENDE, M. G. C. **A probabilistic heuristic for a computationally difficult set covering problem**. [S. l.: s. n.], 1989.

FERREIRA, F. A. L. **Metodologia para reconfiguração de redes de distribuição trifásicas assimétricas e não balanceadas com geração distribuída**. [S. l.], 2010. p. 145,

FRANCO, J.; LAVORATO, M.; RIDER, M. J.; ROMERO, R. An efficient implementation of tabu search in feeder reconfiguration of distribution systems. **IEEE Power and Energy Society General Meeting**, [s. l.], p. 1–8, 2012a.

FRANCO, J.; MEMBER, S.; LAVORATO, M.; MEMBER, S.; RIDER, M. J. An Efficient Implementation of Tabu Search in Feeder Reconfiguration of Distribution Systems. [s. l.], p. 1–8, 2012b.

GHATAK, U.; MUKHERJEE, V. A fast and efficient load flow technique for unbalanced distribution system. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 84, p. 99–110, 2017.

GLOVER, F. Tabu search and adaptive memory programming - advances, applications and challenges. **Interfaces in Computer Science and Operations Research**, [s. l.], p. 1–75, 1996.

GLOVER, F.; LAGUNA, M. Tabu search. **Handbook of Combinatorial Optimization**, [s. l.], v. 5–5, n. July, p. 3261–3362, 2013.

GONZÁLEZ, M. A.; ODDI, A.; RASCONI, R.; VARELA, R. Scatter search with path relinking for the job shop with time lags and setup times. **Computers and Operations Research**, New York, v. 60, p. 37–54, 2015.

GONZÁLEZ, M. A.; VELA, C. R.; VARELA, R. Scatter search with path relinking for the flexible job shop scheduling problem. **European Journal of Operational Research**, New York, v. 245, n. 1, p. 35–45, 2015.

GUERRA, W.; GALVIS, J. C.; VINICIUS, M. Reconfiguração de Sistemas de Distribuição Utilizando Otimização por Nuvem de Partículas. **IV Simpósio Brasileiro de Sistemas Elétricos - SBSE**, [s. l.], p. 1–6, 2012.

HAMOUDA, A.; ZEHAR, K. Improved algorithm for radial distribution networks load flow solution. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 33, n. 3, p. 508–514, 2011.

I. J. RAMIREZ-ROSADO, J. L. B.-A. Genetic Algorithms Applied to the Design of Large Power Distribution Systems. **IEEE Transactions on Power Systems**, [s. l.], v. 13, n. 2, p. 696–703, 1998.

ISABELLA F. FEITEIRA; LEONARDO H. MACEDO; RUBÉN ROMERO. Meta-Heurística Grasp Especializada Para a Reconfiguração De Sistemas De Distribuição De Energia Elétrica. **Proceedings XXII Congresso Brasileiro de Automática**, [s. l.], 2018.

JIA, S.; HU, Z.-H. Path-relinking Tabu search for the multi-objective flexible job shop scheduling problem. **Computers & Operations Research**, [s. l.], v. 47, p. 11–26, 2014.

KARAMI, A.; MOHAMMADI, M. S. Radial basis function neural network for power system load-flow. **Electrical Power & Energy Systems**, [s. l.], v. 30, p. 60–66, 2008.

KUMAR, D.; SINGH, A.; MISHRA, S. K.; JHA, R. C.; SAMANTARAY, S. R. A coordinated planning framework of electric power distribution system: Intelligent reconfiguration. **International Transactions on Electrical Energy Systems**, [s. l.], n. November 2017, p. 1–20, 2018.

LAGUNA, M.; MARTI, R.; CAMPOS, V. Intensification and diversification with elite tabu search solutions for the linear ordering problem. **Computers and Operations Research**, New York, v. 26, n. 12, p. 1217–1230, 1999.

LI, F.; BROADWATER, R. P. Distributed algorithms with theoretic scalability analysis of radial and looped load flows for power distribution systems. **Electric Power Systems Research**, Lausanne, v. 65, p. 169–177, 2003.

LI, H.; JIN, Y.; ZHANG, A.; SHEN, X.; LI, C.; KONG, B. An improved hybrid load flow calculation algorithm for weakly-meshed power distribution system. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 74, p. 437–445, 2016a.

LI, H.; MAO, W.; ZHANG, A.; LI, C. An improved distribution network reconfiguration method based on minimum spanning tree algorithm and heuristic rules. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 82, p. 466–473, 2016b.

LISBOA, A. C.; GUEDES, L. S. M.; VIEIRA, D. A. G.; SALDANHA, R. R. A fast power flow method for radial networks with linear storage and no matrix inversions. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 63, p. 901–907, 2014.

MANTOVANI, J. R. S.; CASARI, F.; ROMERO, R. A. Reconfiguração de sistemas de distribuição radiais utilizando o critério de queda de tensão. **Revista Brasileira de Controle & Automação - SBA**, [s. l.], v. 11, n. 03, p. 150–159, 2000.

MARINHO, R. I. O. P. **Reconfiguração de sistemas de distribuição de energia elétrica utilizando metodologias multipartida e busca tabu**. [S. l.], 2017.

MARTÍ, R.; SANDOYA, F. GRASP and path relinking for the equitable dispersion problem. **Computers and Operations Research**, New York, v. 40, n. 12, p. 3091–3099, 2013.

MASTERI, K.; VENKATESH, B. Real-time smart distribution system reconfiguration using complementarity. **Electric Power Systems Research**, Lausanne, v. 134, p. 97–104, 2016.

MENDES, A.; BOLAND, N.; GUINEY, P.; RIVEROS, C. Switch and tap-changer reconfiguration of distribution networks using evolutionary algorithms. **IEEE Transactions on Power Systems**, [s. l.], v. 28, n. 1, p. 85–92, 2013.

MESTRIA, M.; SATORU OCHI, L.; DE LIMA MARTINS, S. GRASP with path relinking for the symmetric Euclidean clustered traveling salesman problem. **Computers and Operations Research**, New York, v. 40, n. 12, p. 3218–3229, 2013.

MOK, S.; ELANGO VAN, S.; LONGJIAN, C.; SALAMA, M. A New Approach for Power Flow Analysis of Balanced Radial Distribution Systems. **Electric Machines & Power Systems**, [s. l.], v. 28, p. 325–340, 2010.

NASCIMENTO, M. C. V.; RESENDE, M. G. C.; TOLEDO, F. M. B. GRASP heuristic with path-relinking for the multi-plant capacitated lot sizing problem. **European Journal of Operational Research**, [s. l.], v. 200, n. 3, p. 747–754, 2010.

NGUYEN, T. T.; TRUONG, A. V. Distribution network reconfiguration for power loss minimization and voltage profile improvement using cuckoo search algorithm. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 68, p. 233–242, 2015.

NGUYEN, V. P.; PRINS, C.; PRODHON, C. Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking. **European Journal of Operational Research**, [s. l.], v. 216, n. 1, p. 113–126, 2012.

PENG, B.; LÜ, Z.; CHENG, T. C. E. A tabu search/path relinking algorithm to solve the job shop scheduling problem. **Computers and Operations Research**, New York, v. 53, p. 154–164, 2015.

PIÑANA, E.; PLANA, I.; CAMPOS, V.; MARTÍ, R. GRASP and path relinking for the matrix bandwidth minimization. **European Journal of Operational Research**, [s. l.], v. 153, n. 1, p. 200–210, 2004.

POSSAGNOLO, L. H. F. M. **Reconfiguração de Sistemas de distribuição operando em vários níveis de demanda através de uma meta-heurística de busca em vizinhança variável**. [S. l.: s. n.], 2015. 178 p.

RENATO CESPEDES G. New Method for the Analysis of Distribution Networks. **IEEE Transactions on Power Delivery**, [s. l.], v. 5, n. 1, p. 391–396, 1990.

RESENDE, M. G. C.; MARTÍ, R.; GALLEGO, M.; DUARTE, A. GRASP and path relinking for the max-min diversity problem. **Computers and Operations Research**, New York, v. 37, n. 3, p. 498–508, 2010.

ROBERTS, E. S. **The art and science of C: an introduction to computer science.** [S. l.]: Addison-Wesley, 1995.

RODRIGUEZ, F. J.; GLOVER, F.; GARCÍA-MARTÍNEZ, C.; MARTÍ, R.; LOZANO, M. GRASP with exterior path-relinking and restricted local search for the multidimensional two-way number partitioning problem. **Computers and Operations Research**, New York, v. 78, n. August 2016, p. 243–254, 2017.

SHAKARAMI, M. R.; BEIRANVAND, H.; BEIRANVAND, A.; SHARIFIPOUR, E. A recursive power flow method for radial distribution networks: Analysis, solvability and convergence. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 86, p. 71–80, 2017.

SHIRMOHAMMADI, D. A compensation-based power flow method for weakly meshed distribution and transmission networks - Power Systems, IEEE Transactions on. **IEEE Transactions on Power Systems**, [s. l.], v. 3, n. 2, p. 753–762, 1988.

SHIRMOHAMMADI, D.; HONG, H. W. Reconfiguration of electric distribution networks for resistive line losses reduction. **IEEE Transactions on Power Delivery**, [s. l.], v. 4, n. 2, p. 1492–1498, 1989.

SHOUKOUHI, M.; SHOJAEIAN, S. Distribution Network System Reconfiguration for Power Loss Minimization and Voltage Profile Enhancement using Ant Lion Algorithm. In: NATIONAL CONFERENCE OF ENERGY 2015, Islamic Azad University, Iran. **Anais [...]** Iran: Islamic Azad University, 2015.

SILVÉRIO, H. L.; RODRIGUES, L. C. de A.; STEINER, M. T. A. **Meta-Heurísticas em Pesquisa Operacional.** [S. l: s. n.], 2013.

SINGH, S.; GHOSE, T. Improved radial load flow method. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 44, n. 1, p. 721–727, 2013.

SWARNKAR, A.; GUPTA, N.; NIAZI, K. R. A novel codification for meta-heuristic techniques used in distribution network reconfiguration. **Electric Power Systems Research**, Lausanne, v. 81, n. 7, p. 1619–1626, 2011.

TOMOIAGĂ, B.; CHINDRIȘ, M.; SUMPER, A.; VILLAFILA-ROBLES, R.; SUDRIA-ANDREU, A. Distribution system reconfiguration using genetic algorithm based on connected graphs. **Electric Power Systems Research**, Lausanne, v. 104, p. 216–225, 2013.

TORRES, J.; GUARDADO, J. L.; RIVAS-DÁVALOS, F.; MAXIMOV, S.; MELGOZA, E. A genetic algorithm based on the edge window decoder technique to optimize power distribution systems reconfiguration. **International Journal of Electrical Power and Energy Systems**, [s. l.], v. 45, n. 1, p. 28–34, 2013.

VILLEGAS, J. G.; PRINS, C.; PRODHON, C.; MEDAGLIA, A. L.; VELASCO, N. A GRASP with evolutionary path relinking for the truck and trailer routing problem. **Computers and Operations Research**, New York, v. 38, n. 9, p. 1319–1334, 2011.

WANG, Y.; LÜ, Z.; GLOVER, F.; HAO, J. K. Path relinking for unconstrained binary quadratic programming. **European Journal of Operational Research**, [s. l.], v. 223, n. 3, p. 595–604, 2012.

WANG, Y.; LÜ, Z.; GLOVER, F.; HAO, J. K. Probabilistic GRASP-Tabu Search algorithms for the UBQP problem. **Computers and Operations Research**, New York, v. 40, n. 12, p. 3100–3107, 2013.

ZHANG, D.; FU, Z.; ZHANG, L. An improved TS algorithm for loss-minimum reconfiguration in large-scale distribution systems. **Electric Power Systems Research**, Lausanne, v. 77, n. 5–6, p. 685–694, 2007.

ANEXO A – PROBLEMAS DE ALTA COMPLEXIDADE

Este anexo traz definições resumidas de cada problema citado no Capítulo 2, pois alguns desses problemas são utilizados no decorrer do mesmo capítulo para exemplificar a aplicação de certos métodos meta-heurísticos propostos.

- **Ordenamento Linear:** a partir de uma matriz quadrada deseja-se maximizar a soma dos pesos acima da diagonal principal. Uma metodologia utilizada é a troca de elementos de determinados intervalos de uma coluna M_{ij} por M_{ji} maximizando-se o setor acima da diagonal principal (LAGUNA; MARTI; CAMPOS, 1999). Chaovaitwongse *et al.* 2011 utiliza o procedimento de trocas linha/coluna pelo movimento 2-opt;
- **Sequenciamento de tarefas:** se possui certa quantidade de tarefas e certa quantidade de máquinas para se executar. As tarefas e máquinas podem apresentar tempos de execução diferentes. Três objetivos podem ser avaliados, a minimização de execução de todas as tarefas, a minimização da carga total das máquinas e a minimização da carga total de uma máquina. Tratamentos multi-objetivos podem ser encontrados em Jia and Hu 2014 e Peng *et al.* 2015, e análises com um objetivo em Aiex *et al.* 2003 e González *et al.* 2015b, 2015a;
- **Minimização de Largura de Banda:** para uma matriz quadrada seus elementos não-nulos devem se encontrar o mais próximo possível da diagonal principal. A escolha dos elementos para serem trocados em um procedimento de busca local pode ser realizado pela atratividade da troca realizada (PIÑANA *et al.*, 2004);
- **P-Mediana:** a partir de uma distribuição de clientes m , escolhe-se uma quantidade de medianas p , as quais devem perfazer conjuntos de ligação com os clientes, onde nenhum cliente está conectado a mais de uma mediana, de modo a minimizar a soma das distâncias de cada cliente até a mediana mais próxima. Este problema pode ser utilizado para cálculos de pontos de ônibus em uma cidade, onde o ponto de ônibus deve ter a melhor posição para cada bairro/localidade onde foi implantado;

- **Diversidade Máxima e Mínima:** a partir de um conjunto de soluções se é escolhido um subconjunto que possua a máxima diversidade entre as soluções. Isto faz com que a busca local para cada solução seja executada em diferentes espaços de busca de soluções;
- **Produção e Distribuição:** a partir da presença de um ou mais galpões com capacidade de armazenamento e produções específicas, clientes com capacidade de demanda, veículo(s) com capacidade de armazenamento, e rotas de entrega de produtos, deseja-se minimizar o custo total das rotas. Problema clássico de logística de entrega de mercadorias;
- **Alocação de recursos escassos:** no momento de produção de certos materiais primários alguns produtos secundários são muito ou pouco utilizados tornando-se escassos, ou seja, novas remessas deste tipo de produto secundário devem ser requisitadas para que a produção do produto primário continue. Este tipo de problema ordena a alocação de recursos escassos para um conjunto de atividade em um projeto por um tempo determinado;
- **Dimensionamento de lote:** o objetivo é se encontrar um plano de produção que minimize a produção, configuração, custos de inventário, e encontra sem demora as demandas de itens nos períodos do horizonte de planejamento;
- **Otimização em Larga Escala:** problemas clássicos de otimização de uma função;
- **Truck & trailer:** a frota de entregas de produtos é formada por veículos heterogêneos, composta por carros/furgões e trailers. As rotas possuem clientes acessados por somente uma configuração dessa frota, um tipo de automóvel, ou pelo veículo completo (automóvel adicionado ao trailer). O objetivo é se minimizar o custo das rotas;
- **Programação Quadrática Binária:** problema formado por uma matriz quadrada de tamanho n e um vetor x de tamanho n unidimensional e binário. O objetivo é se maximizar os somatórios das multiplicações entre o elemento ij da matriz pelos elementos i e j do vetor. A matriz é construída com constantes e é imutável, restando modificar-se os elementos do vetor binário x ;
- **Localização de rota com 2 escalões:** problema de transporte de mercadorias entre instalações, misturando o problema de rota e distribuição com o problema de truck & trailer. Consiste de depósitos localizados próximos à grandes

idades, considerando grande economia quanto a entregas realizadas diretamente de um depósito principal. Outro porém é de que os municípios querem reduzir ainda mais o movimento no centro das cidades se criando plataformas logisticamente distribuídas nas periferias das quais pequenos veículos podem realizar suas entregas no centro, evitando o trânsito de veículos de grande porte;

- **Projeto de redes com 2 caminhos:** a partir de um grafo conectado com arestas não direcionadas o problema consiste em se encontrar um subconjunto de arestas com peso mínimo onde sempre se tenham 2 caminhos se conectando cada par de vértices origem-destino;
- **Caixeiro Viajante:** consiste em se percorrer determinado conjunto de cidades, retornando à cidade inicial, após o término do percurso, onde a distância percorrida seja a mínima e só se possa visitar cada cidade uma única vez;
- **Problema da Mochila:** uma mochila possui uma determinada capacidade e existem itens com certas capacidades e valores diferentes. O objetivo é se maximizar o valor dos itens contidos na mochila. O problema pode abordar diversas mochilas (ARIN; RABADI, 2016);
- **Particionamento de números:** um problema que possui variantes. Uma delas se consiste num determinado conjunto de números e pretende-se encontrar uma sequência de sinais +/- associados a um conjunto de números, tal que seja mínimo o módulo do resultado do conjunto de números. A outra variante consiste em um conjunto de vetores de coordenadas com o objetivo de se particionar em dois conjuntos disjuntos onde a máxima diferença entre as somas por coordenada dos elementos em cada grupo seja minimizada (RODRIGUEZ *et al.*, 2017).

ANEXO B – FLUXO DE POTÊNCIA EM SISTEMAS RADIAIS

Este anexo tem o objetivo de trazer o algoritmo mais detalhado e as características do algoritmo de cálculo do fluxo de potência de uma solução radial por meio da apresentação das expressões matemáticas necessárias a cada estágio do procedimento.

O procedimento inicia adotando valores para as tensões nas barras do sistema que é tipicamente adotado como a mesma tensão da subestação, assim, para cada barra k assume-se $V_k = V_{ref} + j0$ onde V_{ref} é o valor em módulo de tensão da subestação repassado pelos arquivos do sistema. Uma vez iniciadas as tensões nas barras é possível se conhecer a corrente de carga em todas as barras e as correntes em todos os ramos do sistema.

O procedimento de cálculo das correntes é realizado se partindo das barras terminais em direção à subestação (procedimento *backward*). Uma vez atingida a subestação são calculadas as perdas elétricas ativas e reativas da solução, por um valor aproximado.

Terminado o procedimento *backward* conhece-se a corrente que está saindo da subestação. Neste momento, por meio dos valores das correntes encontradas em todos os ramos de ligação parte-se da subestação às barras terminais se recalculando as tensões em todas as barras (procedimento *forward*). Uma vez alcançadas as barras terminais se repete o procedimento *backward* e as novas perdas são encontradas. O procedimento de parada é realizado por meio da variação entre duas iterações consecutivas ΔP_{at} quando se é menor do que um erro limite adotado ϵ , $\Delta P_{at} < \epsilon$.

B.1 CÁLCULO DA CORRENTE DE POTÊNCIA

Na Figura 59 ilustra-se duas barras de um sistema com distribuição radial. A carga da barra k é representada na forma $S_k = P_k + jQ_k$ e a tensão de barra na forma $V_k = V_{kr} + jV_{ki}$ separando as componentes real (r) e imaginária (i). O objetivo é se encontrar a partir destas informações as relações matemáticas a corrente de carga $I_k = I_{kr} + jI_{ki}$.

Inicialmente, têm-se:

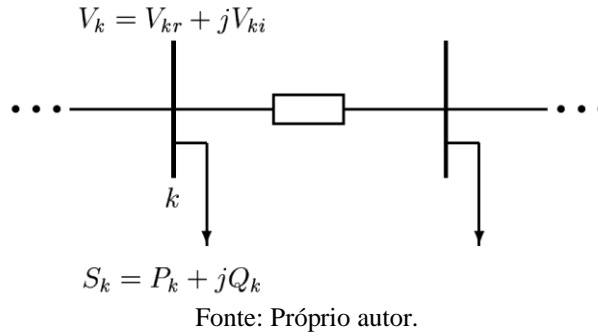
$$S_k = P_k + jQ_k \qquad V_k = V_{kr} + jV_{ki}. \qquad (23)$$

Por definição:

$$S_k = V_k I_k^* \rightarrow I_k^* = \frac{P_k + jQ_k}{V_{kr} + jV_{ki}} = \frac{P_k + jQ_k}{V_{kr} + jV_{ki}} \cdot \frac{(V_{kr} - jV_{ki})}{(V_{kr} - jV_{ki})} \quad (24)$$

$$I_k = \frac{(P_k + jQ_k)(V_{kr} - jV_{ki})}{(V_{kr}^2 + jV_{ki}^2)} = \frac{(P_k V_{kr} + Q_k V_{ki}) - j(P_k V_{ki} - Q_k V_{kr})}{(V_{kr}^2 + jV_{ki}^2)} \quad (25)$$

Figura 59 - Cálculo de corrente de potência.



Adotando que $I_k = I_{kr} + jI_{ki}$ e igualando com a relação obtida em (25) se encontram as relações para a corrente de carga separando as partes real e imaginária:

$$I_{kr} = \frac{(P_k V_{kr} + Q_k V_{ki})}{(V_{kr}^2 + jV_{ki}^2)} \quad (26)$$

$$I_{ki} = \frac{(P_k V_{ki} - Q_k V_{kr})}{(V_{kr}^2 + jV_{ki}^2)} \quad (27)$$

B.2 CÁLCULO DE TENSÃO DA BARRA

Na Figura 60 apresentam-se duas barras consecutivas de um sistema de distribuição radial. Pelo procedimento inicial já se conhecem as tensões em cada barra k , $V_k = V_{kr} + jV_{ki}$, a corrente para cada ramo km , $I_{km} = I_{kmr} + jI_{kmi}$ e a impedância do ramo $Z_{km} = R_{km} + jX_{km}$. Pretende-se, então, encontrar a tensão na barra m , $V_m = V_{mr} + jV_{mi}$.

Sabe-se a seguinte informação:

$$I_{km} = I_{kmr} + jI_{kmi} \quad (28)$$

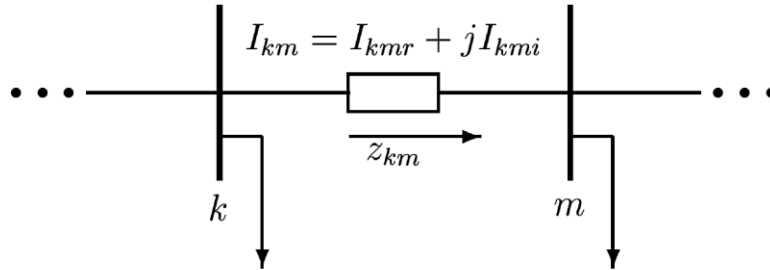
Por definição, têm-se:

$$V_k = V_{kr} + jV_{ki} = V_m + (R_{km} + jX_{km})(I_{kmr} + jI_{kmi}) \quad (29)$$

$$V_{kr} + jV_{ki} = V_{mr} + jV_{mi} + (R_{km}I_{kmr} - X_{km}I_{kmi}) + j(X_{km}I_{kmr} - R_{km}I_{kmi}) \quad (30)$$

Figura 60 - Cálculo de tensão na barra.

$$V_k = V_{kr} + jV_{ki} \quad V_m = V_{mr} + jV_{mi}$$



Fonte: Próprio autor.

Pela relação obtida em (30), as componentes real e imaginária para a tensão elétrica na barra m podem ser separadas e representadas como:

$$V_{mr} = V_{kr} - R_{km}I_{kmr} + X_{km}I_{kmi} \quad (31)$$

$$V_{mi} = V_{ki} - X_{km}I_{kmi} + R_{km}I_{kmr} \quad (32)$$

Uma vez conhecidos os valores da tensão V_k e da corrente I_{km} é possível se determinar o valor de V_m por meio de (31) e (32).

B.3 CÁLCULO DAS PERDAS ELÉTRICAS NO SISTEMA DE DISTRIBUIÇÃO

As perdas elétricas, tanto ativas quanto reativas, podem ser obtidas por deduções de relações simples. As perdas para um ramo com as mesmas características do ramo da Figura 59 possui a seguinte representação:

$$\begin{aligned} S_{km} &= P_{kmp} + jQ_{kmp} = \Delta V_{km} I_{km}^* = (R_{km} + jX_{km}) I_{km} I_{km}^* \\ &= (R_{km} + jX_{km}) I_{km}^2 \end{aligned} \quad (33)$$

Pela relação (33), as perdas ativas e reativas no ramo km são obtidas da seguinte maneira:

$$P_{kmp} = R_{km} I_{km}^2 \quad (34)$$

$$Q_{kmp} = X_{km} I_{km}^2 \quad (35)$$

Como as perdas são calculadas para todos os ramos de ligação e o resultado final é um somatório destes valores, as expressões (34) e (35) podem ser expressadas de forma mais generalizada da seguinte forma:

$$P_t = \sum_{(k,m) \in \Omega} R_{km} I_{km}^2 \quad (36)$$

$$Q_t = \sum_{(k,m) \in \Omega} Q_{km} I_{km}^2 \quad (37)$$

onde Ω representa todo o conjunto de ramos do sistema de distribuição elétrica.

B.4 ALGORITMO DO PROCEDIMENTO DE FLUXO DE POTÊNCIA

1. Iniciar o problema se adotando o valor de tensão nas barras do sistema o mesmo da barra configurada como subestação, $V_k = V_{ref} + j0$, onde k significa o índice de cada barra do sistema e V_{ref} o valor em módulo da tensão na subestação. Iniciar perdas elétricas $P_{per1} = 0$, e escolher o limite de erro (tolerância) ϵ ;
2. Partir das barras terminais a caminho da subestação se calculando a corrente de carga de todas as barras por meio das relações (26) e (27) assim como as correntes I_{km} para todos os ramos de ligação por meio de somatório (procedimento *backward*);
3. Calcular as perdas elétricas ativas do sistema por meio de (36) com as correntes I_{km} calculadas nos ramos do passo anterior. Atualizar as perdas $P_{per2} = P_t$;
4. Se $\Delta P_{per} = |P_{per2} - P_{per1}| \leq \epsilon$ o processo para pois atingiu a convergência. Caso contrário, $P_{per2} = P_{per1}$, e siga para o passo “5”;
5. Conhecidos os valores de corrente I_{km} nos ramos de ligação, parte-se da subestação a caminho das barras terminais se recalculando e atualizando os valores de tensão de cada barra do sistema (procedimento *forward*). Voltar ao passo “2”.

ANEXO C – DADOS DOS PROBLEMAS TESTADOS

Este apêndice anexo tem o intuito de apresentar os dados dos arquivos pertinentes aos sistemas testados e avaliados de 14, 33, 84, 136 e 417 barras. Para todos os sistemas foi inserida a informação adicional da quantidade total de ramos de ligação do sistema de distribuição elétrica avaliado.

C.1 SISTEMA DE 14 BARRAS

As informações iniciais do problema se encontram descritas a seguir, e os dados pertinentes aos ramos de ligação (par de barras a qual fazem parte e impedância) bem como as informações de cada barra (potência ativa, reativa indutiva e reativa capacitiva) se encontram descritos na Tabela 24.

- Quantidade de barras: 14;
- Quantidade total de ramos de ligação: 16;
- Barra de referência da subestação: 14;
- Tensão na barra de referência em p.u.: 1,0;
- Tensão base da barra de referência: 23,0 kV;
- Potência base: 100.000 kVA;
- Máximo erro entre perdas elétricas: 10^{-8} .

Tabela 24 - Dados do Sistema de 14 barras.

Dados de Barras				Dados de Ramos			
Índice	Pot. Ativa (kW)	Pot. Reat. Indutiva (kVAr)	Pot. Reat. Capacitiva (kVAr)	Barra 1	Barra 2	Resistência (%)	Reatância (%)
1	2100	1000	1800	14	13	7,5	10
2	1000	900	0	13	12	8	11
3	1000	700	1800	13	11	9	18
4	1000	900	0	11	10	4	4
5	4500	2000	3700	14	9	11	11
6	600	100	600	9	8	8	11
7	1000	900	0	9	7	11	11
8	5000	3000	1200	8	6	11	11
9	4000	2700	0	8	5	8	11

10	1500	1200	0	14	4	11	11
11	2000	800	1200	4	3	9	12
12	3000	1500	1100	4	2	8	11
13	2000	1600	0	2	1	4	4
14	0	0	0	12	6	4	4
				7	3	4	4
				10	1	9	12

Fonte: (CIVANLAR *et al.*, 1988).

C.2 SISTEMA DE 33 BARRAS

As informações iniciais do problema se encontram descritas a seguir, e os dados pertinentes aos ramos de ligação (par de barras a qual fazem parte e impedância) bem como as informações de cada barra (potência ativa, reativa indutiva e reativa capacitiva) se encontram descritos na Tabela 25.

- Quantidade de barras: 33;
- Quantidade total de ramos de ligação: 37;
- Barra de referência da subestação: 1;
- Tensão na barra de referência em p.u.: 1,0;
- Tensão base da barra de referência: 12,66 kV;
- Potência base: 1.000 kVA;
- Máximo erro entre perdas elétricas: 10^{-8} .

Tabela 25 - Dados do Sistema de 33 Barras.

Índice	Dados de Barras			Dados de Ramos			
	Pot. Ativa (kW)	Pot. Reat. Indutiva (kVAr)	Pot. Reat. Capacitiva (kVAr)	Barra 1	Barra 2	Resistência (Ω)	Reatância (Ω)
1	0	0	0	1	2	0,0922	0,047
2	100	60	0	2	3	0,493	0,2511
3	90	40	0	3	4	0,366	0,1864
4	120	80	0	4	5	0,3811	0,1941
5	60	30	0	5	6	0,819	0,707
6	60	20	0	6	7	0,1872	0,6188
7	200	100	0	7	8	0,7114	0,2351
8	200	100	0	8	9	1,03	0,74
9	60	20	0	9	10	1,044	0,74
10	60	20	0	10	11	0,1966	0,065
11	45	30	0	11	12	0,3744	0,1238
12	60	35	0	12	13	1,468	1,155
13	60	35	0	13	14	0,5416	0,7129

14	120	80	0	14	15	0,591	0,526
15	60	10	0	15	16	0,7463	0,545
16	60	20	0	16	17	1,289	1,721
17	60	20	0	17	18	0,732	0,574
18	90	40	0	2	19	0,164	0,1565
19	90	40	0	19	20	1,5042	1,3554
20	90	40	0	20	21	0,4095	0,4784
21	90	40	0	21	22	0,7089	0,9373
22	90	40	0	3	23	0,4512	0,3083
23	90	50	0	23	24	0,898	0,7091
24	420	200	0	24	25	0,896	0,7011
25	420	200	0	6	26	0,203	0,1034
26	60	25	0	26	27	0,2842	0,1447
27	60	25	0	27	28	1,059	0,9337
28	60	20	0	28	29	0,8042	0,7006
29	120	70	0	29	30	0,5075	0,2585
30	200	600	0	30	31	0,9744	0,963
31	150	70	0	31	32	0,3105	0,3619
32	210	100	0	32	33	0,341	0,5302
33	60	40	0	8	21	2	2
				9	15	2	2
				12	22	2	2
				18	33	0,5	0,5
				25	29	0,5	0,5

Fonte: (BARAN; WU, 1989)

C.3 SISTEMA DE 84 BARRAS

As informações iniciais do problema se encontram descritas a seguir, e os dados pertinentes aos ramos de ligação (par de barras a qual fazem parte e impedância) bem como as informações de cada barra (potência ativa, reativa indutiva e reativa capacitiva) se encontram descritos na Tabela 26.

- Quantidade de barras: 84;
- Quantidade total de ramos de ligação: 96;
- Barra de referência da subestação: 1;
- Tensão na barra de referência em p.u.: 1,0;
- Tensão base da barra de referência: 11,4 kV;
- Potência base: 10.000 kVA;
- Máximo erro entre perdas elétricas: 10^{-5} .

Tabela 26 – Dados do sistema de 84 barras.

Dados de Barras				Dados de Ramos			
Índice	Pot. Ativa (kW)	Pot. Reat. Indutiva (kVAr)	Pot. Reat. Capacitiva (kVAr)	Barra 1	Barra 2	Resistência (Ω)	Reatância (Ω)
1	0	0	0	1	84	0,1944	0,6624
2	100	50	0	1	2	0,2096	0,4304
3	300	200	0	2	3	0,2358	0,4842
4	350	250	0	3	4	0,0917	0,1883
5	220	100	0	4	5	0,2096	0,4304
6	1100	800	0	5	6	0,0393	0,0807
7	400	320	0	6	7	0,0405	0,138
8	300	200	0	7	8	0,1048	0,2152
9	300	230	0	7	9	0,2358	0,4842
10	300	260	0	7	10	0,1048	0,2152
11	0	0	0	11	84	0,0786	0,1614
12	1200	800	0	11	12	0,3406	0,6944
13	800	600	0	12	13	0,0262	0,0538
14	700	500	0	12	14	0,0786	0,1614
15	0	0	0	15	84	0,1134	0,3864
16	300	150	0	15	16	0,0524	0,1076
17	500	350	0	16	17	0,0524	0,1076
18	700	400	0	17	18	0,1572	0,3228
19	1200	1000	0	18	19	0,0393	0,0807
20	300	300	0	19	20	0,1703	0,3497
21	400	350	0	20	21	0,2358	0,4842
22	50	20	0	21	22	0,1572	0,3228
23	50	20	0	21	23	0,1965	0,4035
24	50	10	0	23	24	0,131	0,269
25	50	30	0	25	84	0,0567	0,1932
26	100	60	0	25	26	0,1048	0,2152
27	100	70	0	26	27	0,2489	0,5111
28	1800	1300	0	27	28	0,0486	0,1656
29	200	120	0	28	29	0,131	0,269
30	0	0	0	30	84	0,1965	0,396
31	1800	1600	0	30	31	0,131	0,269
32	200	150	0	31	32	0,131	0,269
33	200	100	0	32	33	0,0262	0,0538
34	800	600	0	33	34	0,1703	0,3497
35	100	60	0	34	35	0,0524	0,1076
36	100	60	0	35	36	0,4978	1,0222
37	20	10	0	36	37	0,0393	0,0807
38	20	10	0	37	38	0,0393	0,0807
39	20	10	0	38	39	0,0786	0,1614
40	20	10	0	39	40	0,2096	0,4304
41	200	160	0	38	41	0,1965	0,4035
42	50	30	0	41	42	0,2096	0,4304
43	0	0	0	43	84	0,0486	0,1656
44	30	20	0	43	44	0,0393	0,0807
45	800	700	0	45	46	0,2358	0,4842
46	200	150	0	44	45	0,131	0,269
47	0	0	0	47	84	0,243	0,828
48	0	0	0	47	48	0,0655	0,1345
49	0	0	0	48	49	0,0655	0,1345
50	200	160	0	49	50	0,0393	0,0807
51	800	600	0	50	51	0,0786	0,1614

52	500	300	0	51	52	0,0393	0,0807
53	500	350	0	52	53	0,0786	0,1614
54	500	300	0	53	54	0,0524	0,1076
55	200	80	0	54	55	0,131	0,269
56	0	0	0	56	84	0,2268	0,7728
57	30	20	0	56	57	0,5371	1,1029
58	600	420	0	57	58	0,0524	0,1076
59	0	0	0	58	59	0,0405	0,138
60	20	10	0	59	60	0,0393	0,0807
61	20	10	0	60	61	0,0262	0,0538
62	200	130	0	61	62	0,1048	0,2152
63	300	240	0	62	63	0,2358	0,4842
64	300	200	0	63	64	0,0243	0,0828
65	0	0	0	65	84	0,0486	0,1656
66	50	30	0	65	66	0,1703	0,3497
67	0	0	0	66	67	0,1215	0,414
68	400	360	0	67	68	0,2187	0,7452
69	0	0	0	68	69	0,0486	0,1656
70	0	0	0	69	70	0,0729	0,2484
71	2000	1500	0	70	71	0,0567	0,1932
72	200	150	0	71	72	0,0262	0,0528
73	0	0	0	73	84	0,324	1,104
74	0	0	0	73	74	0,0324	0,1104
75	1200	950	0	74	75	0,0567	0,1932
76	300	180	0	75	76	0,0486	0,1656
77	0	0	0	77	84	0,2511	0,8556
78	400	360	0	77	78	0,1296	0,4416
79	2000	1300	0	78	79	0,0486	0,1656
80	200	140	0	79	80	0,131	0,264
81	500	360	0	80	81	0,131	0,264
82	100	30	0	81	82	0,0917	0,1883
83	400	360	0	82	83	0,3144	0,6456
84	0	0	0	5	55	0,131	0,269
				7	60	0,131	0,269
				11	43	0,131	0,269
				12	72	0,3406	0,6994
				13	76	0,4585	0,9415
				14	18	0,5371	1,0824
				16	26	0,0917	0,1883
				20	83	0,0786	0,1614
				28	32	0,0524	0,1076
				29	39	0,0786	0,1614
				34	46	0,0262	0,0538
				40	42	0,1965	0,4035
				53	64	0,0393	0,0807

Fonte: (CHIOU; CHANG; SU, 2005).

C.4 SISTEMA DE 136 BARRAS

As informações iniciais do problema se encontram descritas a seguir, e os dados pertinentes aos ramos de ligação (par de barras a qual fazem parte e impedância) bem como as

informações de cada barra (potência ativa, reativa indutiva e reativa capacitiva) se encontram descritos na Tabela 27.

- Quantidade de barras: 136;
- Quantidade total de ramos de ligação: 156;
- Barra de referência da subestação: 136;
- Tensão na barra de referência em p.u.: 1,0;
- Tensão base da barra de referência: 13,8 kV;
- Potência base: 100.000 kVA;
- Máximo erro entre perdas elétricas: 10^{-8} .

Tabela 27 – Dados do sistema de 136 barras.

Dados de Barras				Dados de Ramos			
Índice	Pot. Ativa (kW)	Pot. Reat. Indutiva (kVAr)	Pot. Reat. Capacitiva (kVAr)	Barra 1	Barra 2	Resistência (Ω)	Reatância (Ω)
136	0	0	0	136	1	0,33205	0,76653
1	0	0	0	1	2	0,00188	0,00433
2	47,78	19,009	0	2	3	0,22324	0,51535
3	42,551	16,929	0	3	4	0,09943	0,22953
4	87,022	34,622	0	4	5	0,15571	0,35945
5	311,31	123,855	0	5	6	0,16321	0,37677
6	148,869	59,228	0	6	7	0,11444	0,26417
7	238,672	94,956	0	6	8	0,05675	0,05666
8	62,299	24,786	0	8	9	0,52124	0,27418
9	124,598	49,571	0	8	10	0,10877	0,1086
10	140,175	55,768	0	10	11	0,39803	0,20937
11	116,813	46,474	0	10	12	0,91744	0,31469
12	249,203	99,145	0	10	13	0,11823	0,11805
13	291,447	115,952	0	13	14	0,50228	0,26421
14	303,72	120,835	0	13	15	0,05675	0,05666
15	215,396	85,695	0	15	16	0,29379	0,15454
16	198,586	79,007	0	136	17	0,33205	0,76653
17	0	0	0	17	18	0,00188	0,00433
18	0	0	0	18	19	0,22324	0,51535
19	0	0	0	19	20	0,10881	0,25118
20	30,127	14,729	0	20	21	0,71078	0,37388
21	230,972	112,92	0	20	22	0,18197	0,42008
22	60,256	29,458	0	22	23	0,30326	0,15952
23	230,972	112,92	0	22	24	0,02439	0,0563
24	120,507	58,915	0	24	25	0,04502	0,10394
25	0	0	0	25	26	0,01876	0,04331
26	56,981	27,857	0	26	27	0,11823	0,11805
27	364,665	178,281	0	27	28	0,02365	0,02361
28	0	0	0	28	29	0,18954	0,0997
29	124,647	60,939	0	29	30	0,39803	0,20937
30	56,981	27,857	0	28	31	0,05675	0,05666
31	0	0	0	31	32	0,09477	0,04985

32	85,473	41,787	0	32	33	0,41699	0,21934
33	0	0	0	33	34	0,11372	0,05982
34	396,735	193,96	0	31	35	0,07566	0,07555
35	0	0	0	35	36	0,3696	0,19442
36	181,152	88,563	0	36	37	0,26536	0,13958
37	242,172	118,395	0	35	38	0,05675	0,05666
38	75,316	36,821	0	136	39	0,33205	0,76653
39	0	0	0	39	40	0,11819	0,27283
40	1,254	0,531	0	40	41	2,96288	1,01628
41	6,274	2,66	0	40	42	0,00188	0,00433
42	0	0	0	42	43	0,06941	0,16024
43	117,88	49,971	0	43	44	0,81502	0,42872
44	62,668	26,566	0	43	45	0,06378	0,14724
45	172,285	73,034	0	45	46	0,13132	0,30315
46	458,556	194,388	0	46	47	0,06191	0,14291
47	262,962	111,473	0	47	48	0,11444	0,26417
48	235,761	99,942	0	48	49	0,28374	0,28331
49	0	0	0	49	50	0,28374	0,28331
50	109,215	46,298	0	48	51	0,04502	0,10394
51	0	0	0	51	52	0,02626	0,06063
52	72,809	30,865	0	52	53	0,06003	0,13858
53	258,473	109,57	0	53	54	0,03002	0,06929
54	69,169	29,322	0	54	55	0,02064	0,04764
55	21,843	9,26	0	52	56	0,10881	0,25118
56	0	0	0	56	57	0,25588	0,1346
57	20,527	8,702	0	57	58	0,41699	0,21934
58	150,548	63,819	0	58	59	0,50228	0,26421
59	220,687	93,552	0	59	60	0,3317	0,17448
60	92,384	39,163	0	60	61	0,20849	0,10967
61	0	0	0	47	62	0,13882	0,32047
62	226,693	96,098	0	136	63	0,0075	0,01732
63	0	0	0	63	64	0,27014	0,62362
64	294,016	116,974	0	64	65	0,3827	0,88346
65	83,015	33,028	0	65	66	0,33018	0,7622
66	83,015	33,028	0	66	67	0,3283	0,75787
67	103,77	41,285	0	67	68	0,17072	0,39409
68	176,408	70,184	0	68	69	0,55914	0,29412
69	83,015	33,028	0	68	70	0,05816	0,13425
70	217,917	86,698	0	70	71	0,7013	0,3689
71	23,294	9,267	0	71	72	1,02352	0,53839
72	5,075	2,019	0	70	73	0,06754	0,15591
73	72,638	28,899	0	73	74	1,32352	0,45397
74	405,99	161,5235	0	136	75	0,01126	0,02598
75	0	0	0	75	76	0,72976	1,68464
76	100,182	42,468	0	76	77	0,22512	0,51968
77	142,523	60,417	0	77	78	0,20824	0,48071
78	96,042	40,713	0	78	79	0,0469	0,10827
79	300,454	127,366	0	79	80	0,6195	0,61857
80	141,238	59,873	0	80	81	0,34049	0,33998
81	279,847	118,631	0	81	82	0,56862	0,29911
82	87,312	37,013	0	81	83	0,10877	0,1086
83	243,849	103,371	0	83	84	0,56862	0,29911
84	247,75	105,025	0	136	85	0,01126	0,02598
85	0	0	0	85	86	0,41835	0,96575
86	89,878	38,101	0	86	87	0,10499	0,13641
87	1137,28	482,108	0	86	88	0,43898	1,01338
88	458,339	194,296	0	88	89	0,0752	0,02579
89	385,197	163,29	0	89	90	0,07692	0,17756

90	0	0	0	90	91	0,33205	0,76653
91	79,608	33,747	0	91	92	0,08442	0,19488
92	87,312	37,013	0	92	93	0,1332	0,30748
93	0	0	0	93	94	0,2932	0,29276
94	74,001	31,37	0	94	95	0,21753	0,21721
95	232,05	98,369	0	95	96	0,26482	0,26443
96	141,819	60,119	0	93	97	0,10318	0,23819
97	0	0	0	97	98	0,13507	0,31181
98	76,449	32,408	0	136	99	0,00938	0,02165
99	0	0	0	99	100	0,16884	0,38976
100	51,322	21,756	0	100	101	0,11819	0,27283
101	59,874	25,381	0	101	102	2,28608	0,78414
102	9,065	3,843	0	101	103	0,45587	1,05236
103	2,092	0,887	0	103	104	0,696	1,60669
104	16,735	7,094	0	104	105	0,45774	1,05669
105	1506,522	638,634	0	105	106	0,20298	0,26373
106	313,023	132,694	0	106	107	0,21348	0,27737
107	79,831	33,842	0	107	108	0,54967	0,28914
108	51,322	21,756	0	108	109	0,54019	0,28415
109	0	0	0	107	110	0,0455	0,05911
110	202,435	85,815	0	110	111	0,47385	0,24926
111	60,823	25,784	0	111	112	0,86241	0,45364
112	45,618	19,338	0	112	113	0,56862	0,29911
113	0	0	0	108	114	0,77711	0,40878
114	157,07	66,584	0	114	115	1,08038	0,5683
115	0	0	0	109	116	1,09933	0,57827
116	250,148	106,041	0	116	117	0,47385	0,24926
117	0	0	0	104	118	0,32267	0,74488
118	69,809	29,593	0	118	119	0,14633	0,33779
119	32,072	13,596	0	119	120	0,12382	0,28583
120	61,084	25,894	0	136	121	0,01126	0,02598
121	0	0	0	121	122	0,6491	1,49842
122	94,622	46,26	0	122	123	0,04502	0,10394
123	49,858	24,375	0	123	124	0,5264	0,18056
124	123,164	60,214	0	123	125	0,02064	0,04764
125	78,35	38,304	0	125	126	0,53071	0,27917
126	145,475	71,121	0	125	127	0,09755	0,2252
127	21,369	10,447	0	127	128	0,11819	0,27283
128	74,789	36,564	0	127	129	0,13882	0,32047
129	227,926	111,431	0	129	130	0,04315	0,09961
130	35,614	17,411	0	130	131	0,09192	0,2122
131	249,295	121,877	0	131	132	0,16134	0,37244
132	316,722	154,842	0	132	133	0,37832	0,37775
133	333,817	163,199	0	133	134	0,39724	0,39664
134	249,295	121,877	0	134	135	0,2932	0,29276
135	0	0	0	7	73	0,13132	0,30315
				9	24	0,26536	0,13958
				15	83	0,14187	0,14166
				38	135	0,08512	0,08499
				25	51	0,04502	0,10394
				50	96	0,14187	0,14166
				55	98	0,14187	0,14166
				62	120	0,0394	0,09094
				66	79	0,12944	0,29882
				79	131	0,01688	0,03898
				84	135	0,3317	0,17448
				91	104	0,14187	0,14166
				90	129	0,07692	0,17756

				90	103	0,07692	0,17756
				92	104	0,07692	0,17756
				92	132	0,07692	0,17756
				96	120	0,26482	0,26443
				110	47	0,49696	0,64567
				126	76	0,17059	0,08973
				128	77	0,05253	0,12126
				135	98	0,2932	0,29276

Fonte: (MANTOVANI; CASARI; ROMERO, 2000).

C.5 SISTEMA DE 415 BARRAS

As informações iniciais do problema se encontram descritas a seguir, e os dados pertinentes aos ramos de ligação (par de barras a qual fazem parte e impedância) bem como as informações de cada barra (potência ativa, reativa indutiva e reativa capacitiva) se encontram descritos na Tabela 28.

- Quantidade de barras: 415;
- Quantidade total de ramos de ligação: 473;
- Barra de referência da subestação: 1;
- Tensão na barra de referência em p.u.: 1,0;
- Tensão base da barra de referência: 10,0 kV;
- Potência base: 100.000 kVA;
- Máximo erro entre perdas elétricas: 10^{-8} .

Tabela 28 – Dados do sistema de 415 barras.

Dados de Barras				Dados de Ramos			
Índice	Pot. Ativa (kW)	Pot. Reat. Indutiva (kVAr)	Pot. Reat. Capacitiva (kVAr)	Barra 1	Barra 2	Resistência (Ω)	Reatância (Ω)
1	0	0	0,00	384	68	0,0310675	0,0214200
2	0,1	0	0,00	89	67	0,1078225	0,0743400
3	122	59	0,00	225	224	0,0694450	0,0478800
4	0,1	0	0,00	21	72	0,1937150	0,1335600
5	0,1	0	0,00	66	384	0,0347225	0,0239400
6	114	55	0,00	3	20	0,6103850	0,4208400
7	51	25	0,00	224	222	0,0694450	0,0478800
8	83	40	0,00	72	74	0,0219300	0,0151200
9	0,1	0	0,00	70	382	0,0310675	0,0214200
10	122	59	0,00	20	51	0,0877200	0,0604800

11	122	59	0,00	222	221	0,0877200	0,0604800
12	77	37	0,00	76	72	0,0548250	0,0378000
13	0,1	0	0,00	382	66	0,0328950	0,0226800
14	68	33	0,00	51	52	0,0694450	0,0478800
15	71	34	0,00	221	220	0,0840650	0,0579600
16	0,1	0	0,00	75	81	0,0840650	0,0579600
17	0,1	0	0,00	59	70	0,0537285	0,0370440
18	77	37	0,00	52	56	0,1169600	0,0806400
19	30	14	0,00	1	273	0,0142545	0,0098280
20	13	6	0,00	81	76	0,0529975	0,0365400
21	1	0	0,00	68	85	0,0292400	0,0201600
22	194	94	0,00	56	27	0,0950300	0,0655200
23	77	37	0,00	1	274	0,1154980	0,0796320
24	77	37	0,00	74	75	0,1059950	0,0730800
25	77	37	0,00	68	93	0,0840650	0,0579600
26	122	59	0,00	27	31	0,1023400	0,0705600
27	104	51	0,00	274	275	0,0836995	0,0577080
28	0,1	0	0,00	72	73	0,0986850	0,0680400
29	77	37	0,00	60	58	0,0109650	0,0075600
30	243	118	0,00	31	40	0,1059950	0,0730800
31	77	37	0,00	275	209	0,1396210	0,0962640
32	77	37	0,00	73	80	0,0731000	0,0504000
33	0,1	0	0,00	65	71	0,0712725	0,0491400
34	121	58	0,00	40	47	0,0913750	0,0630000
35	194	94	0,00	275	276	0,1158635	0,0798840
36	122	59	0,00	73	77	0,0767550	0,0529200
37	77	37	0,00	69	65	0,0310675	0,0214200
38	0,1	0	0,00	51	54	0,0548250	0,0378000
39	0,1	0	0,00	276	264	0,1151325	0,0793800
40	96	47	0,00	77	78	0,0511700	0,0352800
41	122	59	0,00	59	69	0,0211990	0,0146160
42	70	34	0,00	54	24	0,0731000	0,0504000
43	58	28	0,00	263	264	0,0687140	0,0473760
44	0,1	0	0,00	79	83	0,2412300	0,1663200
45	122	59	0,00	63	61	0,1096500	0,0756000
46	122	59	0,00	24	29	0,0950300	0,0655200
47	122	59	0,00	262	263	0,0630936	0,0532656
48	86	42	0,00	82	80	0,0731000	0,0504000
49	77	37	0,00	71	63	0,0226610	0,0156240
50	0,1	0	0,00	29	34	0,1151325	0,0793800
51	122	59	0,00	261	262	0,0379730	0,0320580
52	122	59	0,00	78	79	0,0219300	0,0151200
53	122	59	0,00	61	60	0,0438600	0,0302400
54	0,1	0	0,00	34	47	0,1736125	0,1197000
55	42	20	0,00	259	261	0,0511175	0,0431550
56	122	59	0,00	83	90	0,1209805	0,0834120
57	13	6	0,00	58	62	0,1293870	0,0892080
58	77	37	0,00	47	21	0,0840650	0,0579600
59	50	24	0,00	257	259	0,0687140	0,0473760
60	0,1	0	0,00	77	59	0,3874300	0,2671200

61	60	29	0,00	64	67	0,0449565	0,0309960
62	71	34	0,00	20	44	0,0146200	0,0100800
63	122	59	0,00	372	239	0,0265811	0,0224406
64	77	37	0,00	123	1	0,0135240	0,0159530
65	95	46	0,00	67	68	0,0862580	0,0594720
66	121	58	0,00	44	25	0,1279250	0,0882000
67	77	37	0,00	264	265	0,0723690	0,0498960
68	103	50	0,00	120	94	0,2538284	0,2584552
69	52	25	0,00	62	64	0,0603075	0,0415800
70	57	27	0,00	25	35	0,1279250	0,0882000
71	171	83	0,00	369	347	0,1315800	0,0907200
72	121	58	0,00	120	123	0,2948750	0,3002500
73	39	19	0,00	58	20	0,5921100	0,4082400
74	119	58	0,00	35	38	0,0804100	0,0554400
75	49	24	0,00	369	370	0,2083350	0,1436400
76	0,1	0	0,00	114	115	0,0475150	0,0327600
77	50	24	0,00	1	92	0,3888920	0,2681280
78	44	21	0,00	38	41	0,1041675	0,0718200
79	82	40	0,00	370	348	0,1187875	0,0819000
80	132	64	0,00	115	112	0,0402050	0,0277200
81	194	94	0,00	92	2	0,3410115	0,2351160
82	0,1	0	0,00	41	43	0,1169600	0,0806400
83	122	59	0,00	383	251	0,0109650	0,0075600
84	3	1	0,00	94	113	0,0328950	0,0226800
85	52	25	0,00	92	87	0,0128524	0,0108504
86	103	50	0,00	43	48	0,0804100	0,0554400
87	21	10	0,00	383	370	0,0182750	0,0126000
88	21	10	0,00	113	114	0,0767550	0,0529200
89	21	10	0,00	2	17	0,0877200	0,0604800
90	5	3	0,00	48	50	0,0731000	0,0504000
91	0,1	0	0,00	208	277	0,0672520	0,0463680
92	0,1	0	0,00	106	95	0,0840650	0,0579600
93	21	10	0,00	6	7	0,0402050	0,0277200
94	18	9	0,00	20	33	0,0657900	0,0453600
95	122	59	0,00	277	278	0,0328950	0,0226800
96	116	56	0,00	95	112	0,0328950	0,0226800
97	7	3	0,00	7	3	0,0519010	0,0357840
98	64	31	0,00	84	33	0,0420325	0,0289800
99	49	24	0,00	277	279	0,0482460	0,0332640
100	16	8	0,00	94	111	0,0657900	0,0453600
101	77	37	0,00	18	5	0,1224425	0,0844200
102	119	58	0,00	28	36	0,0913750	0,0630000
103	71	34	0,00	279	280	0,0343570	0,0236880
104	77	37	0,00	111	106	0,0657900	0,0453600
105	194	94	0,00	5	6	0,0493425	0,0340200
106	122	59	0,00	36	45	0,1315800	0,0907200
107	55	27	0,00	280	281	0,0621350	0,0428400
108	38	18	0,00	119	121	0,0102235	0,0086310
109	22	10	0,00	17	18	0,1169600	0,0806400
110	0,1	0	0,00	49	45	0,0438600	0,0302400

111	77	37	0,00	281	235	0,0493425	0,0340200
112	122	59	0,00	94	119	0,0560280	0,0660910
113	38	18	0,00	2	14	0,0164475	0,0113400
114	122	59	0,00	55	23	0,0767550	0,0529200
115	27	13	0,00	281	282	0,0537285	0,0370440
116	64	31	0,00	118	84	0,0402050	0,0277200
117	0,1	0	0,00	8	9	0,0950300	0,0655200
118	0,1	0	0,00	23	28	0,0694450	0,0478800
119	0,1	0	0,00	235	234	0,0833340	0,0574560
120	0,1	0	0,00	119	122	0,0511700	0,0352800
121	79	38	0,00	9	3	0,0146200	0,0100800
122	247	119	0,00	33	55	0,0438600	0,0302400
123	0,1	0	0,00	235	236	0,0420325	0,0289800
124	0,1	0	0,00	122	118	0,2010250	0,1386000
125	0,1	0	0,00	16	19	0,1571650	0,1083600
126	0,1	0	0,00	21	50	0,0365500	0,0252000
127	0,1	0	0,00	208	236	0,0767550	0,0529200
128	0,1	0	0,00	104	105	0,0496570	0,0419220
129	77	37	0,00	19	8	0,0219300	0,0151200
130	122	59	0,00	32	44	0,2558500	0,1764000
131	62	30	0,00	232	349	0,0759460	0,0641160
132	105	51	0,00	84	104	0,0731000	0,0504000
133	122	59	0,00	14	16	0,0986850	0,0680400
134	77	37	0,00	37	32	0,1059950	0,0730800
135	77	37	0,00	349	207	0,0321310	0,0271260
136	0,1	0	0,00	105	108	0,0657900	0,0453600
137	85	41	0,00	2	13	0,0321310	0,0271260
138	8	4	0,00	42	37	0,0676175	0,0466200
139	122	59	0,00	380	229	0,0411861	0,0347706
140	35	17	0,00	109	108	0,0496570	0,0419220
141	55	27	0,00	4	12	0,0671830	0,0567180
142	153	74	0,00	46	42	0,0255850	0,0176400
143	316	153	0,00	229	228	0,0630936	0,0532656
144	105	51	0,00	84	109	0,0475150	0,0327600
145	20	10	0,00	12	15	0,1051560	0,0887760
146	95	46	0,00	46	50	0,0365500	0,0252000
147	38	18	0,00	232	231	0,0968575	0,0667800
148	83	40	0,00	110	84	0,0438600	0,0302400
149	50	24	0,00	15	11	0,1329055	0,1122030
150	77	37	0,00	21	49	0,1352350	0,0932400
151	122	59	0,00	232	293	0,0749275	0,0516600
152	102	49	0,00	103	110	0,0767550	0,0529200
153	60	29	0,00	13	4	0,0452755	0,0382230
154	77	37	0,00	39	30	0,1169600	0,0806400
155	194	94	0,00	293	236	0,0694450	0,0478800
156	56	27	0,00	107	103	0,0548250	0,0378000
157	122	59	0,00	11	86	0,0350520	0,0295920
158	0,1	0	0,00	49	39	0,0877200	0,0604800
159	3	1	0,00	221	344	0,0292400	0,0201600
160	1	0	0,00	84	107	0,0657900	0,0453600

161	52	25	0,00	86	91	0,0950300	0,0655200
162	194	94	0,00	53	22	0,0402050	0,0277200
163	10	5	0,00	304	306	0,0694450	0,0478800
164	128	62	0,00	97	116	0,0438600	0,0302400
165	16	8	0,00	10	11	0,0467360	0,0394560
166	37	18	0,00	30	26	0,0548250	0,0378000
167	154	75	0,00	378	377	0,1114775	0,0768600
168	0,1	0	0,00	84	99	0,1315800	0,0907200
169	0,1	0	0,00	3	10	0,1078225	0,0743400
170	78	38	0,00	26	53	0,1005125	0,0693000
171	0,1	0	0,00	304	378	0,0588455	0,0405720
172	27	13	0,00	101	96	0,1059950	0,0730800
173	103	50	0,00	10	88	0,1095375	0,0924750
174	122	59	0,00	22	33	0,0438600	0,0302400
175	72	35	0,00	309	307	0,0986850	0,0680400
176	0,1	0	0,00	100	101	0,0584800	0,0403200
177	122	59	0,00	88	89	0,0804100	0,0554400
178	194	94	0,00	21	57	0,2156450	0,1486800
179	0,1	0	0,00	314	317	0,0815065	0,0561960
180	77	37	0,00	102	100	0,0548250	0,0378000
181	82	40	0,00	99	102	0,0511700	0,0352800
182	53	26	0,00	187	167	0,0847090	0,0715140
183	122	59	0,00	314	318	0,0723690	0,0498960
184	64	31	0,00	138	194	0,0127925	0,0088200
185	86	42	0,00	96	117	0,0785825	0,0541800
186	82	40	0,00	167	175	0,0905510	0,0764460
187	0,1	0	0,00	301	305	0,0438600	0,0302400
188	167	81	0,00	155	151	0,0694450	0,0478800
189	0,1	0	0,00	117	97	0,0402050	0,0277200
190	0,1	0	0,00	175	181	0,0847090	0,0715140
191	131	64	0,00	310	315	0,0614040	0,0423360
192	21	10	0,00	126	124	0,0292400	0,0201600
193	21	10	0,00	98	94	0,0402050	0,0277200
194	21	10	0,00	160	180	0,0438600	0,0302400
195	21	10	0,00	310	305	0,0826030	0,0569520
196	0,1	0	0,00	1	351	0,2266100	0,1562400
197	0,1	0	0,00	116	98	0,0621350	0,0428400
198	0,1	0	0,00	180	177	0,0548250	0,0378000
199	0,1	0	0,00	129	346	0,2616980	0,1804320
200	0,1	0	0,00	1	350	0,2266100	0,1562400
201	52	25	0,00	57	147	0,1571650	0,1083600
202	202	98	0,00	169	162	0,0767550	0,0529200
203	202	98	0,00	90	370	0,1578960	0,1088640
204	73	35	0,00	350	203	0,0233920	0,0161280
205	81	39	0,00	153	141	0,0730250	0,0616500
206	115	56	0,00	162	183	0,0840650	0,0579600
207	118	57	0,00	382	383	0,1502205	0,1035720
208	318	154	0,00	351	379	0,0233920	0,0161280
209	161	78	0,00	147	142	0,1059950	0,0730800
210	126	61	0,00	183	182	0,0328950	0,0226800

211	59	29	0,00	66	252	0,1929840	0,1330560
212	24	12	0,00	203	204	0,0511175	0,0431550
213	24	12	0,00	142	136	0,0804100	0,0554400
214	5	2	0,00	160	181	0,0913750	0,0630000
215	126	61	0,00	384	243	0,2273410	0,1567440
216	56	27	0,00	204	205	0,0621350	0,0428400
217	62	30	0,00	132	153	0,0877200	0,0604800
218	0,1	0	0,00	181	178	0,0584800	0,0403200
219	32	15	0,00	85	244	0,1399865	0,0965160
220	158	77	0,00	203	206	0,1034034	0,0872964
221	126	61	0,00	136	132	0,1498550	0,1033200
222	202	98	0,00	171	163	0,0642620	0,0542520
223	41	20	0,00	352	414	0,0705415	0,0486360
224	81	39	0,00	379	380	0,1034034	0,0872964
225	126	61	0,00	57	149	0,0694450	0,0478800
226	202	98	0,00	163	184	0,0759460	0,0641160
227	18	9	0,00	414	385	0,0720035	0,0496440
228	202	98	0,00	206	207	0,0598805	0,0505530
229	106	51	0,00	57	125	0,0913750	0,0630000
230	175	85	0,00	160	129	0,2278380	0,1923480
231	174	84	0,00	414	401	0,0727345	0,0501480
232	113	55	0,00	1	375	0,0109650	0,0075600
233	202	98	0,00	127	124	0,0292400	0,0201600
234	0,1	0	0,00	129	141	0,0321310	0,0271260
235	202	98	0,00	401	386	0,0592110	0,0408240
236	403	195	0,00	208	209	0,0383775	0,0264600
237	202	98	0,00	128	127	0,2704700	0,1864800
238	126	61	0,00	131	157	0,0365500	0,0252000
239	0,1	0	0,00	401	402	0,0709070	0,0488880
240	29	14	0,00	209	210	0,1169600	0,0806400
241	9	4	0,00	125	128	0,0219300	0,0151200
242	5	2	0,00	130	131	0,0548250	0,0378000
243	0,1	0	0,00	402	387	0,0566525	0,0390600
244	18	9	0,00	209	211	0,0592115	0,0385560
245	45	22	0,00	200	119	0,0767550	0,0529200
246	32	16	0,00	141	130	0,0467360	0,0394560
247	61	30	0,00	402	403	0,0595765	0,0410760
248	81	39	0,00	211	1	0,0562870	0,0388080
249	126	61	0,00	200	201	0,0073100	0,0050400
250	23	11	0,00	157	133	0,1133050	0,0781200
251	14	7	0,00	403	388	0,0833340	0,0574560
252	18	9	0,00	375	202	0,1498550	0,1033200
253	126	61	0,00	158	200	0,0219300	0,0151200
254	5	3	0,00	133	140	0,1059950	0,0730800
255	26	13	0,00	403	404	0,0683485	0,0471240
256	18	9	0,00	202	1	0,4824600	0,3326400
257	126	61	0,00	124	158	0,1023400	0,0705600
258	81	39	0,00	140	147	0,1535100	0,1058400
259	0,1	0	0,00	368	345	0,1151325	0,0793800
260	126	61	0,00	373	352	0,1041675	0,0718200

261	41	20	0,00	199	192	0,0146200	0,0100800
262	0,1	0	0,00	129	154	0,0350520	0,0295920
263	28	14	0,00	345	346	0,0584800	0,0403200
264	30	14	0,00	352	212	0,1136705	0,0783720
265	81	39	0,00	199	197	0,0694450	0,0478800
266	202	98	0,00	137	144	0,0950300	0,0655200
267	318	154	0,00	345	369	0,0529975	0,0365400
268	176	85	0,00	352	353	0,2138175	0,1474200
269	126	61	0,00	159	199	0,2704700	0,1864800
270	6	3	0,00	144	148	0,0950300	0,0655200
271	31	15	0,00	400	404	0,0475150	0,0327600
272	60	29	0,00	353	213	0,2010250	0,1386000
273	56	27	0,00	197	191	0,0109650	0,0075600
274	126	61	0,00	154	156	0,0321310	0,0271260
275	118	57	0,00	404	405	0,0402050	0,0277200
276	126	61	0,00	353	214	0,0500735	0,0345240
277	202	98	0,00	159	190	0,2359000	0,2402000
278	10	5	0,00	156	137	0,0840650	0,0579600
279	126	61	0,00	405	389	0,0588455	0,0405720
280	202	98	0,00	1	215	0,3475905	0,2396520
281	318	154	0,00	159	161	0,0840650	0,0579600
282	82	40	0,00	143	150	0,0986850	0,0680400
283	126	61	0,00	405	406	0,0716380	0,0493920
284	22	10	0,00	215	216	0,1059950	0,0730800
285	13	6	0,00	161	172	0,0877200	0,0604800
286	71	34	0,00	150	148	0,0219300	0,0151200
287	101	49	0,00	406	390	0,0453220	0,0312480
288	113	54	0,00	216	217	0,0993140	0,0838440
289	126	61	0,00	172	188	0,0913750	0,0630000
290	0,1	0	0,00	134	139	0,1197610	0,1011060
291	34	17	0,00	406	391	0,0731000	0,0504000
292	83	40	0,00	216	218	0,0657900	0,0453600
293	59	29	0,00	188	168	0,0913750	0,0630000
294	126	61	0,00	139	143	0,0621350	0,0428400
295	145	70	0,00	404	407	0,0833340	0,0574560
296	202	98	0,00	218	219	0,0029240	0,0020160
297	50	24	0,00	176	179	0,0657900	0,0453600
298	41	20	0,00	154	134	0,0788670	0,0665820
299	202	98	0,00	407	392	0,0785825	0,0541800
300	77	37	0,00	219	220	0,0804100	0,0554400
301	161	78	0,00	168	176	0,1133050	0,0781200
302	477	231	0,00	148	57	0,1462000	0,1008000
303	68	33	0,00	407	408	0,0489770	0,0337680
304	14	7	0,00	220	223	0,0709803	0,0599238
305	4	2	0,00	161	185	0,1315800	0,0907200
306	0,1	0	0,00	129	151	0,0328950	0,0226800
307	66	32	0,00	408	409	0,0668865	0,0461160
308	0,1	0	0,00	223	237	0,0817880	0,0690480
309	60	29	0,00	185	164	0,0913750	0,0630000
310	95	46	0,00	145	146	0,0328950	0,0226800

311	0,1	0	0,00	409	393	0,0738310	0,0509040
312	19	9	0,00	237	238	0,0622173	0,0525258
313	67	32	0,00	164	165	0,0328950	0,0226800
314	18	9	0,00	146	149	0,0548250	0,0378000
315	33	16	0,00	409	394	0,0690795	0,0476280
316	21	10	0,00	237	372	0,0537464	0,0453744
317	33	16	0,00	165	173	0,1388900	0,0957600
318	42	20	0,00	152	135	0,0767550	0,0529200
319	52	25	0,00	408	410	0,0551905	0,0380520
320	43	21	0,00	239	240	0,0058420	0,0049320
321	126	61	0,00	173	179	0,1133050	0,0781200
322	107	52	0,00	135	145	0,1425450	0,0982800
323	202	98	0,00	410	415	0,0657900	0,0453600
324	126	61	0,00	239	243	0,0835406	0,0705276
325	24	12	0,00	179	160	0,0694450	0,0478800
326	101	49	0,00	135	195	0,0091375	0,0063000
327	81	39	0,00	410	411	0,0643280	0,0443520
328	155	75	0,00	243	355	0,0054825	0,0037800
329	23	11	0,00	159	182	0,0540960	0,0638120
330	31	15	0,00	151	152	0,0548250	0,0378000
331	126	61	0,00	411	395	0,0595765	0,0410760
332	13	6	0,00	355	354	0,1279250	0,0882000
333	56	27	0,00	182	186	0,0483000	0,0569750
334	120	58	0,00	1	190	0,1415400	0,1441200
335	126	61	0,00	411	412	0,0676175	0,0466200
336	403	195	0,00	354	242	0,0182750	0,0126000
337	81	39	0,00	186	166	0,0377440	0,0384320
338	177	86	0,00	190	189	0,0731000	0,0504000
339	151	73	0,00	412	399	0,0566525	0,0390600
340	9	4	0,00	354	241	0,0731000	0,0504000
341	29	14	0,00	166	170	0,0377440	0,0384320
342	14	7	0,00	189	198	0,1096500	0,0756000
343	23	11	0,00	412	398	0,0716380	0,0493920
344	45	22	0,00	243	356	0,0091375	0,0063000
345	0,1	0	0,00	170	174	0,0292100	0,0246600
346	36	17	0,00	198	196	0,0365500	0,0252000
347	5	2	0,00	412	413	0,0873545	0,0602280
348	9	4	0,00	356	357	0,0511700	0,0352800
349	80	39	0,00	174	180	0,0636930	0,0648540
350	0,1	0	0,00	189	193	0,0233920	0,0161280
351	0,1	0	0,00	413	397	0,0705415	0,0486360
352	0,1	0	0,00	357	244	0,0127925	0,0088200
353	0,1	0	0,00	159	184	0,2266100	0,1562400
354	0,1	0	0,00	149	138	0,1279250	0,0882000
355	0,1	0	0,00	413	396	0,0694450	0,0478800
356	0,1	0	0,00	357	245	0,0054825	0,0037800
357	0,1	0	0,00	184	187	0,0350520	0,0295920
358	0,1	0	0,00	138	155	0,1059950	0,0730800
359	0,1	0	0,00	290	289	0,0628660	0,0433440
360	0,1	0	0,00	243	246	0,0365125	0,0308250

361	0,1	0	0,00	246	247	0,0526320	0,0362880
362	0,1	0	0,00	265	266	0,1078225	0,0743400
363	0,1	0	0,00	289	288	0,0467840	0,0322560
364	0,1	0	0,00	306	310	0,1264630	0,0871920
365	0,1	0	0,00	247	248	0,0727345	0,0501480
366	0,1	0	0,00	266	267	0,0994160	0,0685440
367	0,1	0	0,00	288	300	0,1867705	0,1287720
368	0,1	0	0,00	310	320	0,0127925	0,0088200
369	0,1	0	0,00	243	249	0,0487807	0,0411822
370	0,1	0	0,00	267	268	0,1564340	0,1078560
371	0,1	0	0,00	300	286	0,1096500	0,0756000
372	0,1	0	0,00	310	314	0,0723690	0,0498960
373	0,1	0	0,00	249	358	0,0431290	0,0297360
374	0,1	0	0,00	268	269	0,0549647	0,0559666
375	0,1	0	0,00	1	373	0,0277780	0,0191520
376	0,1	0	0,00	314	309	0,0698105	0,0481320
377	86	41	0,00	358	359	0,0054825	0,0037800
378	86	41	0,00	269	365	0,0707700	0,0720600
379	0,1	0	0,00	300	285	0,0811410	0,0559440
380	0,1	0	0,00	309	313	0,0785825	0,0541800
381	0,1	0	0,00	359	250	0,0201025	0,0138600
382	0,1	0	0,00	365	366	0,0255850	0,0176400
383	0,1	0	0,00	300	301	0,0932025	0,0642600
384	0,1	0	0,00	313	317	0,0683485	0,0471240
385	68	33	0,00	359	251	0,1407175	0,0970200
386	15	7	0,00	366	270	0,1005125	0,0693000
387	56	27	0,00	301	302	0,0555560	0,0383040
388	12	6	0,00	317	319	0,1235390	0,0851760
389	50	24	0,00	251	252	0,1882325	0,1297800
390	115	56	0,00	366	271	0,0164475	0,0113400
391	14	7	0,00	302	303	0,0453220	0,0312480
392	5	2	0,00	319	318	0,0665210	0,0458640
393	19	9	0,00	251	360	0,0986850	0,0680400
394	34	17	0,00	365	363	0,0483595	0,0492410
395	10	5	0,00	303	304	0,1951770	0,1345680
396	40	19	0,00	318	315	0,0694450	0,0478800
397	28	14	0,00	360	253	0,0497080	0,0342720
398	14	7	0,00	363	362	0,0188720	0,0192160
399	17	8	0,00	304	305	0,2083350	0,1436400
400	41	20	0,00	315	311	0,0833340	0,0574560
401	0,1	0	0,00	253	254	0,0458597	0,0387162
402	0,1	0	0,00	363	381	0,2631600	0,1814400
403	0,1	0	0,00	305	308	0,0504390	0,0347760
404	0,1	0	0,00	311	304	0,1052640	0,0725760
405	0,1	0	0,00	254	361	0,0588455	0,0405720
406	0,1	0	0,00	362	272	0,0164475	0,0113400
407	0,1	0	0,00	308	316	0,2218585	0,1529640
408	0,1	0	0,00	296	260	0,0478805	0,0330120
409	0,1	0	0,00	361	256	0,0402050	0,0277200
410	0,1	0	0,00	362	1	0,0241500	0,0284875

411	0,1	0	0,00	316	312	0,1052640	0,0725760
412	0,1	0	0,00	260	287	0,0358190	0,0246960
413	0,1	0	0,00	254	255	0,0274125	0,0189000
414	0,1	0	0,00	1	364	0,2850900	0,1965600
415	19	9	0,00	312	307	0,0723690	0,0498960
				260	324	0,1078225	0,0743400
				254	257	0,0788670	0,0665820
				364	284	0,1041675	0,0718200
				307	300	0,0782170	0,0539280
				324	326	0,0054825	0,0037800
				257	258	0,1059950	0,0730800
				364	285	0,0584800	0,0403200
				334	336	0,0712725	0,0491400
				324	266	0,1038020	0,0715680
				257	283	0,0537464	0,0453744
				381	268	0,3095785	0,2134440
				336	323	0,0548250	0,0378000
				324	325	0,0482460	0,0332640
				283	282	0,0663067	0,0559782
				268	327	0,0383775	0,0264600
				336	337	0,0646935	0,0446040
				324	323	0,0877200	0,0604800
				282	234	0,0554990	0,0468540
				327	328	0,0548250	0,0378000
				337	338	0,0292400	0,0201600
				323	322	0,0676175	0,0466200
				234	233	0,0277495	0,0234270
				268	297	0,0504390	0,0347760
				338	339	0,0420325	0,0289800
				322	321	0,0950300	0,0655200
				233	376	0,0116840	0,0098640
				297	296	0,0504390	0,0347760
				339	265	0,0679830	0,0468720
				321	299	0,0701760	0,0483840
				230	231	0,0455676	0,0384696
				296	295	0,0741965	0,0511560
				265	367	0,0087720	0,0060480
				299	296	0,0380120	0,0262080
				230	228	0,0262890	0,0221940
				296	298	0,0376465	0,0259560
				367	340	0,0010965	0,0007560
				322	329	0,0314330	0,0216720
				228	371	0,0467360	0,0394560
				295	294	0,0226610	0,0156240
				340	341	0,0274125	0,0189000
				329	330	0,0266815	0,0183960
				371	374	0,0394335	0,0332910
				294	292	0,0917405	0,0632520
				341	342	0,1133050	0,0781200
				330	331	0,0416670	0,0287280

				374	227	0,0036550	0,0025200
				268	292	0,0986850	0,0680400
				334	335	0,0559215	0,0385560
				331	332	0,0274125	0,0189000
				374	226	0,0023368	0,0019728
				292	290	0,0168130	0,0115920
				341	368	0,1790950	0,1234800
				329	333	0,0584800	0,0403200
				226	225	0,0274125	0,0189000
				290	291	0,0054825	0,0037800
				376	230	0,0277495	0,0234270
				333	334	0,0416670	0,0287280
				368	343	0,0091375	0,0063000

Fonte: (I. J. RAMIREZ-ROSADO, 1998).

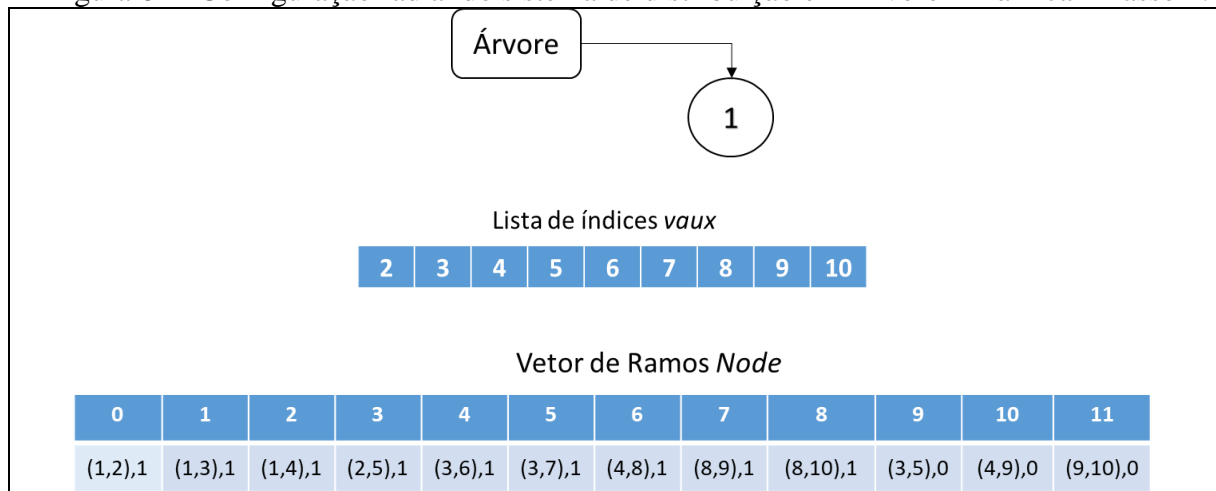
ANEXO D – ILUSTRAÇÃO DO PROCEDIMENTO DE CONSTRUÇÃO DE UMA CONFIGURAÇÃO RADIAL EM ÁRVORE DINÂMICA

Nss Figuras 61-74 descreve-se, passo a passo, o procedimento adotado para a construção de uma solução em uma árvore dinâmica para um problema RSDEE fictício de 10 barras, sendo a barra de índice 1 configurada como a subestação.

A lista de índices *vaux* é construída a cada necessidade de construção de uma solução radial na árvore dinâmica. A lista é formada por todos os índices das barras, menos o da subestação. O vetor de ramos *Node*, onde se encontra todos os elementos (barras) do sistema, possui em seus elementos uma configuração de chaves dos ramos de ligação que representa uma solução radial. O objetivo é se transferir a configuração de ramos de ligação com chave fechada contidos no vetor de ramos *Node* para a estrutura dinâmica de árvore. O vetor de ramos *Node* apresentado pelas Figuras 61-74 possui a informação do par de índices de barras que perfazem um determinado ramo de ligação e a característica da chave daquele ramo de ligação, 0-aberta e 1-fechada.

O procedimento inicia se construindo a lista dinâmica de índices de barras *vaux* sem a presença do índice da barra configurada como subestação, como ilustrado na Figura 61. A árvore possui um único nó fixo, que representa o índice da barra da subestação. O vetor de ramos *Node* é imutável até que uma nova solução vizinha seja construída, procedimento este posterior à construção da solução pela árvore dinâmica e cálculo de suas perdas elétricas.

Figura 61 - Configuração radial do sistema de distribuição em Árvore Dinâmica – Passo 1.

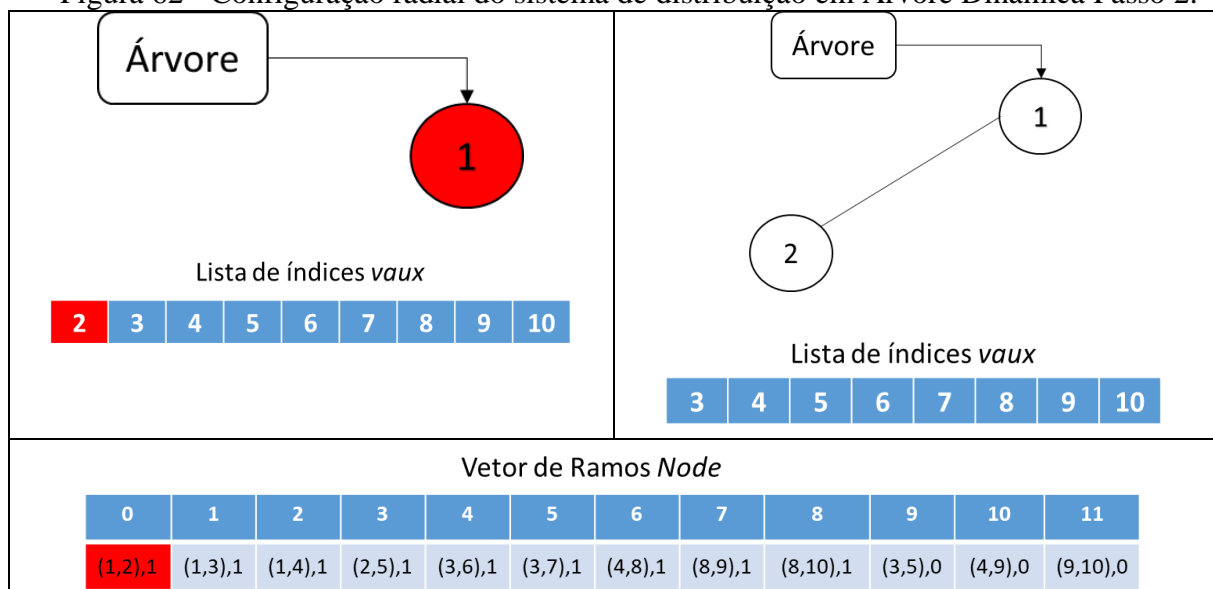


Fonte: Próprio autor

O procedimento inicia pelo primeiro nó da árvore, percorrendo e utilizando os índices da lista *vaux* e para cada elemento *vaux* se busca no vetor de ramos *Node* um ramo de ligação com chave fechada que possua como índices de barra os índices presentes no nó da árvore avaliado e no *vaux*.

Na Figura 62, marcados em vermelho, o par de índices formado pelo nó da árvore e o primeiro elemento de *vaux* já encontrou um ramo de ligação com chave fechada (1-2) na busca no vetor de ramos *Node* (imagem da esquerda). Assim a barra de índice “2” é inserida como nó filho do nó da árvore dinâmica de barra com índice “1”, e o índice de barra “2” é removido de *vaux*, informando, dessa maneira, que esta barra já foi conectada à árvore dinâmica (imagem da direita).

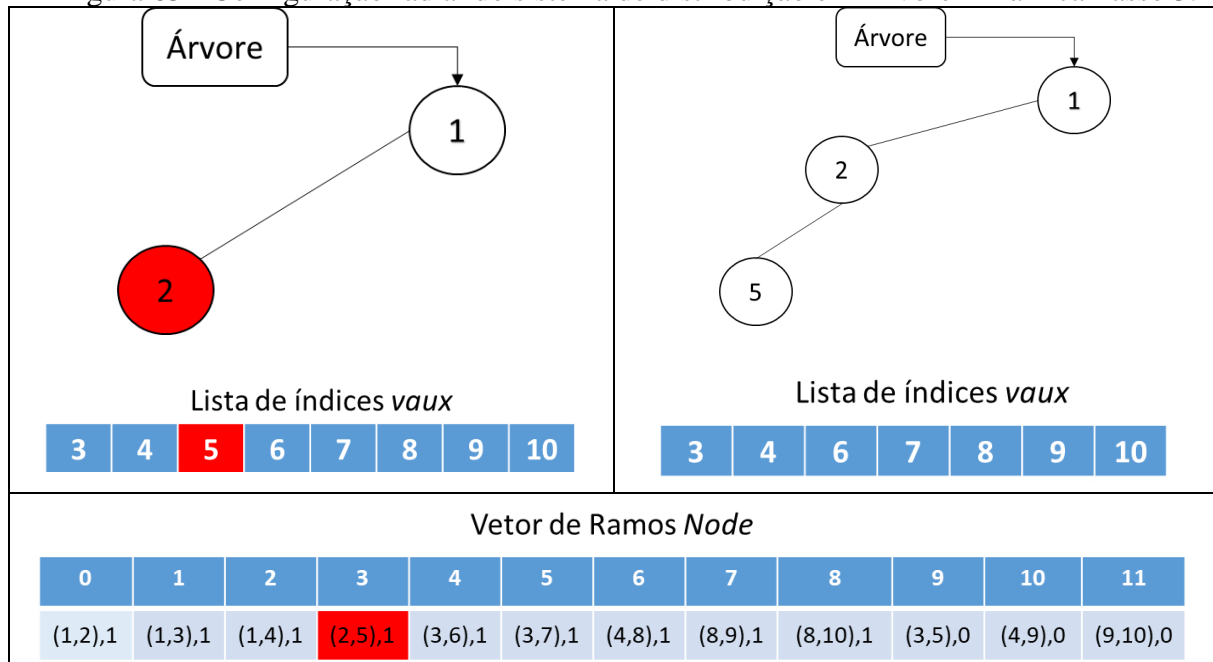
Figura 62 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 2.



Fonte: Próprio autor

Partindo do estágio anterior (Figura 62), agora o nó a ser avaliado é o filho do nó de índice “1” da árvore dinâmica, o nó de índice “2”. Na Figura 63 a lista *vaux* apresenta somente o índice de barra “5” marcado, o que indica a busca de um ramo de ligação com chave no vetor de ramos *Node* para os pares de índice (2-3) e (2-4) não existe. A busca retornou que o ramo (2-5) perfaz as condições adequadas (imagem da esquerda), e então o índice de barra “5” de *vaux* é inserido como filho do nó que está sendo avaliado (índice de barra “2”), e então removido de *vaux* (imagem da direita).

Figura 63 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 3.



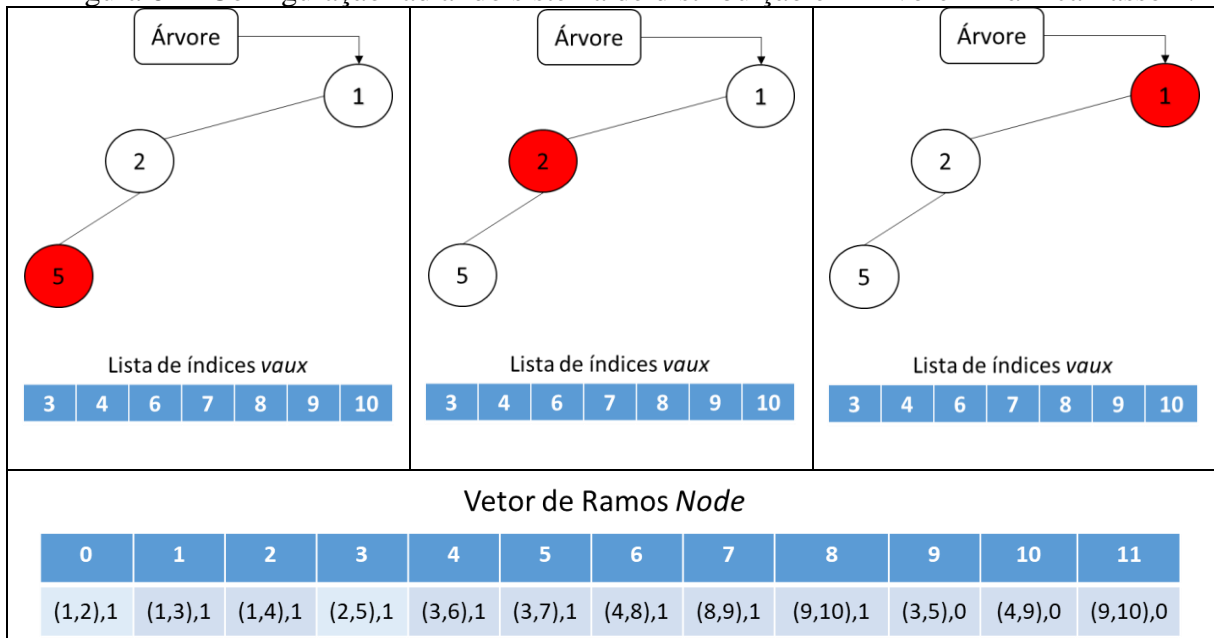
Fonte: Próprio autor.

O procedimento continua a partir do estágio anterior (Figura 63). Na Figura 64 o nó a ser avaliado neste momento (Figura 64 - imagem da esquerda) é o nó de índice “5”, filho do nó de índice “2”. Todas as comparações entre o nó com índice de barra “5” e todos os índices de barra de *vaux* foram realizadas e confrontadas com os ramos de ligação do vetor de ramos *Node* e nenhum ramo de ligação adequado foi encontrado, indicando que para a barra de índice “5” não existem ramos de ligação com chave fechada (5-3; 5-4; 5-6; 5-7; 5-8; 5-9; 5-10).

A característica da recursividade faz com que após uma função ter terminado seu processo ela retorna ao ponto onde foi acionada, que neste caso é o nó com índice de barra “2” (Figura 64 - imagem do meio). Novamente são realizadas todas as comparações entre o índice de barra “2” e os índices de *vaux* contra os elementos do vetor de ramos *Node* na tentativa de se verificar se existe mais ramos de ligação com chave fechada que possuem como algum de suas barras a barra de índice “2”.

Assim como resultado do nó de índice “5”, não foi encontrado um ramo de ligação com chave fechada que possua a configuração adequada de um par de índices formado pelo índice do nó avaliado “2” e os índices de barra de *vaux* (2-3; 2-4; 2-6; 2-7; 2-8; 2-9; 2-10).

Figura 64 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 4.



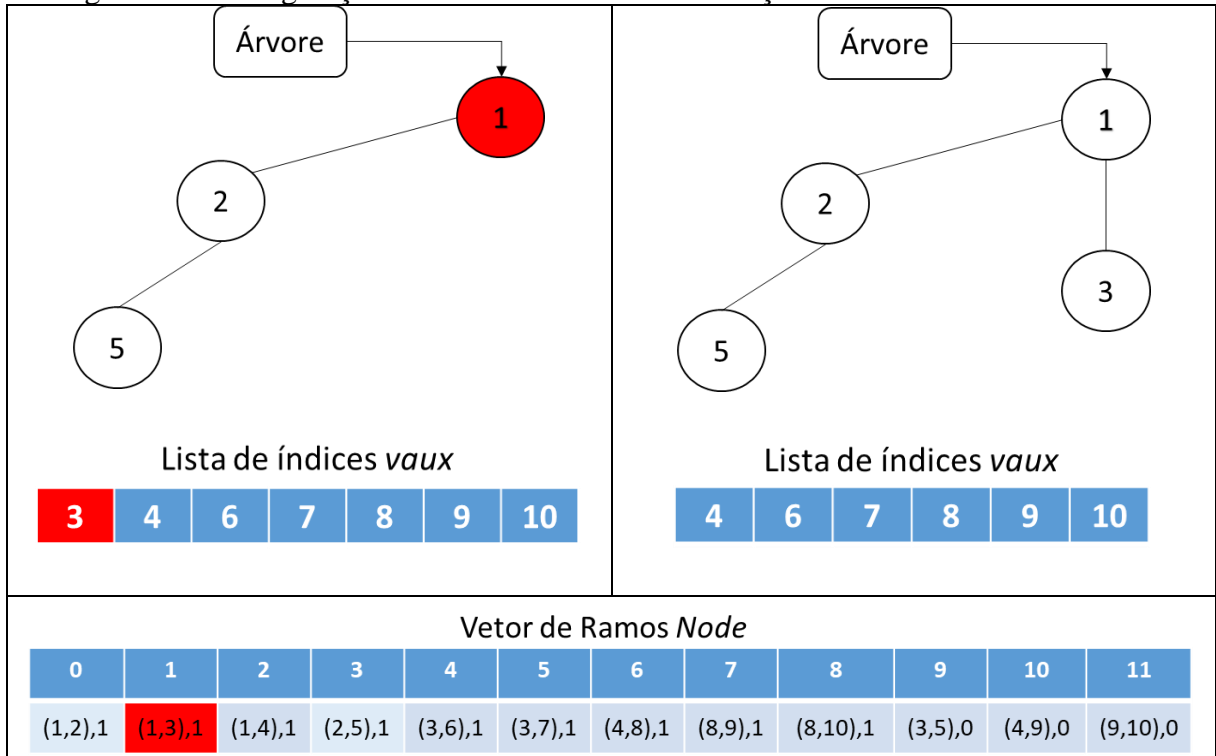
Fonte: Próprio autor.

Pela recursividade o procedimento termina e retoma o ponto onde foi acionado, retornando ao nó de índice “1” (Figura 64 - imagem da direita), que passa por uma nova verificação de existência de um ramo de ligação com chave fechada onde esteja presente a barra de índice “1”.

Continuando o procedimento anterior (Figura 64), a Figura 65 ilustra a continuação do procedimento onde partindo-se do nó de índice “1”, o par de índices (1-3) entre os índices do nó avaliado e de *vaux* encontra durante a varredura do vetor de ramos *Node* um ramo com os mesmos índices de barra e com sua chave de circuito fechada (Figura 65 - imagem da esquerda). Sendo um ramo válido, o índice de barra “3” de *vaux* é inserido como um novo filho do nó de índice “1” avaliado na árvore, remove-se o nó de índice “3” da lista *vaux*, e se repassa a nova busca para o nó filho de índice “3” recentemente inserido na árvore (Figura 65- imagem da direita).

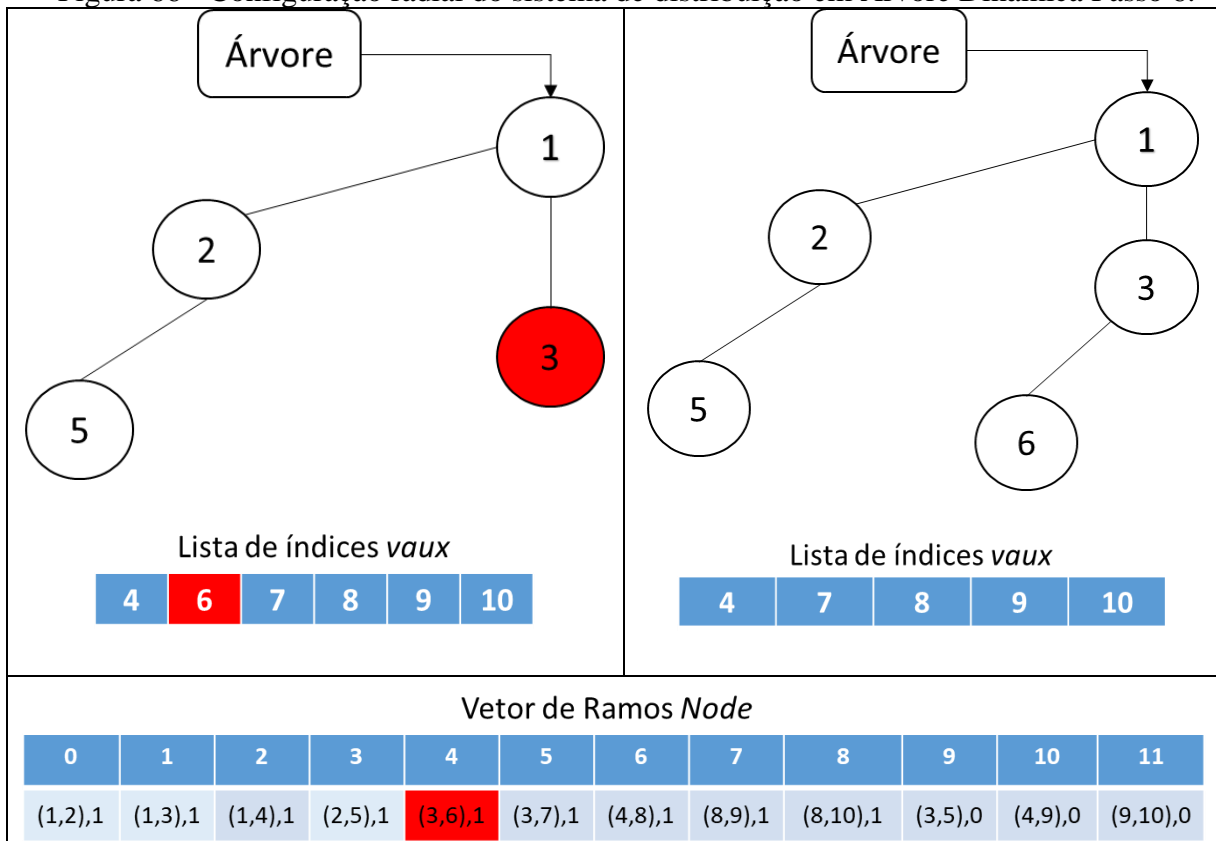
Na Figura 66, partindo-se do nó da árvore em avaliação com índice “3”, a comparação (3-4) não retornou um ramo de ligação adequado quando executada a varredura pelo vetor de ramos *Node*. A varredura retornou um ramo adequado para o par de índices (3-6) (Figura 66 - imagem da esquerda), fazendo com que assim o elemento de índice “6” de *vaux* seja inserido como nó filho do nó de índice “3” em avaliação da árvore. Em sequência o elemento de índice “6” é removido de *vaux* e o procedimento aciona uma nova busca para o nó filho da árvore de índice “6” (Figura 66 - imagem da direita).

Figura 65 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 5.



Fonte: Próprio autor.

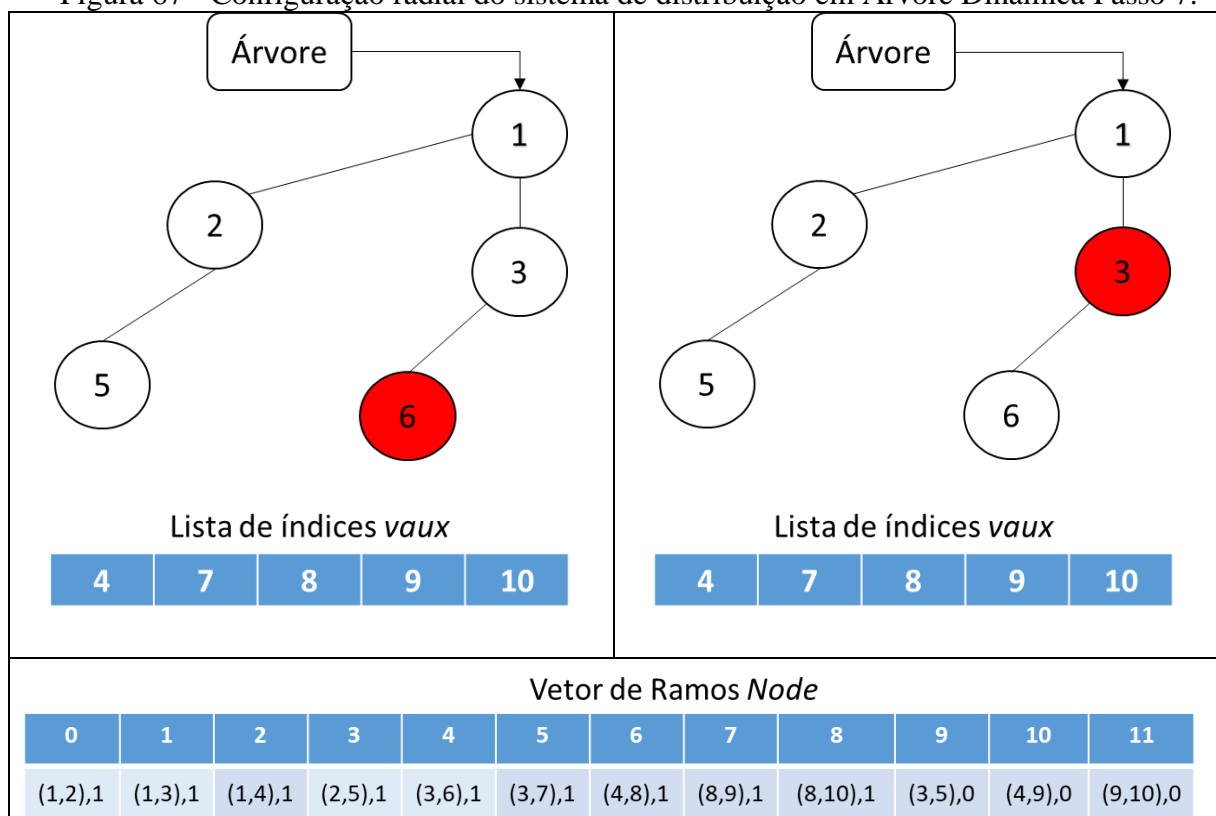
Figura 66 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 6.



Fonte: Próprio autor.

Partindo-se do nó da árvore em avaliação com índice “6” na Figura 67, após todas as configurações entre os pares de índices formados pelo índice do nó da árvore avaliado de índice “6” e os índices de *vaux* (6-4; 6-7; 6-8;6-9; 6-10) nenhum ramo de ligação no vetor de ramos *Node* possui tais combinações de índices com a característica de chave de circuito fechada (Figura 67 – imagem da esquerda). Assim, o procedimento retorna a quem o acionou anteriormente, o nó de índice “3” (Figura 67 – imagem da direita) e se reinicia uma nova busca a partir deste nó.

Figura 67 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 7.



Fonte: Próprio autor.

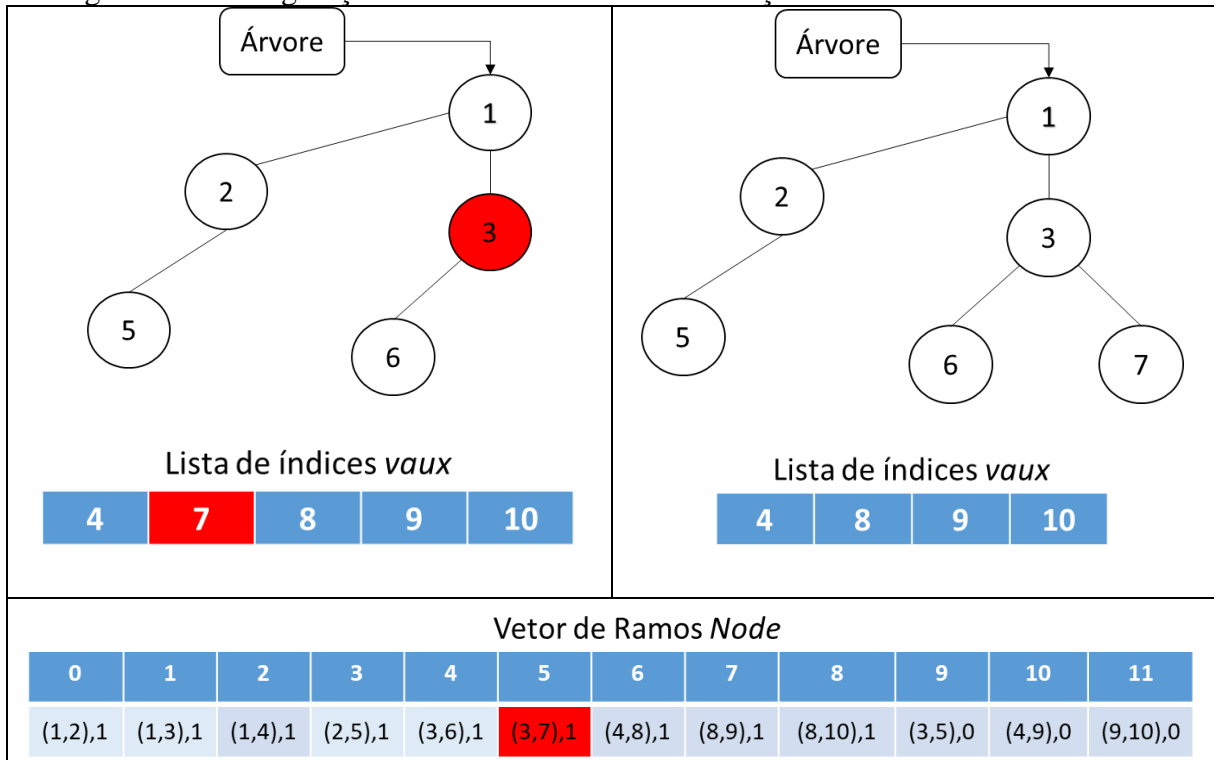
Na Figura 68 uma nova busca é realizada para o nó da árvore dinâmica de índice “3” na tentativa de se encontrar outros ramos de ligação que possuam como alguns de seus índices o nó de índice “3” avaliado na árvore.

A varredura do vetor de ramos *Node* para o par de índices (3-4) formados pelo nó da árvore e o(s) elemento(s) de *vaux* não retornou nenhum ramo que respeitasse as características exigidas. O próximo índice de *vaux* fez o par de índices (3-7) cuja varredura deste ramo

com característica de chave fechada retornou um elemento qualificado (Figura 68 - imagem da esquerda).

O índice “7” de *vaux* é inserido como um novo filho do nó de índice “3” da árvore, se remove o elemento de índice de barra “7” de *vaux* (Figura 68 – imagem da direita), e se reinicia o processo de busca a partir do nó de índice “7” da árvore dinâmica.

Figura 68 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 8.



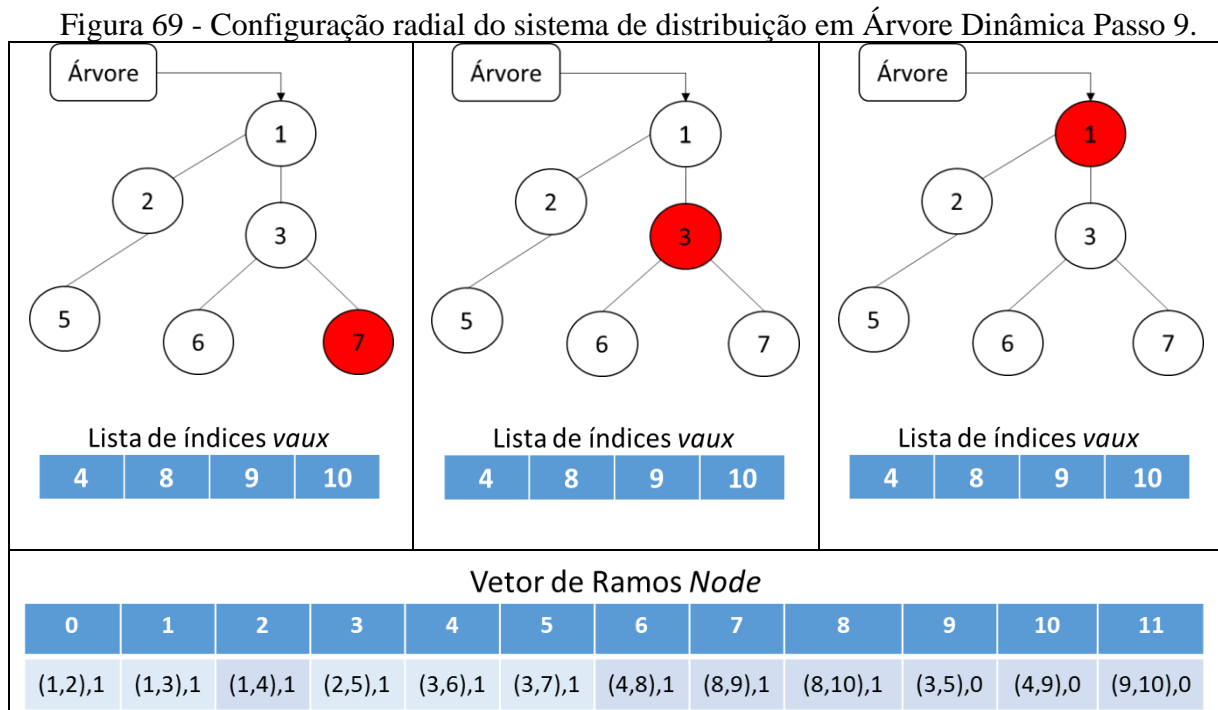
Fonte: Próprio autor.

Assim como no estágio da Figura 64, na Figura 69 a varredura do vetor de ramos *Node* pelo par de índices formado entre o índice “7” do nó avaliado na árvore dinâmica e os índices de barra contidos em *vaux* (7-4; 7-8; 7-9; 7-10) não retornou nenhum ramo qualificado com chave fechada (Figura 69 - imagem da esquerda).

O procedimento então retorna ao momento onde foi acionado, ou seja, o nó da árvore de índice “3” (Figura 69 - imagem do meio). Uma nova varredura é realizada no vetor de ramos no intuito de se encontrar um ramo de ligação que possua como o par de índices os índices encontrados no nó da árvore dinâmica de índice “3” e os índices de barra contidos em *vaux* (3-4; 3-8; 3-9; 3-10). Mais uma vez nenhum ramo que possua tais características de

índices de barra e que possua uma chave fechada foi encontrado (Figura 69 - imagem do meio).

O procedimento novamente retorna ao momento onde foi acionado, ou seja, o nó da árvore de índice “1” (Figura 69 - imagem da direita) na procura de mais elementos a serem inseridos.

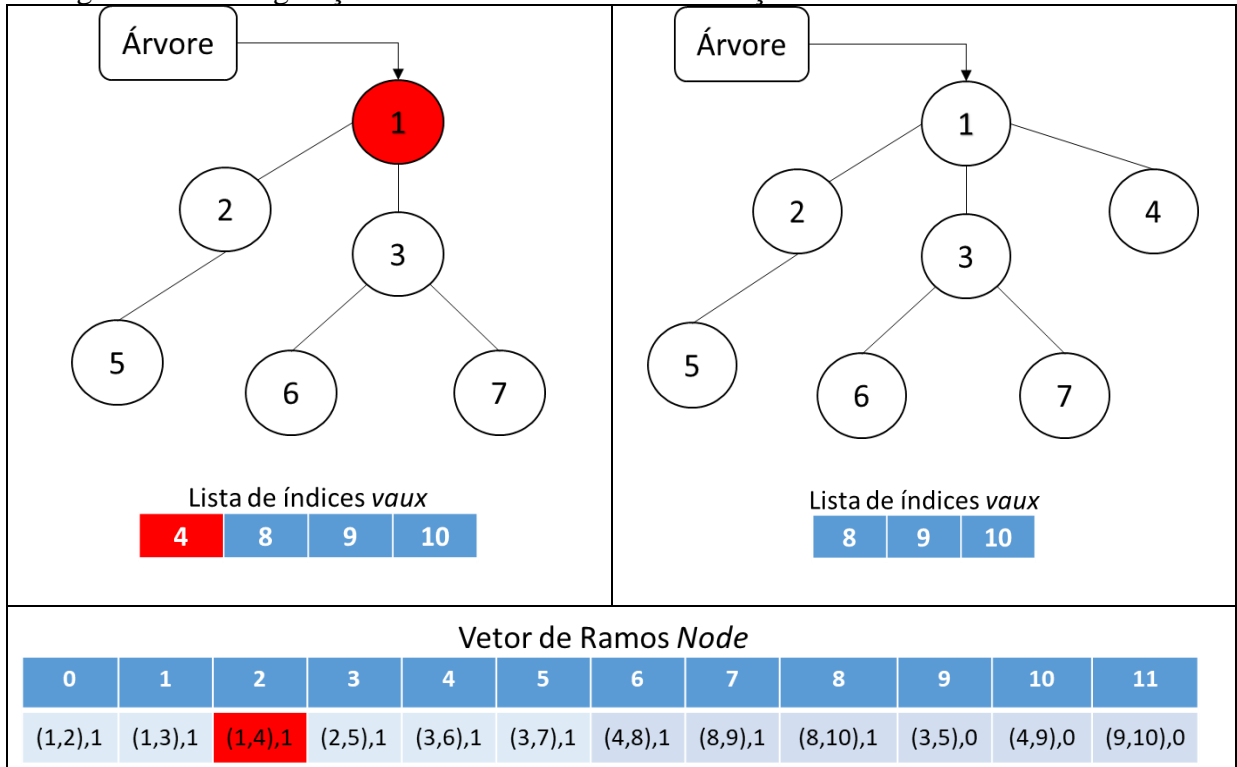


Fonte: Próprio autor.

Na Figura70 partindo-se do nó da árvore em avaliação com índice “1”, uma nova varredura por mais ramos de ligação qualificados dá-se início. A varredura do vetor de ramos *Node* para o par de índices formado pelo índice do nó da árvore com índice “1” e o primeiro elemento de *vaux* o elemento de barra com índice “4” (1-4) já retornou um ramo qualificado (Figura 49 - imagem da esquerda).

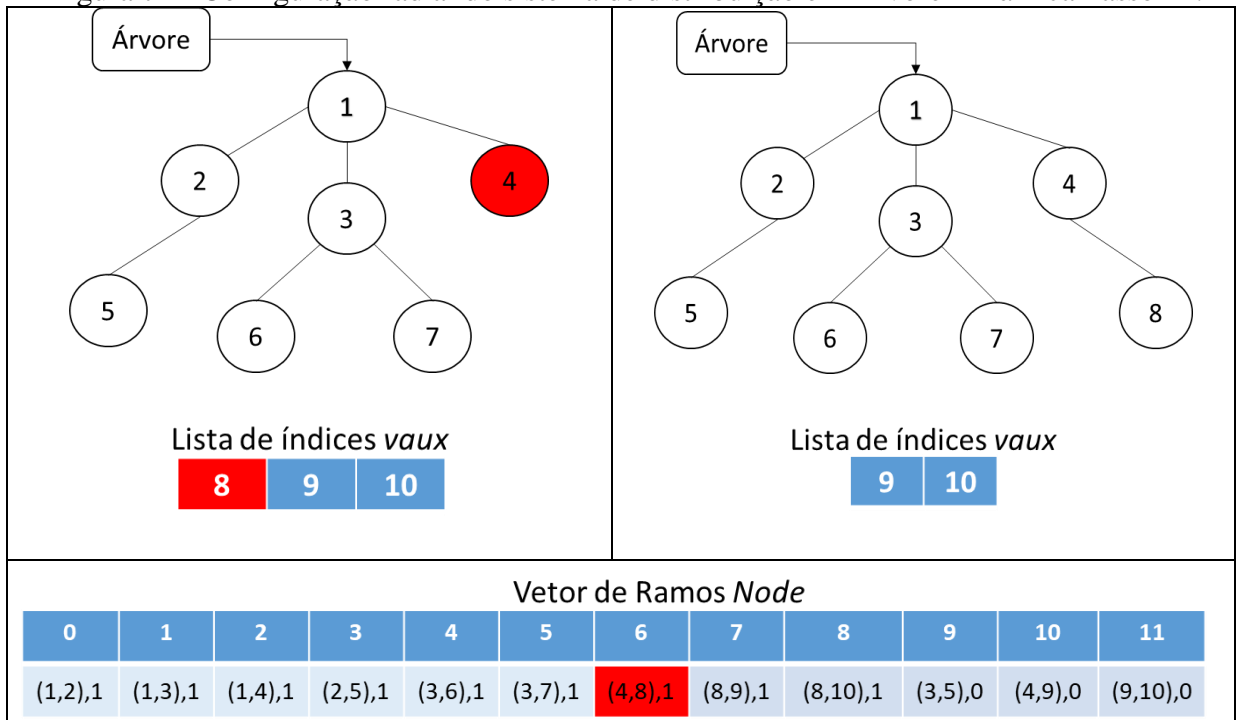
Sendo assim, o índice de barra “4” é inserido como um novo filho do nó da árvore dinâmica de índice “1”, e então removido seu índice de *vaux* (Figura 70 - imagem da direita). O procedimento então continua a busca a partir deste novo nó inserido, de índice “4”. Na Figura 71 partindo-se do nó da árvore em avaliação com índice “4”, uma nova varredura por mais ramos de ligação qualificados dá-se início.

Figura 70 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 10.



Fonte: Próprio autor.

Figura 71 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 11.



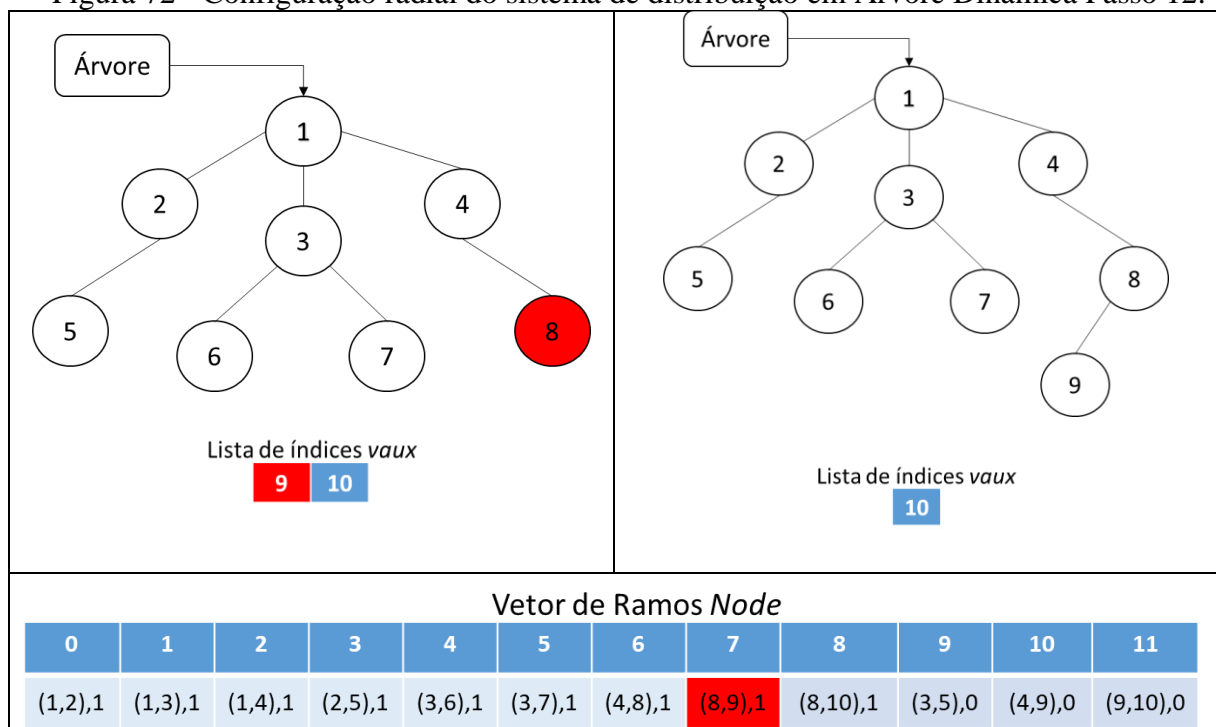
Fonte: Próprio autor.

A varredura do vetor de ramos *Node* para o par de índices formado pelo índice do nó da árvore com índice “4”, e o primeiro elemento de *vaux* o elemento de barra com índice “8” (4-8) já retornou um ramo qualificado (Figura 71 - imagem da esquerda). Sendo assim, o índice de barra “8” é inserido como um novo filho do nó da árvore dinâmica de índice “4”, e então removido seu índice de *vaux* (Figura 71 - imagem da direita). O procedimento então continua a busca a partir deste novo nó inserido, de índice “8”.

Continuando, na Figura 72 partindo-se do nó da árvore em avaliação com índice “8”, uma nova varredura por mais ramos de ligação qualificados dá-se início. A varredura do vetor de ramos *Node* para o par de índices formado pelo índice do nó da árvore com índice “8”, e o elemento de *vaux* o elemento de barra com índice “9” (8-9) já retornou um ramo qualificado (Figura 72 - imagem da esquerda).

Sendo assim, o índice de barra “9” é inserido como um novo filho do nó da árvore dinâmica de índice “8”, e então removido seu índice de *vaux* (Figura 72 - imagem da direita). O procedimento então continua a busca a partir deste novo nó inserido, de índice “9”.

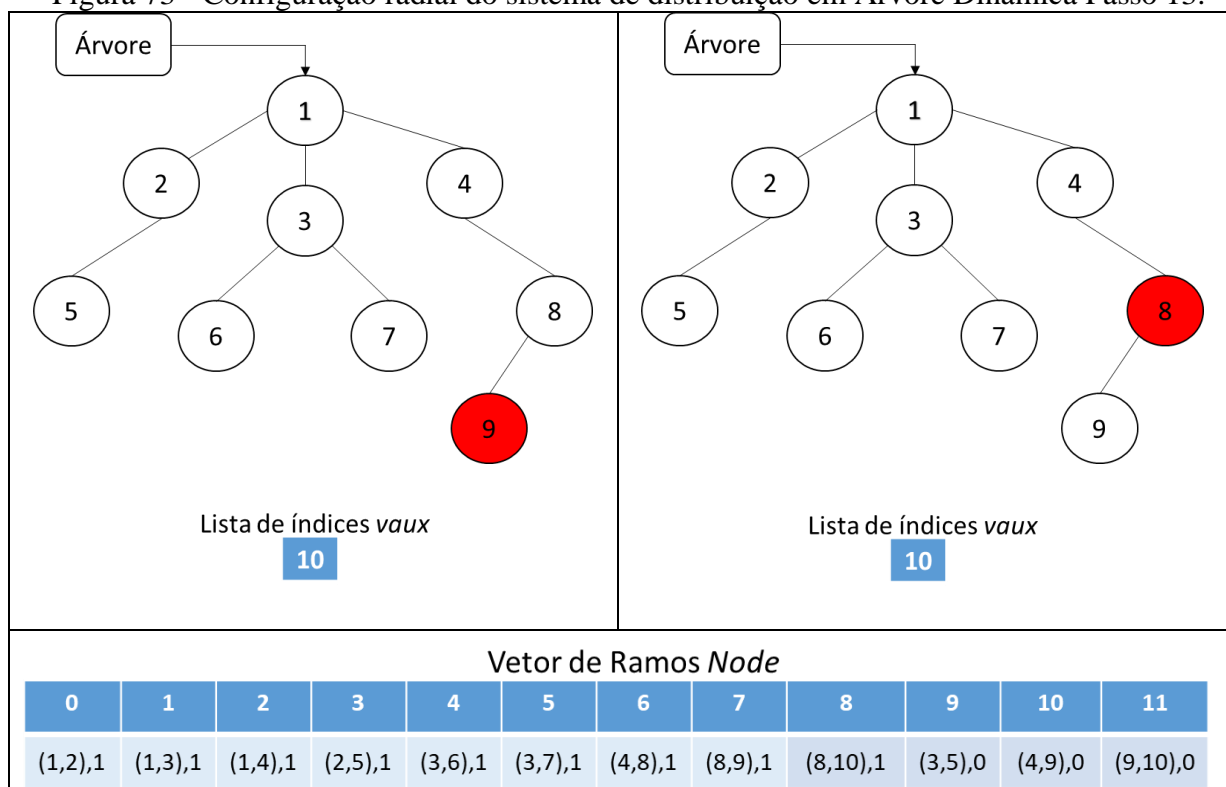
Figura 72 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 12.



Fonte: Próprio autor.

Na Figura 73 partindo-se do nó da árvore em avaliação de índice “9”, busca-se no vetor de ramos *Node* um ramo de ligação com chave fechada que possua como índices de barra o par de índices formados entre a combinação do índice do nó da árvore avaliada “9” e os índices de *vaux*. A varredura do vetor de ramos *Node* não apresentou nenhum ramo qualificado para a comparação de índices (9-10) (Figura 73 - imagem da esquerda), fazendo com que o procedimento retorne para o momento em que foi acionado, o nó da árvore dinâmica de índice “8” (Figura 73 - imagem da direita).

Figura 73 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 13.



Fonte: Próprio autor.

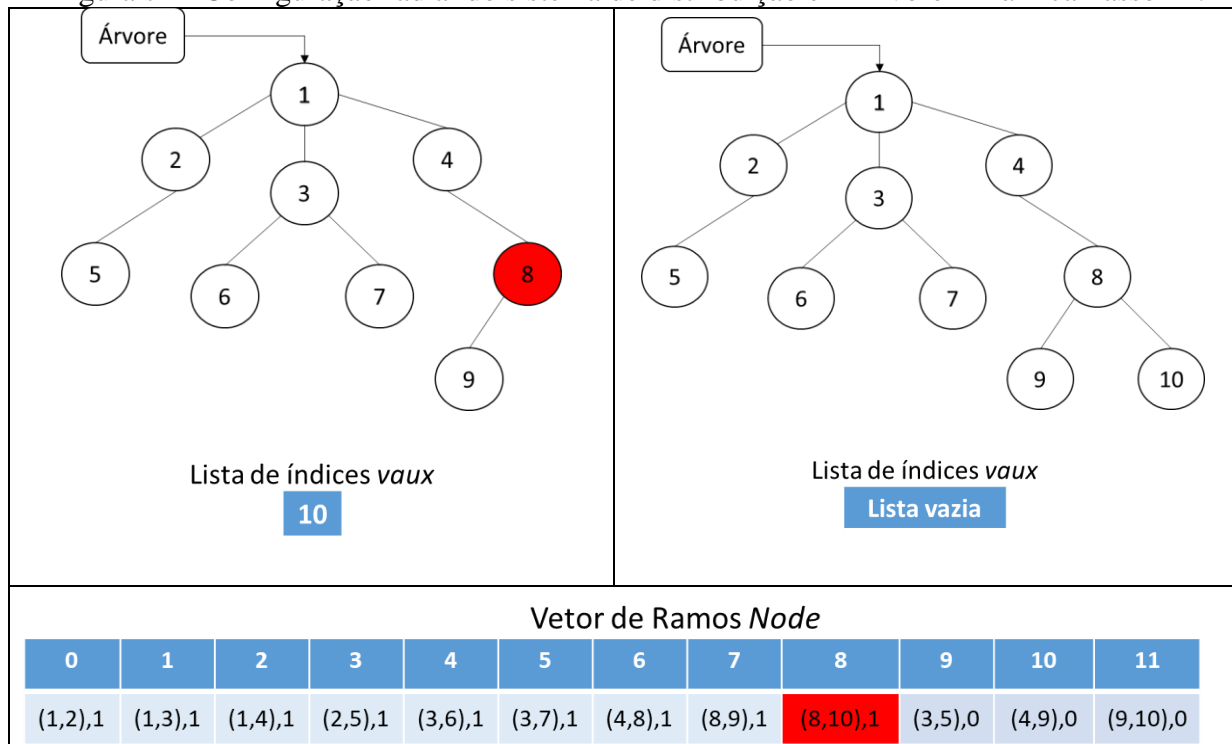
Na Figura 74 partindo-se novamente do nó da árvore em avaliação com índice “8”, uma nova varredura por mais ramos de ligação qualificados dá-se início. A varredura do vetor de ramos *Node* para o par de índices formado pelo índice do nó da árvore com índice “8” e o único elemento restante de *vaux*, o elemento de barra com índice “10” (8-10), já retornou um ramo qualificado (Figura 74 - imagem da esquerda).

Sendo assim, o índice de barra “10” é inserido como um novo filho do nó da árvore dinâmica de índice “8”, e então removido seu índice de *vaux* (Figura 74 - imagem da direita). O procedimento então continua a busca a partir deste novo nó inserido, de índice “10”.

Como *vaux* está vazio o processo pode terminar neste momento e a árvore dinâmica resultante é submetida ao procedimento de cálculo de fluxo de potência. Mas pela técnica da recursividade o procedimento continua as comparações, até que toda o processo recursivo seja concluído.

O nó que está sendo avaliado neste momento na árvore é o nó de índice “10” (Figura 74 – imagem da direita). Como *vaux* está vazio não há com o que se comparar, então a função retorna para quem a chamou, o nó da árvore de índice “8”. A busca então é realizada a parte deste nó de índice “8”, que encontra novamente *vaux* vazio e retorna a função para quem a acionou, o nó da árvore de índice “4”, que encontra novamente *vaux* vazio e retorna a função para quem a acionou, o nó da árvore de índice “1”. A busca mais uma vez é realizada para este nó, que novamente encontra *vaux* vazio e retorna a função para o momento de quem a acionou, que neste caso foi a primeira função (Figura 61). Neste momento todo o processo recursivo acaba e solução construída na árvore dinâmica é repassada para o procedimento de cálculo de fluxo de potência.

Figura 74 - Configuração radial do sistema de distribuição em Árvore Dinâmica Passo 14.



Fonte: Próprio autor.