

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Dissertação de Mestrado

Reconfiguração da Rede de Distribuição
Inteligente usando Reconhecimento Simulado



João Ricardo Barreto de Carvalho
Amorim

João Pessoa-PB, Brasil
Julho - 2022

JOÃO RICARDO BARRETO DE CARVALHO AMORIM

**RECONFIGURAÇÃO DA REDE DE DISTRIBUIÇÃO INTELIGENTE USANDO
RECOZIMENTO SIMULADO**

Exame de Defesa Final apresentado ao Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, da Universidade Federal da Paraíba - UFPB, como requisito final para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Yuri Percy Molina Rodriguez

JOÃO PESSOA

2022

Catálogo na publicação
Seção de Catalogação e Classificação

A524r Amorim, João Ricardo Barreto de Carvalho.

Reconfiguração da rede de distribuição inteligente usando recozimento simulado / João Ricardo Barreto de Carvalho Amorim. - João Pessoa, 2022.

66 f. : il.

Orientação: Yuri Percy Molina Rodriguez.

Dissertação (Mestrado) - UFPB/CEAR.

1. Sistemas de distribuição - Energia elétrica. 2. Recozimento simulado. 3. OpenDSS. 4. Reconfiguração de redes. 5. Minimização de perdas. I. Rodriguez, Yuri Percy Molina. II. Título.

UFPB/BC

CDU 621.311(043)

UNIVERSIDADE FEDERAL DA PARAÍBA – UFPB
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS – CEAR
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA - PPGE

A Comissão Examinadora, abaixo assinada, aprova a Dissertação

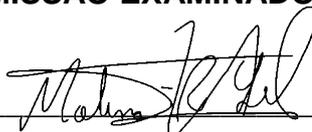
**RECONFIGURAÇÃO DA REDE DE DISTRIBUIÇÃO INTELIGENTE USANDO
RECOZIMENTO SIMULADO**

Elaborada por

JOÃO RICARDO BARRETO DE CARVALHO AMORIM

como requisito parcial para obtenção do grau de
Mestre em Engenharia Elétrica.

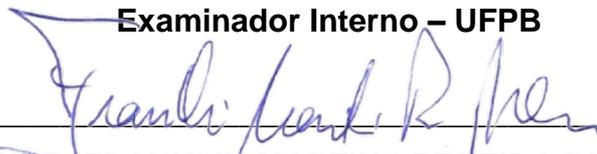
COMISSÃO EXAMINADORA



PROF. DR. YURI PERCY MOLINA RODRIGUEZ
Orientador – UFPB



PROF. DR. HELON DAVID DE MACEDO BRAZ
Examinador Interno – UFPB



PROF. DR. FRANKLIN MARTINS PEREIRA PAMPLONA
Examinador Externo – IFPB

A todos que me ajudaram e ajudam na jornada.

AGRADECIMENTOS

A Deus pelo dom da vida.

A minha mãe por tudo.

A minhas irmãs, minha segunda mãe Valdete, meu sobrinho e afilhado Enrique, por todas as vezes que me fizeram continuar indo em frente.

Por todos os membros da minha família, por me ajudarem a me lembrar de quem eu sou e onde eu devo ir.

Ao grupo da graduação Eletrochoro, que fez eu chegar onde estou.

A minha família na Igreja, Gramados em Cristo, por referência de união e amizade.

A dirigente 2021 e 2022 do ECRI, por me fazerem feliz mesmo sem perceber.

A todo o time de Engenharia residente na Fábrica Jeep Goiana, que têm tanto me ensinado nesses últimos meses.

E em especial ao professor Yuri por todo o apoio e determinação em me ajudar.

“Eu tenho um sonho”
Marthin Luther King.

LISTA DE ILUSTRAÇÕES

FIGURA 1 CARREGAMENTO DA INTERFACE COM DO OPEN DSS COM O VISUAL STUDIO (MENU PROJETO - ADICIONAR REFERÊNCIA)....	13
FIGURA 2 ALGORITMO RECOZIMENTO SIMULADO	24
FIGURA 3 SISTEMA DE 5 BARRAS	29
FIGURA 4 CAMINHOS BARRA-GERAÇÃO	32
FIGURA 5 CAMINHOS DE E PARA	33
FIGURA 6 MALHA GERADA	33
FIGURA 7 SISTEMA DE 16 BARRAS	35
FIGURA 8 APLICAÇÃO DO MÉTODO - SOLUÇÃO INICIAL	35
FIGURA 9 SOLUÇÃO GERADA - CHAVES 8, 11 E 12 ABERTAS - SISTEMA DE 16 BARRAS	37
FIGURA 10 SOLUÇÃO GERADA - CHAVES 7, 8 E 12 ABERTAS - SISTEMA DE 16 BARRAS	38
FIGURA 11 SOLUÇÃO GERADA - CHAVES 7, 8 E 13 ABERTAS - SISTEMA DE 16 BARRAS	39
FIGURA 12 SOLUÇÃO GERADA - CHAVES 6, 7 E 13 ABERTAS - SISTEMA DE 16 BARRAS	40
FIGURA 13 TELA INICIAL SOFTWARE	42
FIGURA 14 CARACTERÍSTICAS DE CARGA.....	43
FIGURA 15 CARACTERÍSTICAS DE LINHA.....	43
FIGURA 16 CARACTERÍSTICAS DAS CHAVES	44
FIGURA 17 PARÂMETROS DISPONÍVEIS PARA AJUSTE.....	44
FIGURA 18 INICIALIZAÇÃO DO ALGORITMO – ÚNICA VEZ	45
FIGURA 19 INICIALIZAÇÃO DO ALGORITMO – REPETIDAS VEZES	45
FIGURA 20 PÁGINA DE RESULTADOS	46
FIGURA 21 SOLUÇÕES PARA 5 BARRAS.....	46
FIGURA 22 SISTEMA DE 33 BARRAS	48
FIGURA 23 SISTEMA DE 69 BARRAS DE SAVIER E DAS.....	51

LISTA DE TABELAS

TABELA 1 VERBOS MAIS UTILIZADOS NO OPENDSS®	14
TABELA 2 PARÂMETROS DO ELEMENTO <i>CIRCUIT</i>	15
TABELA 3 PARÂMETROS DO ELEMENTO <i>LINE</i>	16
TABELA 4 MODELOS DO ELEMENTO <i>LOAD</i>	17
TABELA 5 PARÂMETROS DO ELEMENTO <i>LOAD</i>	18
TABELA 6 MODELOS DO ELEMENTO <i>GENERATOR</i>	19
TABELA 7 PARÂMETROS DO ELEMENTO <i>GENERATOR</i>	19
TABELA 8 PARÂMETROS DO ELEMENTO <i>SWTCONTROL</i>	20
TABELA 9 RESULTADOS OBTIDOS SISTEMA 5 BARRAS	47
TABELA 10 RESULTADOS SISTEMA 33 BARRAS	49
TABELA 11 COMPARAÇÃO COM OUTRAS REFERÊNCIAS - SISTEMA 33 BARRAS	50
TABELA 12 RESULTADOS SISTEMA 69 BARRAS	52
TABELA 13 COMPARAÇÃO COM OUTRAS LITERATURAS - SISTEMA 69 BARRAS	53

RECONFIGURAÇÃO DA REDE DE DISTRIBUIÇÃO INTELIGENTE USANDO RECOZIMENTO SIMULADO

Resumo: Este trabalho tem propõe um método de Reconfiguração Inteligente de Redes de Distribuição de Energia Elétrica usando a meta-heurística de recozimento simulado, se implementaram duas novas classes na linguagem C#, a primeira para caracterizar os parâmetros do algoritmo de Recozimento Simulado, e a segunda para realizar interface entre a metaheurística implementada e acoplamento com o Software OpenDSS, que é um simulador de fluxo de potência em sistemas de distribuição. Para a realização de este trabalho se utilizará o ambiente de desenvolvimento integrado do Visual Studio 2019. O algoritmo foi implementado em uma interface desenvolvida para propiciar a alteração ágil e dinâmica dos parâmetros de simulação. O algoritmo será validado em um sistema de 5 barras e testado em sistemas de 33 barras e 69 barras, bastante utilizados na literatura. Os resultados dos testes indicam que o algoritmo proposto atinge os resultados e soluções pretendidas de maneira eficaz, rapidamente e precisamente.

Palavras-chaves : Recozimento Simulado, Sistemas de Distribuição, OpenDSS, Reconfiguração de Redes, Minimização de Perdas.

DISTRIBUTION SYSTEM SMART RECONFIGURATION UTILIZING SIMULATED ANNEALING

Abstract: This work proposes an Electrical Energy Distribution Smart Reconfiguration method using the Simulated Annealing metaheuristic. It has been implemented two new classes in the C# language, the first to characterize the Simulated Annealing algorithm parameters, and the second one to make the interface between the implemented metaheuristic and the coupling with the OpenDSS software, which is a distribution systems power flow simulator. To this work, will be utilized the integrated development environment of Visual Studio 2019. The algorithm was implemented in an interface developed to provide an agile and dynamic parameters modification. The algorithm will be validated in a 5 bus system and tested in both 33 and 69 buses, well used in the literature. The test results suggest that the proposed algorithm reaches the intended outcomes and solutions in an efficient, fast and precisely.

Keywords: Simulated Annealing, Distribution Systems, OpenDSS, Network Reconfiguration, Loss Minimization

1 INTRODUÇÃO

1 INTRODUÇÃO

O sistema de distribuição é uma importante parte do Sistema Elétrico que abrange, de forma geral, as ligações finais entre a geração e o consumidor. Grande parte dos sistemas de distribuição utilizam um ramo principal com derivações ao longo do caminho, o qual ganha o nome de sistema radial. Esse tipo de sistema é utilizado principalmente por causa do design simples, geralmente com um baixo custo e um esquema de proteção robusto. Essa configuração sugere que todos os componentes entre uma carga e uma geração devem funcionar, apesar de que prováveis desligamentos podem acontecer devido a uma falha em um único componente. A confiabilidade de um sistema de distribuição radial pode ser melhorada pela reconfiguração de rede, possível graças ao fechamento de chaves normalmente abertas e a abertura de chaves normalmente fechadas [1].

A definição de redes de distribuição reconfiguráveis e flexíveis são interpretadas em dois aspectos. Quando a rede opera normalmente, ela se refere a reduzir perdas, balancear as suas carga e melhorar o seu estado de operação. Na situação de ocorrência de curtos ou desligamento, a reconfiguração remete a achar possíveis configurações para garantir o abastecimento do maior número de cargas possíveis. A solução ótima para a reconfiguração será aquela que fornece o menor nível de perdas [2].

Ao alterar a topologia das redes, podem ser observadas uma série de consequências: (i) mudança no balanceamento do carregamento dos alimentadores, (ii) redução das perdas de energia, (iii) melhoria nos níveis de tensão e (iv) melhoria na confiabilidade. A conexão de fontes de geração distribuída (GD), como a eólica e a fotovoltaica, além do advento de carros elétricos, vem alterando a estrutura das redes de distribuição. O seu incremento tem fortes influências no fluxo de energia e no perfil de tensão nas redes, fazendo com que a importância de reconfigurações de rede aumente cada vez mais, exigindo uma maior velocidade em se adaptar às cargas e GDs (que possuem um comportamento dinâmico ao longo do dia) [3,4,5,6,7]. Tal problema para encontrar a solução ótima é considerado de otimização combinatória. Tal solução não é facilmente encontrada, pois existem inúmeras possibilidades de configuração em uma rede de distribuição, as quais aumentam conforme a dimensão

da rede evolui. Considerando que cada chave possui dois estados, o número de configurações será de 2^n , onde n é a quantidade de chaves no sistema [8,9].

Atualmente, vários métodos são utilizados para resolver o problema de otimização de reconfiguração de rede. Este trabalho aplica uma meta-heurística para resolver o problema, por meio de um algoritmo chamado Recozimento Simulado ou *Simulated Annealing*, com auxílio de uma Interface Gráfica desenvolvida em linguagem C# no ambiente de desenvolvimento Visual Studio, para melhorar a interação do usuário com o algoritmo.

1.1 OBJETIVO GERAL

Apresentar o desenvolvimento de um software com interface amigável capaz de resolver o problema de reconfiguração de redes de distribuição minimizando as perdas ativas.

1.2 OBJETIVOS ESPECÍFICOS

1. Implementação de algoritmo de recozimento simulado, com o desenvolvimento de classes em C#.
2. Implementação de melhorias na capacidade de encontrar solução ótima no algoritmo proposto.
3. Modelar redes no ambiente do OpenDSS.
4. Implementar interface gráfica no Visual Studio com interação com algoritmo de recozimento simulado que seja capaz de mostrar características dos sistemas, solução inicial, iterações do algoritmo e solução ótima encontrada.
5. Testar e comparar o algoritmo proposto com outras literaturas.

1.3 ORGANIZAÇÃO DO TRABALHO

O trabalho foi organizado de forma que em cada capítulo são apresentados os seguintes tópicos:

- Capítulo 2: É apresentado um breve histórico sobre o tema de Recozimento Simulado.
- Capítulo 3: Breve explanação sobre o software OpenDSS, incluindo seus membros e forma de utilização.
- Capítulo 4: Abordada a fundamentação teórica sobre o algoritmo utilizado.
- Capítulo 5: As melhorias implementadas no algoritmo são apresentadas.
- Capítulo 6: Os resultados alcançados com este trabalho são mostrados e analisados, além de serem comparados com algumas literaturas.
- Capítulo 7: São apresentadas as conclusões, melhorias deste trabalho e sugestões para trabalhos futuros.

2 REVISÃO DE LITERATURA

2 REVISÃO DE LITERATURA

A revisão bibliográfica sobre o problema de reconfiguração de redes é apresentada neste capítulo em forma cronológica, focando mais recentemente na utilização do recozimento simulado para a resolução deste problema.

Em 1975, [10] propôs em seu trabalho um método de resolução em que partindo-se de uma rede malhada com todas as chaves fechadas, eram então abertas sequencialmente guiadas pela minimização de perdas ativas. Entre suas dificuldades estavam o alto tempo de processamento e a utilização exclusiva da componente real da corrente.

Após 13 anos, em [11], parte-se de uma solução inicial radial em um sistema com mais de dois alimentadores. Esse trabalho estuda a reconfiguração da rede com transferências de cargas entre alimentadores para diminuir as perdas, através de um método denominado “Troca de Ramos”. Para diminuir o número de possibilidades no espaço de busca, foi utilizado um método baseado nas tensões dos terminais da chave a ser fechada para garantir redução de perdas. O trabalho é considerado de grande importância dentro do ambiente estudado.

Logo depois, em 1989, [12] propôs um método heurístico robusto para a reconfiguração de redes de distribuição, que pode ser utilizado tanto em projetos como em operação. Este trabalho utiliza como ponto de partida uma rede malhada com todas as chaves fechadas. Como forma de procura pela solução com menor perdas, fontes de correntes fictícias são modeladas e aquela que possui, ao se calcular o fluxo de potência, o menor valor de corrente, indica a chave a ser aberta. Como limitações deste trabalho, a sua aplicação em uma suposta operação de redes deve ter um reajuste no número limite de operações de troca das chaves.

No mesmo ano, [13] propôs uma continuação do trabalho de [11]. Nele, foi proposta uma melhoria no método de Troca de Ramos, utilizando cálculos de fluxo de carga específicos de redes radiais, otimizando a busca da melhor solução visando a minimização das perdas e equilíbrio das cargas. Aqui se começou a visualizar os problemas de reconfiguração como sendo de natureza combinatória.

Em 1995, [14] desenvolveram um método combinando troca de ramos e abertura sequencial, incluindo melhoria para os dois métodos. Ele é utilizado para

descobrir a melhor configuração de chaves e também otimiza a alocação de capacitores no sistema. Foi aplicado em um sistema com cinco alimentadores de 20 kV, com 63 nós e 80 barras.

Em [15] é afirmado que a partir dos anos 2000 é observada uma predominância de métodos baseados em modelos físicos ou biológicos para a solução de problemas de reconfiguração, destacando-se a utilização de técnicas meta-heurísticas.

Em 1999, são discutidos em [16] os métodos Busca Tabu, Recozimento Simulado e um Algoritmo Genético, com algumas adaptações, são comparados em suas aplicações ao problema de reconfiguração de redes.

Em [17] utilizou-se a técnica de Recozimento Simulado em conjunto com a Busca Tabu para a otimização de sistemas reais. Segundo eles, o recozimento simulado teve êxito em evitar mínimos locais, mas com alto custo computacional, enquanto que a busca tabu encontrou por meio de um algoritmo de descida a solução, mas não garantiu a convergência.

Em [18] é utilizado o Recozimento Simulado para a resolução do problema de reconfiguração de redes de distribuição levando em consideração as restrições de carga na função objetivo de otimização.

No mesmo ano, [19] utilizou o recozimento simulado para aprimorar a confiabilidade da rede, minimizando o custo causado pela interrupção de fornecimento ao cliente, observando as limitações operacionais do sistema. A técnica de recozimento simulado foi utilizada em conjunto com a análise de confiabilidade.

Em [20] é utilizado o algoritmo de Recozimento Simulado Imune (SAI) para o problema de reconfiguração de redes de distribuição. Neste trabalho, chaves compartilhadas entre malhas são utilizadas para geração de soluções possíveis, além de utilizar resultados históricos e atuais para otimizar a busca.

Em [21], o OpenDSS é utilizado em conjunto com o recozimento simulado para analisar os impactos da injeção de Geração Distribuída em redes de distribuição tanto na operação normal com em situações de pós-curto com diferentes níveis de carga. Segundo os autores, a integração de Geração Distribuída aumenta as soluções possíveis e diminui as perdas na solução ótima.

Em [21] é discutido o uso do Recozimento Simulado aplicado ao algoritmo de Colônia de Formigas para realizar a otimização do problema de reconfiguração de

redes de distribuição. O método foi chamado de Algoritmo Híbrido Melhorado de Colônia de Formigas e foi eficiente na busca pela solução ótima.

Foi introduzido em [22] um novo algoritmo de resolução do problema de reconfiguração utilizando uma abordagem de aproximação sequencial, incluindo modificações próximas e aleatórias na rede no método de recozimento simulado.

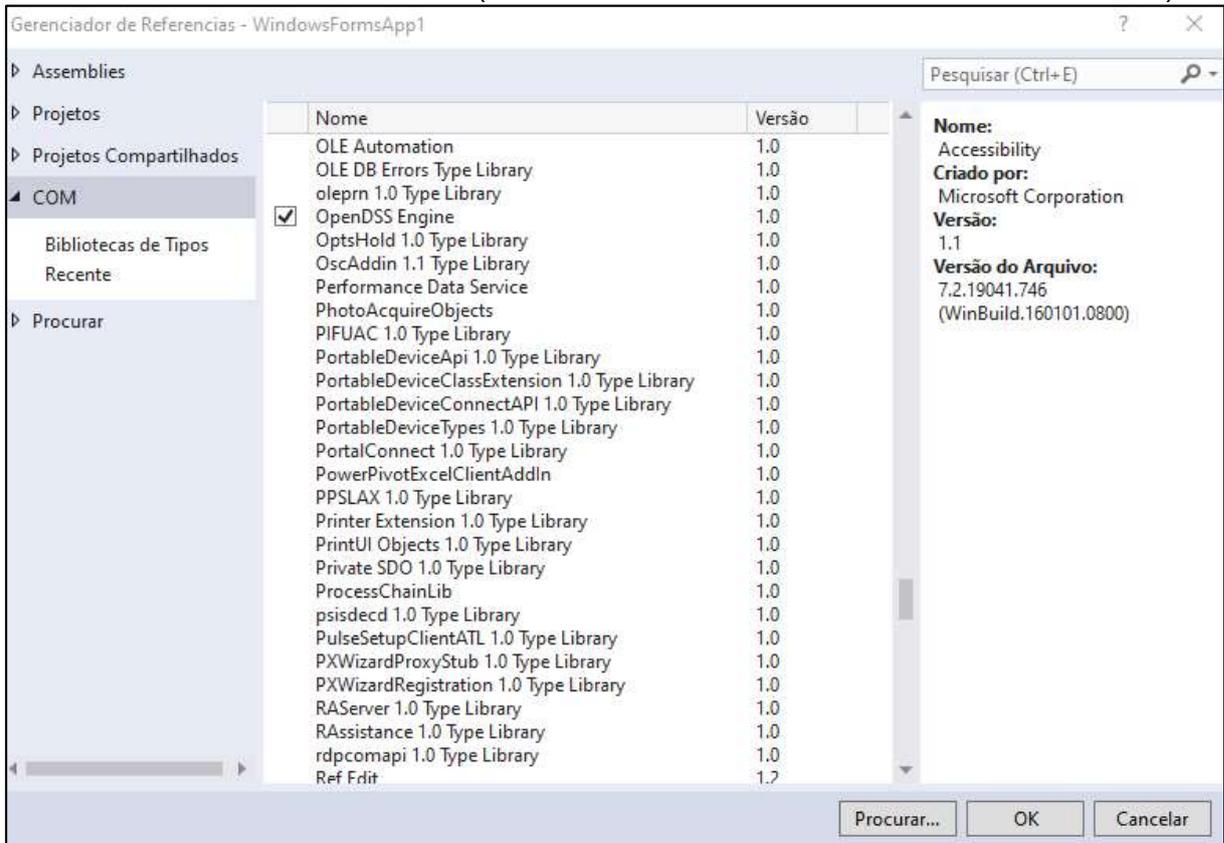
Foi aplicado em [23] o método de recozimento simulado para a programação ótima de configuração de uma rede de distribuição ao longo de um dia, levando em consideração diversas variáveis de otimização.

Como visto, o Recozimento Simulado tem amplo uso no escopo de otimização aplicada a reconfiguração de redes de distribuição. Mesmo assim, diversas melhorias no método ainda podem ser feitas. Este trabalho propõe uma nova abordagem na geração de vizinhos garantindo a radialidade da rede e um espaço de busca maior.

3 SOFTWARE OPENDSS

No Brasil, o OpenDSS® foi adotado pela Agência Nacional de Energia Elétrica para a padronização do cálculo do fluxo de potência das distribuidoras, e consequentemente mensurar o nível de perdas de forma mais padronizada [25, 16, 26]. O OpenDSS® é baseado em linhas de comando que definem os circuitos e podem ser originárias do usuário, de arquivo de texto ou a partir de interação com programas externos por meio de uma interface chamada *Component Object Model* (COM), como pode-se observar na Figura 1 [27, 26].

FIGURA 1 CARREGAMENTO DA INTERFACE COM DO OPEN DSS COM O VISUAL STUDIO (MENU PROJETO - ADICIONAR REFERÊNCIA)



Fonte: Autor, 2022.

```
private DSS dssOBJ = new DSS(); (exemplo de definição de um
objeto do OpenDSS em C# utilizando a interface COM
carregada no Visual Studio)
```

3.1 APLICAÇÃO OPENDSS®

O uso do OpenDSS® é amplamente difundido. Sua aplicação junto à otimização de reconfiguração de rede geralmente é feita aliada a alguma programação externa (Matlab, Python, C#, VBA, entre outros). Em [27], utiliza-se o OpenDSS® juntamente com o Matlab® para desenvolvimento de algoritmo de otimização chamado de Evolutionary Particle Swarm Optimization para alocação ótima de banco de capacitores. Em [28], são comparados três opções de otimização para banco de capacitores também utilizando o OpenDSS® e o Matlab®.

Em [2], o OpenDSS® foi abordado em conjunto com o Matlab® para analisar os impactos da implementação de geração distribuída em uma rede de distribuição utilizando um algoritmo de recozimento simulado. Além de diversas outras utilizações do OpenDSS® em análises e estudos que requerem o cálculo de fluxo de potência em uma rede de distribuição. O uso de outras linguagens de programação que não sejam em Matlab® pode ser visto em [29], onde o autor desenvolve uma ferramenta *System Control Interface* (SCI) em Python para trabalhar em conjunto com o OpenDSS®, visando auxiliar novos estudos, e também em [15], onde o autor utilizou a linguagem C# para implementar um novo algoritmo do tipo *Improved Binary particle Swarm Optimization* (IBPSO) e compará-lo aos resultados de outros métodos propostos na literatura.

3.2 LINGUAGEM OPENDSS®

O OpenDSS® utiliza comandos definidos por linhas de texto e cada comando é formado, em geral, por um verbo e alguns parâmetros [26]:

Verbo TipoElemento.Nome Param1 = Valor1 Param2=Valor2

Os verbos mais utilizados são mostrados conforme Tabela 1.

TABELA 1 Verbos mais utilizados no OpenDss®

Verbo	Descrição
New	Cria um novo elemento.
Edit	Edita um elemento.
Set	Define as opções de solução.
Solve	Soluciona o circuito definido.
Show	Apresenta relatórios em .txt
Export	Salva relatórios em .csv
Plot	Plota resultados do fluxo de potência

Fonte: Pegado, 2019

3.3 ELEMENTOS BÁSICOS

Nesta sessão serão apresentados os elementos pertencentes ao OpenDSS®. Os elementos utilizados no trabalho serão postos em negrito neste capítulo, enquanto que os disponíveis para uso mas que não foram utilizados estarão em formatação padrão.

3.3.1 CIRCUITO

O elemento Circuito (*Circuit* no OpenDSS) é um equivalente Thévenin da rede de transmissão vista pela subestação ou o equivalente Thévenin da subestação vista por um alimentador.

Todo circuito no OpenDSS deve ser inicializado pelo elemento *Circuit* [30]. São demonstrados Tabela 2 os parâmetros do elemento *Circuit*.

TABELA 2 Parâmetros do elemento *Circuit*

Parâmetro	Descrição
basekV	Tensão nominal de linha (em kV)
bus1	Nome da barra a qual o terminal do elemento é conectado
R1	Resistência de sequência positiva da fonte (em ohms)
X1	Reatância de sequência positiva da fonte (em ohms)
mvasc3	Potência de curto-circuito trifásico (em MVA)
mvasc1	Potência de curto-circuito monofásico (em MVA)
pu	Valor da tensão na barra (em por unidade)

Fonte: Radatz, 2015, Adaptado

Um exemplo de definição do elemento *Circuit* no OpenDSS é apresentado a seguir:

```
New Circuit.SubestaçãoA bus1=Barra1 basekv=13.8 pu=1
~mvasc3=2000.0 mvasc1=2000.0
```

3.3.2 Linha de Transmissão

No OpenDSS, as linhas são modeladas conforme modelo π , com capacitores *shunt*. Seu comando é o *Line* e os parâmetros são dispostos na Tabela 3.

TABELA 3 Parâmetros do elemento *Line*

Parâmetro	Descrição
Bus1	Nome da barra em qual o terminal 1 é conectado
Bus2	Nome da barra em qual o terminal 2 é conectado
phases	Número de fases
baseFreq	Frequência de base das reatâncias (em Hz)
R1	Resistência de sequência positiva por unidade de distância (em ohms)
R0	Resistência de sequência zero por unidade de distância (em ohms)
X1	Reatância de sequência positiva por unidade de distância (em ohms)
X0	Reatância de sequência zero por unidade de distância (em ohms)
C1	Capacitância total de sequência positiva por unidade de distância (em nF)
C0	Capacitância total de sequência zero por unidade de distância (em nF)
Rmatrix	Matriz de resistências da linha por unidade de distância
Xmatrix	Matriz de reatâncias da linha por unidade de distância
Units	Unidade de distância utilizada
Normamps	Corrente nominal da linha
length	Comprimento da linha

Fonte: Pegado, 2019, Adaptado

A seguir é apresentado um exemplo de definição de Linha no OpenDSS.

```
New Line.Linha1 phases=3 bus1=Barra1 bus2=Barra2
~basefreq=60 units=km length=1.0 normaps=400.0 R1=0.015
~X1=0.1 R0=10.0 X0=30.5 C1=800.0 C0=270.0
```

3.3.3 Cargas

No OpenDSS, as cargas são descritas pelo elemento *Load*. Este pode ser descrito por sua potência ativa nominal, sua potência reativa e seu fator de potência, além de poder ser descrito por uma curva de carga. É possível descrever a carga apenas com uma das seguintes combinações:

- Potência ativa e fator de potência;
- Potência ativa e potência reativa;
- Potência aparente e fator de potência;

Além disso, o OpenDSS fornece vários tipos de carga, conforme Tabela 4.

TABELA 4 Modelos do elemento *Load*

Modelo	Descrição
1	Potência ativa e reativa constantes
2	Impedância constante
3	Potência ativa constante e potência reativa quadrática
4	Potência ativa não linear e potência reativa quadrática
5	Corrente constante
6	Potência ativa constante, podendo ser modificada pelo formato de curva de carga, mas potência reativa fixa em seu valor nominal
7	Similar ao modelo 6, mas com a potência reativa variando com o quadrado da tensão
8	Modelo especial ZIP

Fonte: Radatz, 2015. Adaptado

Os parâmetros do elemento *Load* podem ser observados na Tabela 5.

TABELA 5 Parâmetros do elemento *Load*

Parâmetros	Descrição
Bus1	Nome da barra a qual a carga é conectada
Phases	Número de fases
Conn	Tipo de conexão
Model	Modelo da carga
kV	Tensão nominal
kVA	Potência aparente nominal
kW	Potência ativa nominal
kvar	Potência reativa nominal
pf	Fator de potência
daily	Nome da curva de carga associada à carga

Fonte: Pegado, 2019. Adaptado

Um exemplo definindo um elemento *Load* é apresentado a seguir:

```
New Load.Carga phases=3 model=5 bus1=Barra1 kV=0.380
~kw=20 kvar=16 conn=delta
```

3.3.4 Gerador

Este elemento é, assim como a carga, de conversão de energia. Para caracterizá-lo, é necessário fornecer sua potência nominal, seu fator de potência, sua

tensão e o modelo utilizado. Também como a carga, para o gerador são disponibilizados diversos modelos, conforme Tabela 6:

TABELA 6 Modelos do elemento *Generator*

Modelo	Descrição
1	Potência constante
2	Impedância constante
3	Modelo de gerador como barra PV
4	Potência ativa constante, podendo seguir uma curva de despacho e potência reativa fixa
5	Potência ativa constante e reatância com valor fixo. P segue a curva de despacho e Q calculada para o valor da reatância fixada
6	Modelo definido pelo usuário
7	P constante limitado pela corrente

Fonte: Radatz, 2015. Adaptado

Os parâmetros do elemento *Generator* estão explicitados na Tabela 7:

TABELA 7 Parâmetros do elemento *Generator*

Parâmetros	Descrição
Bus1	Nome da barra a qual o gerador está conectado
Phases	Número de fases
Conn	Tipo de conexão
Model	Modelo do gerador
kV	Tensão nominal
kW	Potência ativa nominal
kvar	Potência reativa nominal
pf	Fator de potência
maxkvar	Valor máximo de potência reativa em kvar (modelo 3)
minkvar	Valor mínimo de potência reativa em kvar (modelo3)

Fonte: Pegado, 2019. Adaptado

Um exemplo de Gerador modelado no OpenDSS é descrito a seguir:

```
New Generator.Gerador bus1=Barra1 kV=13.8 kW=10000 model=3
~maxkvar=50000 minkvar=1000
```

3.3.5 Chave

Uma chave pode ser representado por um sistema binário: Aberta ou fechada; No sistema de distribuição são chamadas de chaves de seccionamento ou chaves fusível. No OpenDSS, podem ser representadas e alteradas conforme os itens a seguir:

- Criando uma linha curta de impedância desprezível;
- A partir do parâmetro Switch do elemento Line;
- A partir do elemento SwtControl.

O modelo utilizado neste trabalho é o SwtControl. Neste, é possível definir o tempo de operação, estado normal de operação, o elemento de circuito em que a chave opera, entre outros. São apresentados na Tabela 8 os parâmetros deste elemento:

TABELA 8 Parâmetros do elemento *SwtControl*

Parâmetros	Descrição
Action	Realiza a comutação da chave
Delay	Tempo de atraso para atuação da chave
Normal	Estado normal da chave
SwitchedObj	Elemento em que se encontra a chave
SwitchedTerm	Número do terminal do interruptor do elemento controlado

Fonte: Pegado, 2019. Adaptado

Um exemplo de definição de uma chave, assim como realizar a alteração de estado no OpenDSS são mostrados a seguir:

```
New SwtControl.Chave1 SwitchedObj=Line.Linha1
```

```
Edit SwtControl.Chave1 Action=Open
```

4 FUNDAMENTAÇÃO TEÓRICA

4 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a descrição do algoritmo de Recozimento Simulado assim como a sua aplicação ao problema de reconfiguração de redes de distribuição.

O Recozimento simulado tem uma larga variedade de aplicações na área de sistemas de potência, como alocação de capacitores e fluxo de potência ótimo. O recozimento simulado imita o comportamento dos átomos no recozimento de metais [19,33].

É um algoritmo meta-heurístico bastante utilizado para problemas combinatórios, e foi proposto neste nicho em [31]. O conceito de Recozimento Simulado é baseado no processo físico de recozimento, que é utilizado para obter a melhor organização das partículas em um metal, ou seja, obter o estado com o menor nível de energia de um sólido. O processo basicamente compreende dois passos:

1. Aumentar a temperatura do processo a um valor máximo no qual o sólido derrete.
2. Cuidadosamente diminuir a temperatura até que as partículas se organizem no estado sólido.

Enquanto o processo se desenvolve, as partículas dos sólidos se organizam aleatoriamente de acordo com a temperatura. Se o resfriamento é devagar o suficiente de forma que o sólido consiga atingir o nível de equilíbrio térmico, o sólido cristalino de baixa energia seria formado quando o sistema congelasse (temperatura tendendo a zero). Entretanto, se o recozimento é muito rápido, o sólido pode vir a se tornar vidro com uma estrutura não cristalina ou um cristal defeituoso com estruturas amorfas [17].

Neste método de otimização, as seguintes equivalências entre o sistema físico e o problema de otimização são utilizadas [35,16]:

- Estados (organizações de partículas) de um sistema físico são equivalentes a soluções do problema de otimização.
- A energia do sistema físico é equivalente a função objetivo do problema de otimização.

Uma das particularidades úteis deste método é que ele permite escapar de mínimos locais através dos Critérios de Metropolis [33]. Dado um problema do tipo:

$$\begin{aligned} & \text{minimize}_x f(x) \\ & \text{s. a. } x \in X. \end{aligned}$$

Onde x é uma variável que atende a restrições específicas, $f(x)$ é a função objetivo e X é o espaço de busca limitado pelas restrições propostas pelo problema.

Resolve-se esta equação utilizando Recozimento Simulado partindo de uma solução viável inicial x e uma temperatura inicial T_0 . Aplica-se então uma perturbação na solução inicial, verifica-se sua viabilidade e observa-se se ela atende aos critérios de Metropolis ou não. Atendendo aos critérios de Metropolis, a solução é salva como a atual e o número de sucessos é incrementado. A seguir são exemplificados os critérios de Metropolis:

- Dada um estado inicial com energia E_i e um estado subsequente com energia E_{i+1} , o estado subsequente será aceito se $\Delta E = E_{i+1} - E_i < 0$;
- Caso o critério de energia não seja atendido, o estado subsequente pode ser aceito desde que dada uma probabilidade $p_x = e^{-\Delta E/T} > rand$, onde T é a temperatura atual do processo e $rand$ é um número aleatório entre 0 e 1.

Se o número de perturbações ou de sucessos não atingiu o número máximo em uma iteração, o processo se repete com a solução mais nova encontrada. Caso contrário, a iteração é incrementada e a temperatura é atualizada para um valor menor. O processo se repete até que a condição de finalização seja atingida.

Existem várias opções de definições das mais diversas características deste método: Temperatura Inicial, decaimento de temperatura, método de perturbação, condição de finalização, entre outros. O recozimento simulado foi utilizado em diversos trabalhos de reconfiguração de redes, entre eles [17, 35, 16].

A Figura 2 mostra o pseudocódigo do Recozimento Simulado utilizado neste trabalho.

FIGURA 2 ALGORITMO RECOZIMENTO SIMULADO

Passo

1. Dado uma solução inicial S_0 e uma temperatura inicial T_0 ;
2. S_m (solução ótima global) recebe S_0 ;
3. S (solução atual) recebe S_0 ;
4. Início da Iteração;
 - a. Zere número de sucessos e de perturbações;
 - b. Pertube S para S_t (solução a ser testada) utilizando o método de perturbação;
 - c. Verifique validade de S_t ; caso não válido volte para 4.b; caso válido continue para 4.d;
 - d. Incremente número de perturbações;
 - e. Verifique se S_t satisfaz Critérios de Metropolis;
 - i. Se um dos critérios for satisfeito, S recebe S_t e o número de sucessos é incrementado;
 - ii. Verifique se S_t possui solução melhor que S_m ; Caso positivo, S_m recebe S_t ;
 - f. Verifique se o número de sucessos e perturbações são menores que o máximo por iteração; Caso positivo, retorne a 4.b; Caso negativo, fim da iteração; aplica-se a atualização na temperatura e incrementa-se o número de iterações;
5. Verifique se a temperatura é maior que o mínimo; Caso positivo, retorne a 4.a; Caso negativo fim do algoritmo;
6. A solução ótima é S_m ;

Fonte: Autor, 2022.

4.1 APLICAÇÃO DO RECOZIMENTO SIMULADO EM RECONFIGURAÇÃO DE REDES DE DISTRIBUIÇÃO

Para que um problema de otimização seja satisfeito, deve-se encontrar a melhor solução possível obedecendo a restrições impostas pelo problema. No presente trabalho, a melhor solução será aquela que apresenta a menor quantidade de perdas em todo o sistema, levando assim a obter diversas melhorias, tanto para o consumidor, que será abastecido, quanto para a concessionária, que abastecerá. Entre as restrições do trabalho, é exigida a ramificação da rede (radialidade do sistema), satisfazer aos critérios de máxima e mínima tensão em todas as cargas,

obedecer aos limites de fluxo de potência em cada linha de transmissão, e atender a todos os consumidores. A função objetivo é mostrada na Equação 1:

$$f(x) = \min \left(\sum_{l=1}^{N_l} r_l i_l^2 \right) \quad (1)$$

Onde:

- N_l : número total de linhas;
- r_l : resistência da linha l ;
- i_l : corrente que flui na linha l ;

4.2 ALGORITMO DE RECOZIMENTO SIMULADO PROPOSTO

Um dos objetivos específico do presente trabalho é utilizar o algoritmo de otimização Recozimento Simulado em conjunto com o OpenDSS para ser possível encontrar a melhor configuração de rede de forma generalizada e o mais rápido possível. Para tanto, foi desenvolvida uma ferramenta em linguagem C# utilizando o Visual Studio. Dentre seus objetivos primários destaca-se atender a qualquer rede escrita no OpenDSS, obtendo uma grande capacidade de generalização. Os parâmetros do algoritmo podem ser modificados antes de se iniciar o processo, permitindo uma adaptação ao tamanho da rede estudada.

A ferramenta, de forma geral, utilizou-se da COM fornecida pelo OpenDSS e apresentada no Capítulo 3 para ser possível unir a rápida linguagem de programação C# com uma resolução de fluxo eficiente. Os passos de como o software desenvolvido pelo autor funciona são descritos a seguir:

1. Inicializa-se o programa e o arquivo contendo a rede a ser estudada é carregada pelo usuário.

2. O programa mostra características da rede selecionada, e checa se a configuração inicial da rede está definida com todas as chaves fechadas. Depois, o usuário deve conferir as características da rede mostrada na interface gráfica e verificar se escolheu a rede devidamente.
3. O usuário deve configurar os parâmetros do Recozimento Simulado de acordo com a rede estudada.
4. Após a configuração dos parâmetros, o usuário pode iniciar o algoritmo de Recozimento simulado.
5. A solução inicial para o algoritmo é escolhida aleatoriamente pelo programa, satisfazendo inclusive as restrições propostas. Tal solução também é mostrada na tela ao fim do processamento do algoritmo.
6. A temperatura inicial do algoritmo é definida através do seguinte método: A partir da solução inicial, se define uma variável *perdas*, que é a soma das perdas totais em kW. A temperatura inicial T_0 é comumente definida em outras literaturas como sendo:

$$T_0 = \frac{-perdas(kW)}{\ln C} \quad (2)$$

Onde C é uma constante definida pelo usuário.

7. Com a temperatura e a solução inicial definidas, o loop do algoritmo tem início.
8. Ao chegar a uma temperatura menor que 0,01, o loop é finalizado e é mostrada na tela a melhor solução encontrada durante a varredura.

Uma descrição mais detalhada do algoritmo desenvolvido é apresentada no Capítulo 5.

5 MÉTODO PROPOSTO

Swing (neste caso podemos tomar como a barra 0). Desta forma, a barra 0 não deve deixar de ter conexão com o resto do sistema. Desta forma, temos o seguinte conjunto de chaves que podem fazer parte da solução inicial do problema:

$$C_1 = [S_1, S_2, S_3, S_4, S_5, S_6, S_7]$$

Importante ressaltar que neste conjunto de chaves entram as chaves que fazem parte de malhas. Aquelas que ao serem abertas fazem uma carga deixar de ser alimentada (já compõem um ramo) não são incluídas no método. A proposta é criar um número A de grupos de chaves sem repetição. Uma chave deve ser aberta em cada grupo para garantir a radialidade. Pode-se então definir um número de chaves NI nos intervalos:

$$NI = \frac{N}{A}$$

Onde N é o número total de chaves e A é o número de chaves a serem abertas para ser possível garantir a radialidade. Desta forma, para o sistema de 5 barras exemplo, temos:

$$NI = \frac{7}{3} \cong 2$$

Com isso, é possível definir A subconjuntos CI_n no conjunto de chaves C_1 :

$$CI_1 = [S_1, S_2]$$

$$CI_2 = [S_3, S_4]$$

$$CI_3 = [S_5, S_6, S_7]$$

Nota-se que em caso de chaves sobrando, elas são inseridas no último intervalo. Com a definição dos subconjuntos, um número aleatório entre o menor e

maior índice de chaves decide quais das chaves de cada subconjunto será aberta. Supondo-se os seguintes números aleatórios a_n escolhidos:

$$a_1 = 1,69 \in [0,2[= S_2$$

$$a_2 = 2,45 \in [2,4[= S_3$$

$$a_3 = 6,01 \in [4,7[= S_7$$

O método proposto então utiliza esses números para escolher uma solução inicial possível Sol_p :

$$Sol_p = [S_2, S_3, S_7]$$

Graficamente é possível observar que tal solução é plausível, pois atende a todos os consumidores e é uma solução radial. No algoritmo desenvolvido, são utilizadas funções fornecidas pelo OpenDSS (*Topology.NumLoops* e *Topology.NumIsolatedLoads*) para verificar se a solução encontrada é radial e atende a todos os consumidores. Caso a solução não seja plausível, o algoritmo é repetido até que uma solução inicial válida seja encontrada.

5.2 MÉTODO PROPOSTO DE GERAÇÃO DE VIZINHOS APLICADO A UM SISTEMA GENÉRICO DE 5 BARRAS

Seja uma nova solução inicial válida para o sistema anterior $Sol_0 = [S_1, S_3, S_5]$, definida de forma a facilitar a compreensão. Aqui, é necessária a definição de um novo conceito, chamado de Caminho Barra-Geração, que nada mais é o caminho de chaves de cada barra à barra swing, como mostrado a seguir e na Figura 4:

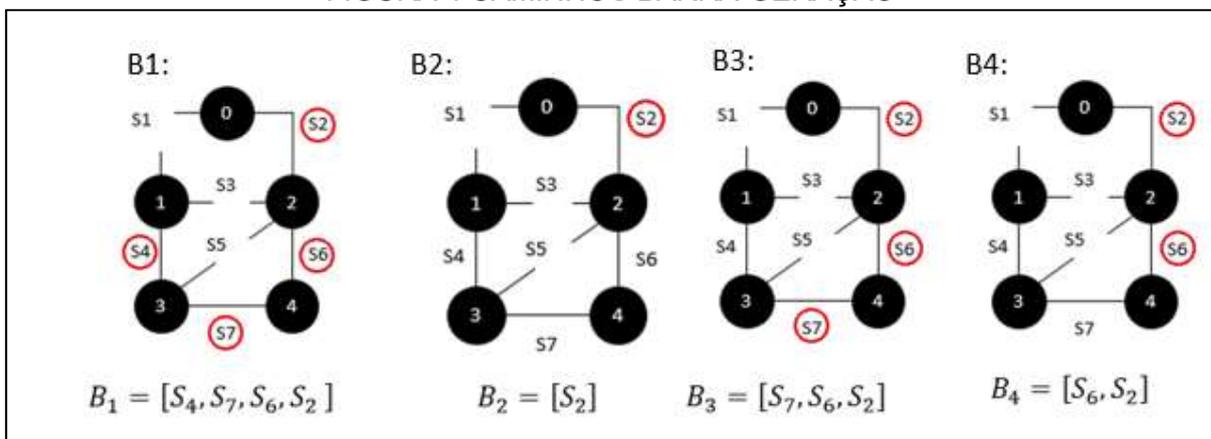
$$B_1 = [S_4, S_7, S_6, S_2]$$

$$B_2 = [S_2]$$

$$B_3 = [S_7, S_6, S_2]$$

$$B_4 = [S_6, S_2]$$

FIGURA 4 CAMINHOS BARRA-GERAÇÃO



Fonte: Autor , 2022.

Também é possível, a partir da solução inicial, construir a Matriz Conexão (MC). Cada linha dessa Matriz representa as barras as quais cada chave aberta é conectada, onde a primeira coluna é a barra “de” e a segunda coluna a barra “para”. Assim, por exemplo, na matriz *MC* o elemento “1” na segunda linha é a barra “de” da chave que se encontra na segunda posição no vetor “Chaves Abertas”, enquanto que o elemento “2” na mesma linha representa a barra “para” da mesma chave:

$$MC = \begin{bmatrix} & (de) & (para) \\ (S_1) & 0 & 1 \\ (S_3) & 1 & 2 \\ (S_5) & 2 & 3 \end{bmatrix}$$

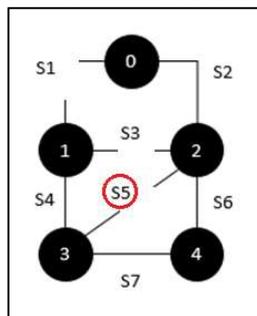
Agora, o algoritmo escolhe uma das chaves que participam da solução inicial aleatoriamente. Por exemplo ao escolher a S_5 , partindo do Caminho Barra-Geração, temos:

$$B_2 = [S_2]$$

$$B_3 = [S_7, S_6, S_2]$$

É mostrado na Figura 5 o esquema gráfico para melhor entendimento do desenvolvimento.

FIGURA 5 CAMINHOS DE E PARA

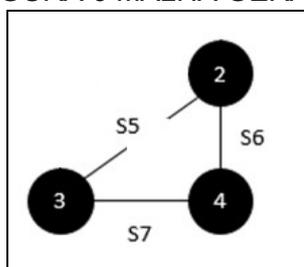


Fonte: Autor, 2022.

Para definir a malha formada a partir dessa chave é feita a diferença simétrica dos conjuntos. Desta forma é possível obter a Malha conforme descrito a seguir e mostrado graficamente na Figura 6. Convém notar que chaves que já fazem parte de outra Malha elaborada anteriormente não podem ser adicionadas às malhas subsequentes.

$$Malha_{s_5} = \text{diff}(B_2, B_3) = [S_7, S_6]$$

FIGURA 6 MALHA GERADA



Fonte: Autor, 2022.

Desta forma, podem-se obter as outras duas malhas formadas pelas chaves restantes na solução inicial. Suponha-se que as chaves foram escolhidas na seguinte ordem: S_1 e depois S_3 . As seguintes malhas são formadas:

$$Malha_{s_1} = [S_2, S_4]$$

$$Malha_{s_3} = [\quad]$$

De posse das Malhas elaboradas, uma garantia da radialidade é fechando a chave aberta e abrindo uma das chaves que fazem parte da Malha gerada pela chave. A forma escolhida para escolher qual chave virá a ser aberta foi a aleatoriedade. Conforme explicado acima, temos as 3 malhas geradas:

$$Malha_{S5} = [S_7, S_6]$$

$$Malha_{S1} = [S_2, S_4]$$

$$Malha_{S3} = [\quad]$$

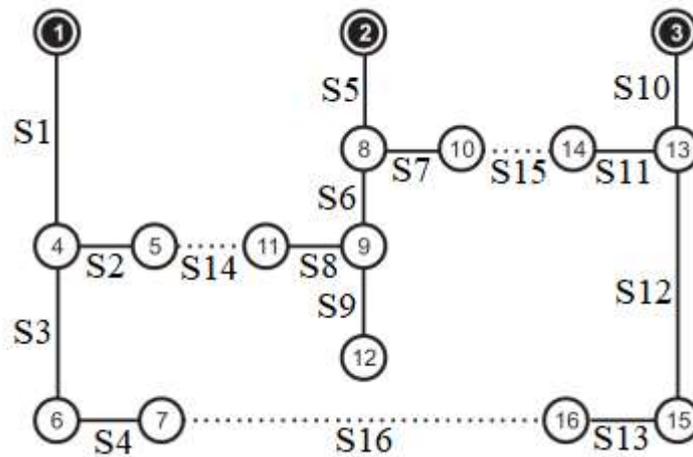
Aqui, pode-se escolher quaisquer uma das 3 malhas para se fazer a comutatividade. Supondo que aleatoriamente se escolha a malha formada pela chave 5, obrigatoriamente a chave 5 será fechada, e por randomização ao se fechar qualquer uma das chaves que fazem parte desta malha a radialidade está garantida.

Tal método é eficaz principalmente porque garante uma adequabilidade a qualquer alteração feita no sistema (as malhas geradas sempre serão diferentes, dependendo da ordem das chaves escolhidas na solução de partida).

5.3 RECOZIMENTO SIMULADO PROPOSTO APLICADO A UM SISTEMA DE 16 BARRAS

Nesta Seção será apresentado o método de recozimento simulado proposto aplicado ao Sistema de 16 Barras proposto por Civanlar *et al* (1988), com a descrição de três iterações completas. Tal sistema possui 16 ramos, tendo cada um uma chave acoplada. São 12 chaves normalmente fechadas e 3 chaves normalmente abertas. A representação unifilar do sistema é apresentada na Figura 7.

FIGURA 7 SISTEMA DE 16 BARRAS



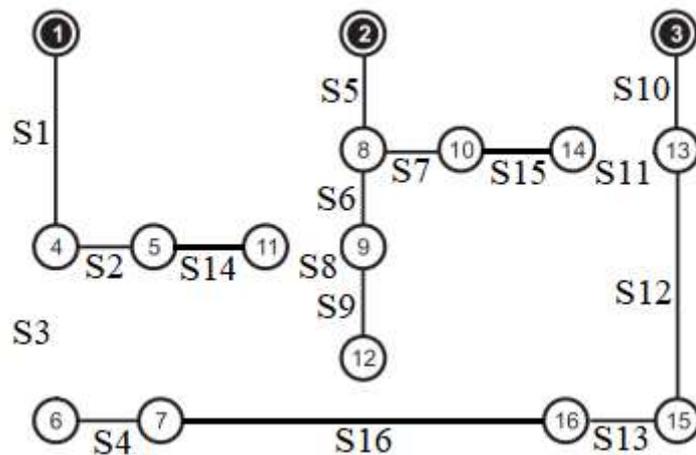
Fonte: Fontan *et al.*, 2008.

Seja uma solução inicial aleatória R_0 com perdas p_0 escolhida pelo método apresentado na Seção 5.1 como descrita a seguir e mostrada na Figura 8:

$$R_0 = \{S_3, S_8, S_{11}\}$$

$$p_0 = 563,79 \text{ kW}$$

FIGURA 8 APLICAÇÃO DO MÉTODO - SOLUÇÃO INICIAL



Fonte: Autor, 2022.

Sejam os parâmetros do Recozimento Simulado como definidos a seguir, definidos pelo autor de forma a facilitar a compreensão:

$$Per_{iter} = 2$$

$$Suc_{iter} = 2$$

$$\alpha = 0,9$$

$$Rand = 0,6$$

$$C = 0,8$$

$$T_f = 0,01$$

Onde Per_{iter} é a quantidade de perturbações por iteração, Suc_{iter} é o número de sucessos por iteração, α é a taxa de decaimento da temperatura, $Rand$ é um número definido entre 0 e 1 para representar um número aleatório, C é uma constante utilizada para cálculo de temperatura e T_f é a temperatura final do algoritmo. Per_{iter} e Suc_{iter} são livres para escolha. Os valores de α , $rand$ e C são típicos da literatura. O valor de $rand$ pode ser constante durante todo o algoritmo ou também variável (entre 0 e 1).

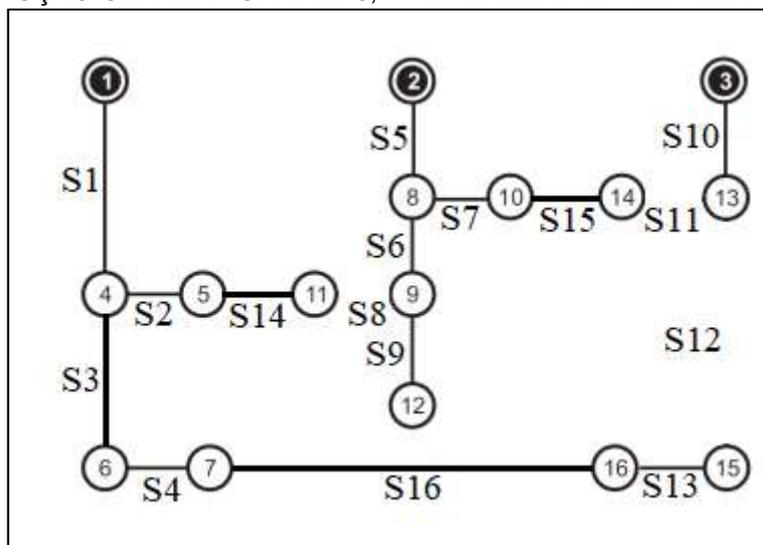
Com base na solução inicial, a temperatura inicial definida é: $T_0 = -\frac{563,79}{\ln 0.8} = 2526,57$. Aqui, tem-se início ao algoritmo do método proposto:

1. Iteração 1 do Recozimento Simulado:

- Geração de perturbação 1:
 - i. Ordem de escolha das chaves: {3,8,11}
 - ii. Malhas geradas:
 1. $M_3 = \{1,4,16,13,12,10\}$
 2. $M_8 = \{14,2,6,5\}$
 3. $M_{11} = \{15,7\}$
 - iii. Escolha da malha de comutação: M_3
 - iv. Escolha da chave de comutação: C_{12}
 - v. Vizinho gerado: $ST_1 = \{8, 11, 12\}$; $pt_1 = 588,46 \text{ kW}$
 - vi. $\Delta_p = 24,67 \text{ kW}$ (falha)
 - vii. $Termo \text{ exponencial} = 0,99 > 0,6$ (sucesso)
 - viii. $pert = pert + 1 = 1$

- ix. $suc = suc + 1 = 1$
- x. $S_1 \leftarrow ST_1 = \{8, 11, 12\}$ – Mostrado na Figura 9.

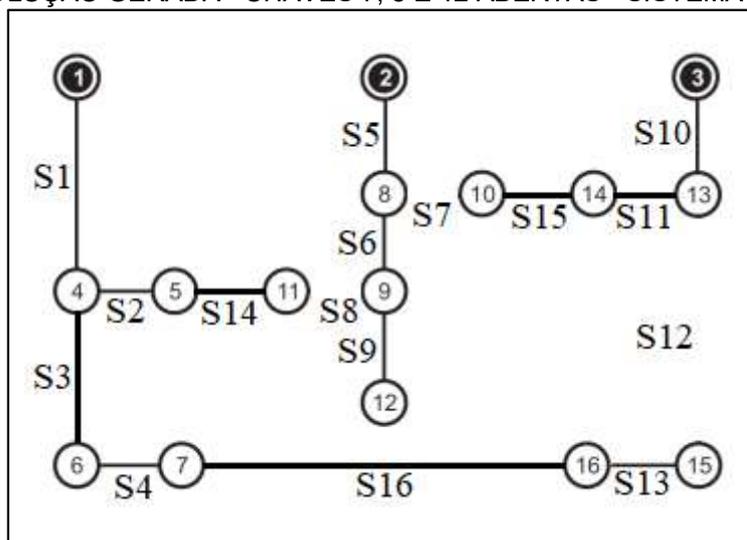
FIGURA 9 SOLUÇÃO GERADA - CHAVES 8, 11 E 12 ABERTAS - SISTEMA DE 16 BARRAS



Fonte: Autor, 2022.

- $suc \geq Suc_{iter}$? Não
- $pert \geq Per_{iter}$? Não
- Geração de perturbação 2:
 - i. Ordem de escolha das chaves: $\{11, 12, 8\}$
 - ii. Malhas geradas:
 1. $M_{11} = \{15, 7, 5, 10\}$
 2. $M_{12} = \{13, 16, 4, 3, 1\}$
 3. $M_8 = \{14, 2, 6\}$
 - iii. Escolha da malha de comutação: M_{11}
 - iv. Escolha da chave de comutação: C_7
 - v. Vizinho gerado: $ST_2 = \{7, 8, 12\}$; $pt_2 = 527,98 \text{ kW}$
 - vi. $\Delta_p = -60,38 \text{ kW}$ (sucesso)
 - vii. $pert = pert + 1 = 1$
 - viii. $suc = suc + 1 = 1$
 - ix. $S_1 \leftarrow ST_2 = \{7, 8, 12\}$ – Mostrado na Figura 10.

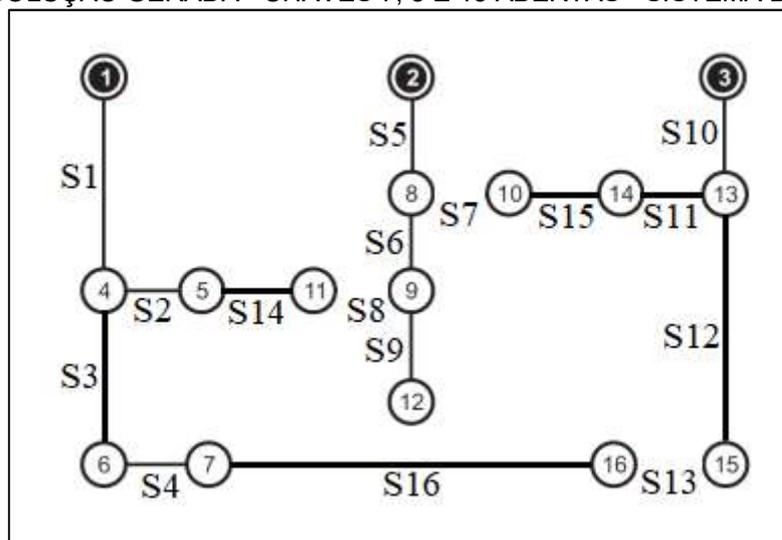
FIGURA 10 SOLUÇÃO GERADA - CHAVES 7, 8 E 12 ABERTAS - SISTEMA DE 16 BARRAS



Fonte: Autor, 2022.

- $suc \geq Suc_{iter}$? Sim
 - $T \leftarrow 0,9 * T = 2273$
2. Iteração 2 do Recozimento Simulado: (Parte da solução S_1)
- Geração de Perturbação 1:
 - i. Ordem de escolha das chaves: {7,8,12}
 - ii. Malhas geradas:
 1. $M_7 = \{5,15,11,10\}$
 2. $M_8 = \{14,2,1,6\}$
 3. $M_{12} = \{13,16,4,3\}$
 - iii. Escolha da malha de comutação: M_{12}
 - iv. Escolha da chave de comutação: C_{13}
 - v. Vizinho gerado: $ST_1 = \{7,8,13\}$; $pt_1 = 495,05 \text{ kW}$
 - vi. $\Delta_p = -32,85 \text{ kW}$ (sucesso)
 - vii. $pert = pert + 1 = 1$
 - viii. $suc = suc + 1 = 1$
 - ix. $S_2 \leftarrow ST_1 = \{7,8,13\}$ – Mostrado na Figura 11.

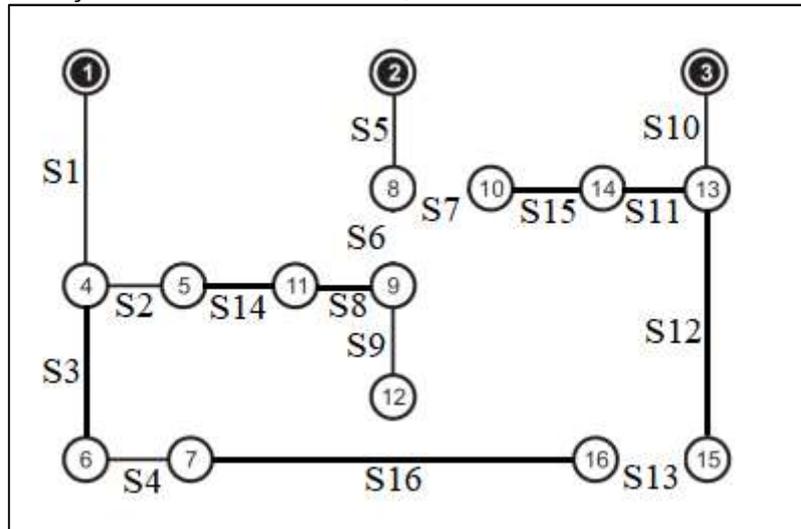
FIGURA 11 SOLUÇÃO GERADA - CHAVES 7, 8 E 13 ABERTAS - SISTEMA DE 16 BARRAS



Fonte: Autor, 2022.

- $suc \geq Suc_{iter}$? Não
- $pert \geq Per_{iter}$? Não
- Geração de perturbação 2:
 - i. Ordem de escolha das chaves: {7,13,8}
 - ii. Malhas geradas:
 1. $M_7 = \{5,15,11,10\}$
 2. $M_{13} = \{16,4,3,1,12\}$
 3. $M_8 = \{14,2,6\}$
 - iii. Escolha da malha de comutação: M_8
 - iv. Escolha da chave de comutação: C_6
 - v. Vizinho gerado: $ST_2 = \{6,7,13\}$; $pt_2 = 770,16 \text{ kW}$
 - vi. $\Delta_p = 275,16 \text{ kW}$ (falha)
 - vii. $Termo \text{ exponencial} = 0,88 > 0,6$ (sucesso)
 - viii. $pert = pert + 1 = 1$
 - ix. $suc = suc + 1 = 1$
 - x. $S_2 \leftarrow ST_2 = \{6,7,13\}$ - Mostrado na Figura 12.

FIGURA 12 SOLUÇÃO GERADA - CHAVES 6, 7 E 13 ABERTAS - SISTEMA DE 16 BARRAS



Fonte: Autor, 2022.

- $suc \geq Suc_{iter}$? Sim
- $T \leftarrow 0,9 * T = 2046,529$

Pode-se observar como sendo característica do recozimento simulado que nas primeiras iterações, devido a temperatura elevada, as perturbações quase sempre irão resultar em sucessos, mesmo que a solução não seja melhor que a sua solução mãe. O algoritmo desenvolvido tem a capacidade de salvar a melhor solução global encontrada enquanto percorre o espaço de busca atrás de soluções melhores. Então, mesmo que ao final do loop ele esteja em uma solução pior que a melhor encontrada, a solução ótima global é que será mostrada ao usuário.

6 RESULTADOS

6 RESULTADOS

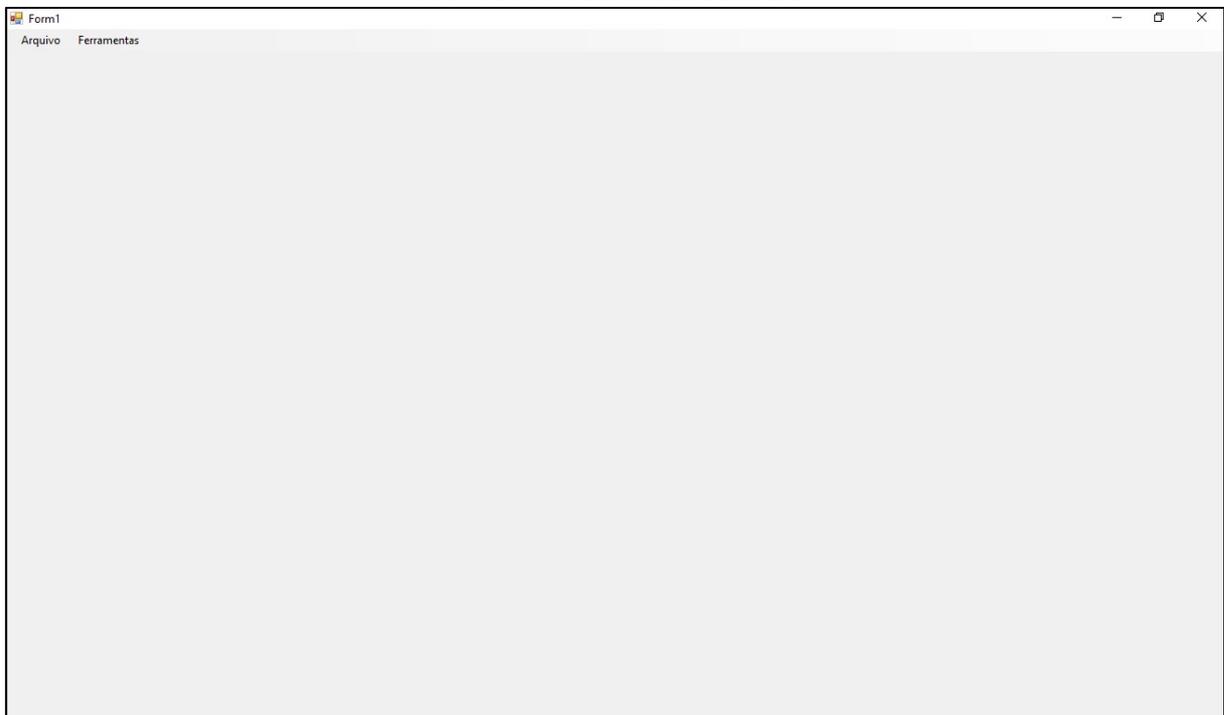
Neste capítulo são apresentados os resultados obtidos com o software desenvolvido em um sistema de 5 barras (utilizado para validação do algoritmo desenvolvido), além dos sistemas de 33 barras e 69 barras, e comparados os resultados com algumas referências. No início do capítulo há uma apresentação da interface gráfica desenvolvida.

6.1 INTERFACE GRÁFICA

Foi desenvolvida uma interface gráfica que permite uma interação usuário-algoritmo de forma clara e fácil, permitindo que um usuário comum possa alterar os parâmetros do recozimento simulado de forma simples e se chegar a uma otimização da rede de forma dinâmica.

A interface gráfica desenvolvida para o trabalho é demonstrada a seguir. A tela inicial é mostrada na Figura 13.

FIGURA 13 Tela Inicial Software



Fonte: Autor, 2022

É possível, a partir da opção “Arquivo”, escolher um código que contenha uma estrutura desenvolvida no OpenDSS. Por exemplo, ao abrir o código do sistema de 5 barras utilizado no Capítulo 5, obtêm-se as Figuras 14 a 16, que contêm as características do sistema.

FIGURA 14 CARACTERÍSTICAS DE CARGA

	Nome	Carga (kW)	Carga (kVAR)
▶	loadb2	128	128
	loadb3	32	16
	loadb4	160	80
	loadb5	74	37
*			

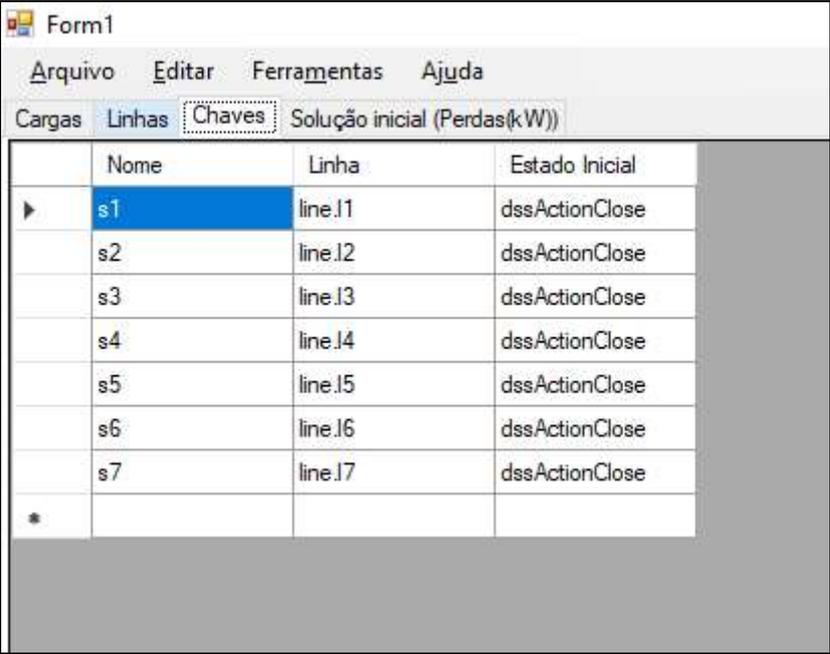
Fonte: Autor, 2022

FIGURA 15 CARACTERÍSTICAS DE LINHA

	Nome	De	Para
▶	l1	b1	b2
	l2	b1	b3
	l3	b2	b3
	l4	b2	b4
	l5	b3	b4
	l6	b3	b5
	l7	b4	b5
*			

Fonte: Autor, 2022

FIGURA 16 CARACTERÍSTICAS DAS CHAVES

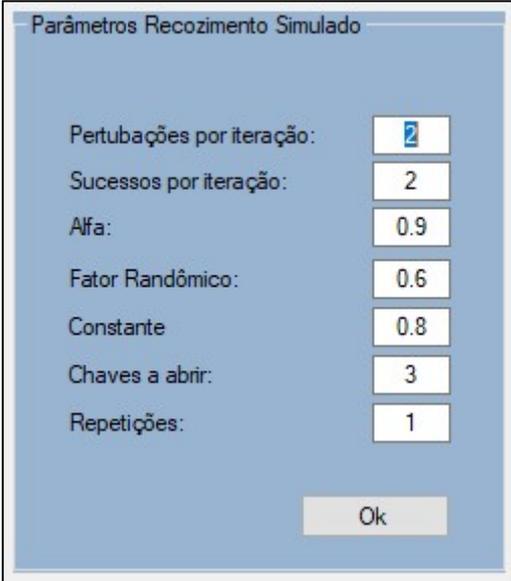


	Nome	Linha	Estado Inicial
▶	s1	line.11	dssActionClose
	s2	line.12	dssActionClose
	s3	line.13	dssActionClose
	s4	line.14	dssActionClose
	s5	line.15	dssActionClose
	s6	line.16	dssActionClose
	s7	line.17	dssActionClose
*			

Fonte: Autor, 2022

Logo após o carregamento do código, é possível, através da aba ferramentas, escolher os parâmetros a serem utilizados no algoritmo de Recozimento Simulado, e também rodar a otimização. Os parâmetros disponíveis para alteração são mostrados na Figura 17.

FIGURA 17 PARÂMETROS DISPONÍVEIS PARA AJUSTE



Pertubações por iteração:	2
Sucessos por iteração:	2
Alfa:	0.9
Fator Randômico:	0.6
Constante	0.8
Chaves a abrir:	3
Repetições:	1

Ok

Fonte: Autor, 2022.

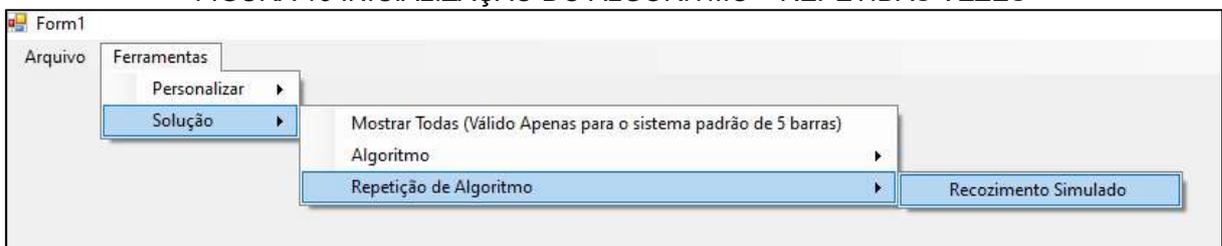
O software desenvolvido oferece três opções de execução do algoritmo: Uma exclusiva para o sistema de 5 barras, desenvolvida para validação do algoritmo, sob descrição de “Mostrar todas (Válido Apenas para o sistema padrão de 5 barras)”, onde se mostram todos os resultados possíveis para o sistema de 5 barras. A segunda opção oferece a possibilidade de execução única do algoritmo de recozimento simulado, conforme Figura 18. A terceira opção, destacada na Figura 19, é a possibilidade de rodar o algoritmo de recozimento simulado repetidas vezes através da escolha do parâmetro Repetições, mostrado na Figura 19.

FIGURA 18 INICIALIZAÇÃO DO ALGORITMO – ÚNICA VEZ



Fonte: Autor, 2022.

FIGURA 19 INICIALIZAÇÃO DO ALGORITMO – REPETIDAS VEZES



Fonte: Autor, 2022.

Desta forma, é possível alterar os parâmetros de forma dinâmica e rápida, podendo ser comparados diversos resultados a partir de troca de abas, conforme mostrado na Figura 20.

FIGURA 20 PÁGINA DE RESULTADOS

Iteração	Chave Aberta 01	Chave Aberta 02	Chave Aberta 03	Perdas (kW)	Temo Dif	Temo Exponencial	Temo randomico	Troca?	Temperatura
37	1	5	7	5,35	1,74	0,0226444163775852	0,6		0,459364464380...
37	1	4	7	3,69	0,080000000000...	0,84016779257902	0,6	Sim (Por exponen...	0,459364464380...
38	1	5	7	5,35	1,66	button2 383301	0,6		0,413428017942...
38	1	5	6	7,37	3,68	0,000136227161409987	0,6		0,413428017942...

Fonte: Autor, 2022.

A interface desenvolvida cumpre seu papel de tornar prática e fácil a interação com o OpenDSS e a configuração e execução do algoritmo de recozimento simulado.

6.2 SISTEMA DE 5 BARRAS

O software desenvolvido também tem a opção de mostrar todas as soluções possíveis para o sistema de 5 barras, como já mencionado neste trabalho. Os resultados obtidos são mostrados na Figura 21.

FIGURA 21 SOLUÇÕES PARA 5 BARRAS

Cargas	Linhas	Chaves	Solução inicial (Perdas(kW))	Soluções Possíveis
	Chave Aberta 1	Chave Aberta 2	Chave Aberta 3	Solução (Perdas)
	3	5	7	9,92
	2	5	7	16,16
	1	5	7	5,35
	3	4	7	3,61
	2	4	7	14,86
	1	4	7	3,69
	2	3	7	19,24
	1	3	7	5,63
	3	5	6	16
	2	5	6	18,13
	1	5	6	7,37
	3	4	6	3,82
	2	4	6	15,1
	1	4	6	3,9
	2	3	6	19,21
	1	3	6	5,98
	3	4	5	6,04
	2	4	5	17,59
	1	4	5	6,14
	2	3	5	19,51
	1	3	5	14,09

Fonte: Autor, 2022.

Pode-se observar que o melhor resultado é obtido quando as chaves 3, 4 e 7 estão abertas. Para os resultados neste sistema, os parâmetros a seguir foram escolhidos:

$\alpha = 0.9$ $rand = 0.6$ $C = 0.8$ Chaves a abrir = 3

Os resultados obtidos foram produzidos em 100 repetições conforme parâmetros descritos e mostrados na Tabela 9.

TABELA 9 RESULTADOS OBTIDOS SISTEMA 5 BARRAS

PARÂMETROS		Resultados				
Perturbações por iteração	Sucessos por iteração	Média de perdas (kW)	Desvio padrão (kW)	Perdas ótimas	Solução ótima	Nº de recorrência
2	2	4,721	1,03	3,61	3,4,7	45
3	2	4,519	1,03	3,61	3,4,7	55
3	3	4,014	0,829	3,61	3,4,7	80
4	4	3,61	0,00	3,61	3,4,7	100

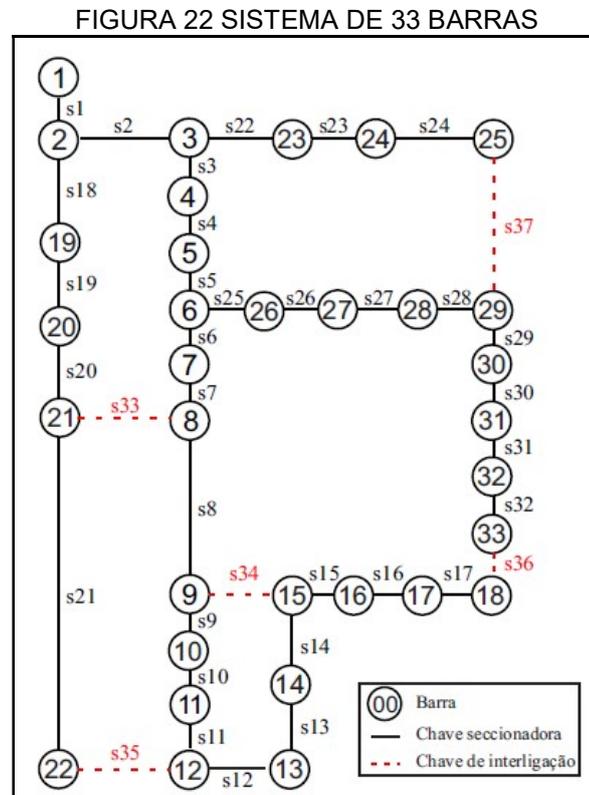
Fonte: Autor, 2022.

Pode-se observar que um melhor resultado foi obtido quando mais possibilidades são incluídas dentro de uma mesma iteração.

6.3 SISTEMA DE 33 BARRAS

O sistema de 33 barras utilizado neste trabalho foi proposto por Baran e Wu (1989), possuindo 37 ramos e tensão de 12,66 kV. Cada ramo possui uma chave, sendo 32 seccionadoras e 5 de interligação. O total de carga ativa é de 3715 kW e o total de reativa é de 2330 kvar. Com as chaves em seus estados normais, a perda

total nas linhas é de 202,67 kW. O diagrama desse sistema é apresentado na Figura 22.



Assim como utilizado na Seção 6.2, as únicas alterações dos parâmetros entre testes foram nos parâmetros sucesso e perturbação por iteração. Os parâmetros a seguir permaneceram constantes:

$\alpha = 0.9$
$rand = 0.6$
$C = 0.8$
Chaves a abrir = 5

São mostrados na Tabela 10 os resultados obtidos. Pode-se notar que há uma pequena alteração entre os valores de perdas para uma mesma solução, podendo-se

admitir que tal variação é resultante do método de cálculo da potência ativa utilizado na programação.

TABELA 10 RESULTADOS SISTEMA 33 BARRAS

PARÂMETROS		Resultados				
Pertubações por iteração	Sucessos por iteração	Média de perdas (kW)	Desvio padrão (kW)	Perdas ótimas	Solução ótima	Nº de recorrência
2	2	139,69	1,02	139,32	7,9,14,32,37	77
3	3	139,33	0,06	139,32	7,9,14,32,37	98
4	4	139,36	0,25	139,32	7,9,14,32,37	97
6	6	139,32	0,01	139,32	7,9,14,32,37	100

Fonte: Autor, 2022.

Observa-se na Tabela 10 que um aumento no número de geração de vizinhos por iteração tende a melhorar o resultado obtido. Tal fenômeno pode-se observar nas variações do primeiro pro segundo cenário e do terceiro pro quarto cenário. Apesar disso, o terceiro cenário obteve resultados piores que o segundo, tanto em média de perdas como em variação. O último cenário, com 6 perturbações e 6 sucessos, conseguiu achar a solução ótima em todas as vezes. Apesar disso, todos os cenários obtêm uma boa média de perdas, com o pior cenário excedendo o melhor cenário em apenas 0,2%. Por fim, o resultado ótimo obtido pelo método proposto é se abrindo as chaves 7, 9, 14, 32 e 37, obtendo perdas de 139,32 kW.

O algoritmo desenvolvido foi comparado, na mesma ferramenta utilizada para simulação, com mais três referências. Os resultados obtidos se encontram na Tabela 11.

TABELA 11 COMPARAÇÃO COM OUTRAS REFERÊNCIAS - SISTEMA 33 BARRAS

Parâmetro	FONTE	Cenário 01	Cenário 02	Cenário 03	Cenário 04
Média	AUTOR	139,69	139,33	139,36	139,32
	[24]*	145,3	143,4	142,69	141,27
	Recozimento Simulado Padrão	140,35	139,83	139,63	139,69
	[32]*	140,35	139,94	139,64	139,63
Desvio padrão (kW)	AUTOR	1,02	0,06	0,25	0,01
	[24]*	3,372	3,4682	3,0015	2,2262
	Recozimento Simulado Padrão	1,7866	1,3884	0,8702	1,0243
	[32]*	1,7756	1,7761	1,1009	0,9381
Perdas Ótima (kW)	AUTOR	139,32	139,32	139,32	139,32
	[24]*	139,32	139,32	139,32	139,32
	Recozimento Simulado Padrão	139,32	139,32	139,32	139,32
	[32]*	139,32	139,32	139,32	139,32
Nº de recorrência	AUTOR	77	98	97	100
	[24]*	6	14	24	36
	Recozimento Simulado Padrão	66	76	80	86
	[32]*	56	85	85	85

Fonte: Autor, 2022. (*Os algoritmos das referências foram adaptados para permitir comparação entre métodos.)

Onde os cenários são descritos a seguir:

Cenário 01: $Per_{iter} = 2, Suc_{iter} = 2$
Cenário 02: $Per_{iter} = 3, Suc_{iter} = 3$
Cenário 03: $Per_{iter} = 4, Suc_{iter} = 4$
Cenário 04: $Per_{iter} = 6, Suc_{iter} = 6$

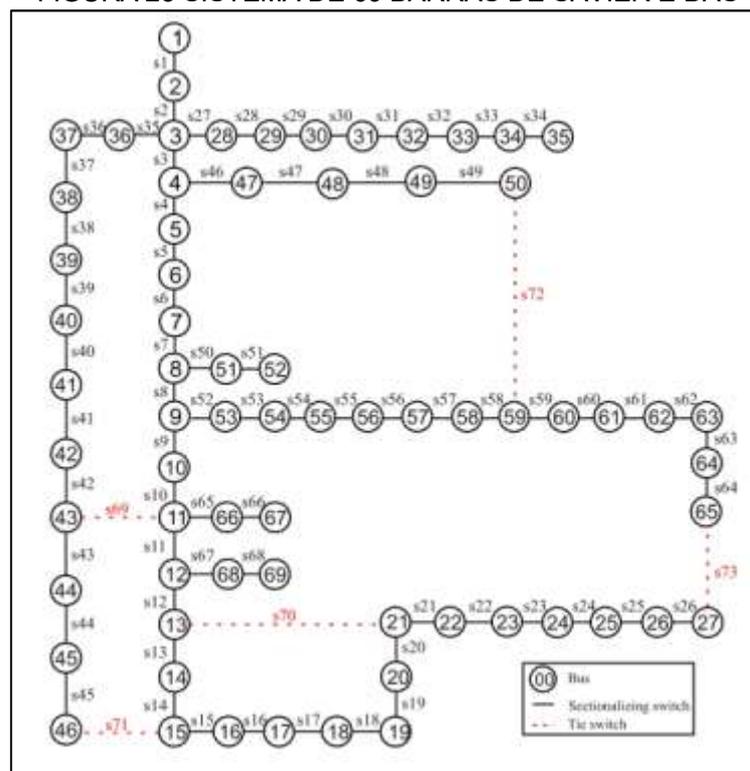
Pode-se observar que o algoritmo desenvolvido pelo autor é bem mais eficiente em qualquer critério, principalmente na precisão, levando em consideração os mesmos parâmetros

de busca. É importante ressaltar que os algoritmos dos outros trabalhos foram ajustados para que a comparação com o método desenvolvido fosse possível.

6.4 SISTEMA DE 69 BARRAS

O sistema de 69 barras utilizado neste trabalho foi proposto por Savier e Das (2007) e possui 73 ramos e tensão de 12,66 kV. Cada ramo possui uma chave, como mostrado no diagrama unifilar descrito na Figura 23, sendo 68 chaves normalmente fechadas (contínuas) e 5 chaves normalmente abertas (tracejadas). Com as chaves normalmente abertas em seu estado original, o sistema possui perdas de 224,96 kW nas linhas.

FIGURA 23 SISTEMA DE 69 BARRAS DE SAVIER E DAS



Fonte: Pegado, 2019.

Novamente, nos testes, os parâmetros a seguir permanecem constantes.

Alpha = 0.9
Fator randômico = 0.6
Constante = 0.8
Chaves a abrir = 5

Os parâmetros sucessos por iteração e perturbações por iteração são modificados para se realizar os testes, conforme é mostrado na Tabela 12.

Novamente, observa-se que conforme o número de alterações por iteração aumenta, a precisão do algoritmo aumenta, apesar do aumento do custo de tempo de execução obtido nas simulações. Os tempos foram calculados para previsão do tempo total de simulação e não fazem parte da comparação. Observa-se que o pior caso em termos de recorrência da melhor solução possui uma média de perdas próxima ao melhor caso, excedendo esta em apenas 0,5%. A solução ótima encontrada foi ao se abrir as chaves 14, 55, 61, 69 e 70, com perdas estimadas em 98,4 kW.

TABELA 12 RESULTADOS SISTEMA 69 BARRAS

PARÂMETROS		Resultados				
Perturbações por iteração	Sucessos por iteração	Média de perdas (kW)	Desvio padrão (kW)	Perdas ótimas	Solução ótima	Nº de recorrência
2	2	98,91	1,90	14,55,61,69,70	98,4	68
4	4	98,44	0,20	14,55,61,69,70	98,4	91
6	6	98,41	0,10	14,55,61,69,70	98,4	96
8	8	98,4	0,00	14,55,61,69,70	98,4	100

Fonte: Autor, 2022.

Na Tabela 13, pode-se observar os resultados obtidos quando comparados com outras literaturas.

TABELA 13 COMPARAÇÃO COM OUTRAS LITERATURAS - SISTEMA 69 BARRAS

Parâmetro	FONTE	Cenário 01	Cenário 02	Cenário 03	Cenário 04
Média	AUTOR	98,91	98,44	98,41	98,4
	[24]*	107,09	102,27	100,93	99,862
	Recozimento Simulado Padrão	99,504	98,536	98,419	98,425
	[32]*	99,626	98,737	98,477	98,406
Desvio padrão (kW)	AUTOR	1,9	0,2	0,1	0
	[24]*	7,89	3,91	3,02	2,3
	Recozimento Simulado Padrão	2,06	0,29	0,032	0,038
	[32]*	2,41	1,24	0,21	0,019
Perdas Ótima (kW)	AUTOR	98,4	98,4	98,4	98,4
	[24]*	98,49	98,4	98,4	98,4
	Recozimento Simulado Padrão	98,4	98,4	98,4	98,4
	[32]*	98,4	98,4	98,4	98,4
Nº de recorrência	AUTOR	68	91	96	100
	[24]*	4	4	2	18
	Recozimento Simulado Padrão	35	40	50	55
	[32]*	55	75	45	85

Fonte: Autor, 2022. (*Os algoritmos das referências foram adaptados para permitir comparação entre métodos.)

Onde os cenários são descritos a seguir:

Cenário 01: $Per_{iter} = 2, Suc_{iter} = 2$

Cenário 02: $Per_{iter} = 4, Suc_{iter} = 4$

Cenário 03: $Per_{iter} = 6, Suc_{iter} = 6$

Cenário 04: $Per_{iter} = 8, Suc_{iter} = 8$

Novamente, assim como ocorreu com o sistema de 33 barras, o algoritmo desenvolvido saiu-se melhor, principalmente no quesito precisão.

7 CONCLUSÕES

7 CONCLUSÕES

Neste trabalho foi desenvolvido um software com Interface Gráfica, que foi capaz de utilizar os recursos da linguagem C#, juntamente com o ambiente de desenvolvimento Visual Studio para que a integração entre o usuário e o OpenDSS fosse dinamizado. Através da linguagem C# foi possível a aplicação de um algoritmo que garante a radialidade do sistema de forma iteligente, além de ter sido possível a implementação do Recozimento Simulado. Com a Integração Gráfica, foi possível dinamizar e obter resultados de forma simples com uma interface prática.

A implementação da forma automática e inteligente do encontro de malhas permitiu poupar bastante tempo de processamento, pois restringe e elimina as combinações de chaves que não são válidas.

O método de encontro da solução inicial também foi uma importante implementação no trabalho, facilitando o encontro de uma solução inicial radial aleatória.

O algoritmo de Recozimento simulado foi testado em dois sistemas, obtendo bons resultados com diferentes parâmetros, principalmente quando comparado com diversas literaturas.

Como forma de melhoria para o trabalho, propõe-se buscar por métodos de paradas mais específicos a cada situação, desenvolver um código de programação mais compacto e melhorar a interface gráfica desenvolvida.

REFERÊNCIAS

- [1] Richard E. Brown. Distribution reliability assessment and reconfiguration optimization. 2001.
- [2] S. Nie et al. Analysis of the impact of DG on distribution network reconfiguration using OpenDSS. 2012.
- [3] H. S. Eichkoff e A. P. C. de Mello R. C. Marques. Analysis of the distribution network reconfiguration using the OpenDSS software. 2018.
- [4] Ana Paula Carboni de. Mello. Reconfiguração de Redes de Distribuição Considerando Multivariáveis e Geração Distribuída. PhD thesis, Universidade Federal de Santa Maria, 2014.
- [5] D. P. et al. BERNADON. Sistema de distribuição das redes elétricas inteligentes: uma abordagem para reconfiguração de redes. 2015.
- [6] M. Antoneie et al. Development of distribution network model in OpenDSS using matlab and gis data. 2019.
- [7] M. Z. Zeb et al. Optimal placement of electric vehicle charging stations in the active distribution network. 2020.
- [8] B. A. R. de Andrade and N. R. Ferreira. Simulated annealing and tabu search applied on network reconfiguration in distribution systems. 2018.
- [9] H-D. Chiang and Rene Jean-Jumeau. Optimal Network reconfigurations in distribution systems, i. a new formulation and a solution methodology. 1990.
- [10] Merlin, A. Search for a minimal-loss operating spanning tree configuration for an urban power distribution system. Proc. of 5th PSCC, 1975, v. 1, p. 1–18, 1975.
- [11] Civanlar, S. et al. Distribution feeder reconfiguration for loss reduction. IEEE Transactions on Power Delivery, IEEE, v. 3, n. 3, p. 1217–1223, 1988.
- [12] Shirmohammadi, D.; Hong, H. W. Reconfiguration of electric distribution networks for resistive line losses reduction. IEEE Transactions on Power Delivery, IEEE, v. 4, n. 2, p. 1492–1498, 1989.
- [13] BARAN, M. E.; WU, F. F. Network reconfiguration in distribution systems for loss reduction and load balancing. IEEE Power Engineering Review, IEEE, v. 9, n. 4, p. 101–102, 1989.

- [14] PEPONIS, G.; PAPADOPOULOS, M.; HATZIARGYRIOU, N. Distribution network reconfiguration to minimize resistive line losses. *IEEE Transactions on Power Delivery*, IEEE, v. 10, n. 3, p. 1338–1342, 1995.
- [15] Raoni de Araújo Pegado et al. Reconfiguração de redes de distribuição de energia elétrica usando otimização por enxame de partículas aprimorado. 2019.
- [16] A. Augugliaro, L. Dusonchet e R. Sanseverino. Genetic, Simulated Annealing and Tabu Search algorithms: Three Heuristic Methods for Optimal Reconfiguration and Compensation of Distribution Networks. 1999.
- [16] E. Riva Sanseverino A. Augugliaro, L. Dusonchet. Genetic, simulated annealing and tabu search algorithms: Three heuristic methods for optimal reconfiguration and compensation of distribution networks. 1999.
- [17] Jeon, Y.-J.; Kim, J.-C. Network reconfiguration in radial distribution system using simulated annealing and tabu search. In: IEEE. 2000 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 00CH37077). [S.l.], 2000. v. 4, p. 2329–2333.
- [18] Zhigang, MA. Study on Distribution Network Reconfiguration Based on Genetic Simulated Annealing Algorithm. 2008.
- [19] A. Skoonpong e S. Sirisumrannukul. Network reconfiguration for reliability worth enhancement in distribution systems by simulated annealing. 2008.
- [20] Chen, Jie; Zhang, Fan; Zhang Yuexi. Distribution Network Reconfiguration Based on Simulated Annealing Immune Algorithm. ICSGCE 2011: 27-30 September 2011, Chengdu, China. 2011.
- [21] Chen, E.; Zhang, S.; Wang T. Research on Distribution Network Reconstruction Based on Improved Simulated Annealing – Ant Colony Algorithm. 2017.
- [22] Koziel, S.; Rojas, A. L.; Moskwa, S. Power Loss Reduction through Distribution Network Reconfiguration Using Feasibility-Preserving Simulated Annealing. 2018.
- [23] Zhang, J.; Li, Z.; Wang, B. Withi-day rolling optimal scheduling problem for active distribution networks by multi-objective evolutionary algorithm based on decomposition integrating with thought of simulated annealing. 2021.
- [24] Andrade, B. A. R. de; Ferreira, N. R. Simulated annealing and tabu search applied on network reconfiguration in distribution systems. In: IEEE. 2018 Simposio Brasileiro de Sistemas Eletricos (SBSE). [S.l.], 2018. p. 1–6.
- [25] Electric Power Research Institute. The open distribution system, opendss. 2016.

-
- [26] DUGAN, R. C.; MCDERMOTT, T. E. An open source platform for collaborating on smart grid research. In: IEEE. 2011 IEEE Power and Energy Society General Meeting. [S.l.], 2011. p. 1–7.
- [27] F. Eldali M. E. Samper, A. Vargas and S. Suryanarayanan. Assessments of battery storage option for distribution expansion planning using an opendss-based framework. 2017.
- [28] D. A. Pavlyuchenko A. Z. Abass and Z.S. Hussain. Methods comparison for optimal capacitor placement in distribution system. 2020.
- [29] J. Taylor and B. Deaver. Platform for virtual prototyping of advanced distribution management systems using python and opendss. 2016.
- [30] P Radatz. Modelos avançados de análise de redes elétricas inteligentes utilizando o software OpenDSS. 2015.
- [31] Kirkpatrick, S.; Gellat, C. D.; Vecchi M.P. Optimization by Simulated Annealing. 1983.
- [32] Hong-Chan Chang and Cheng-Chien Kuo. Network reconfiguration in distribution systems using simulated annealing. 1994.
- [33] Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. Equation of State Calculations by Fast Computing Machines. Journal of Chemical Physics. 21 (6): 1087–1092. 1953.
- [34] David de Macêdo Braz, Helon. Configuração de Sistemas de Distribuição usando um Algoritmo genético Sequencial. Tese de doutorado. 2010.
- [35] ANEEL. Nota técnica nro 0057/2014-srd/aneel. 2014.