UNIVERSIDADE FEDERAL DO PARANÁ

LUCAS ROBERTO FERREIRA

HYBRID INTELLIGENT CONTROL FOR SMART GRIDFUNCTIONALITIES INTEGRATION

CURITIBA

2020

LUCAS ROBERTO FERREIRA

HYBRID INTELLIGENT CONTROL FOR SMART GRID FUNCTIONALITIES INTEGRATION

Tese apresentada ao curso de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Alexandre Rasi Aoki

Coorientador: Germano Lambert-Torres.

CURITIBA
2020

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

F383h Ferreira, Lucas Roberto
Hybrid intelligent control for smart grid functionalities integration [recurso eletrônico] / Lucas Roberto Ferreira – Curitiba, 2020.

Tese - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-graduação em Engenharia Elétrica.

Orientador: Alexandre Rasi Aoki

1. Redes Elétricas - Distribuição. 2. Sistema Avançado de Gerenciamento da Distribuição (ADMS). I. Universidade Federal do Paraná. II. Bonfim, Márlio José do Couto. III. Título.

CDD: 621.3192

Bibliotecária: Roseny Rivelini Morciani CRB-9/1585

MINISTÉRIO DA EDUCAÇÃO
SETOR DE TECNOLOGIA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA ELÉTRICA - 40001016043P4

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **LUCAS ROBERTO FERREIRA** intitulada: **HYBRID INTELLIGENT CONTROL FOR SMART GRID FUNCTIONALITIES INTEGRATION**, sob orientação do Prof. Dr. ALEXANDRE RASI AOKI, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 31 de Julho de 2020.

Assinatura Eletrônica
03/08/2020 16:33:45.0
ALEXANDRE RASI AOKI
Presidente da Banca Examinadora

Assinatura Eletrônica
03/08/2020 16:56:21.0
MARCUS THEODOR SCHILLING
Avaliador Externo (UNIVERSIDADE FEDERAL FLUMINENSE)

Assinatura Eletrônica
04/08/2020 10:56:20.0
THELMA SOLANGE PIAZZA FERNANDES
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
03/08/2020 16:49:42.0
FABIO ALESSANDRO GUERRA
Avaliador Externo (INSTITUTO GNARUS)

RESUMO

Ao longo dos anos, as redes de distribuição de energia estão ficando mais inteligentes e automatizadas, consequentemente problemas complexos emergem, onde estes são os gatilhos para melhorar antigos estudos e iniciar novas linhas de pesquisa. A Rede Elétrica Inteligente é o conceito abrangente para entender os novos problemas e alterar o comportamento tradicional do sistema para uma nova abordagem, partindo para uma rede com mais intercomunicação entre os elementos ativos. Para contribuir com avanços, a ideia principal desta tese é iniciar uma nova linha de pesquisa para combinar diferentes funcionalidades do Sistema Avançado de Gerenciamento da Distribuição (ADMS), a serem resolvidas por apenas um algoritmo ao mesmo tempo. Para iniciar os estudos dessa linha de desenvolvimento, foram selecionados os problemas mais comuns que causam grande impacto nas redes de distribuição, as interrupções inesperadas e as sobrecargas, resolvidas pelos algoritmos de Auto-Recuperação e Descarte de Carga, respectivamente. Os estudos atuais concentram-se em resolver o problema de Auto-Recuperação primeiro e depois, se o sistema iniciar ou manter uma sobrecarga, executar o descarte de carga para reduzir a carga e manter o sistema no modo operacional. No entanto, em vez de ter as duas funcionalidades trabalhando em um modo sequencial, por que não desenvolver um algoritmo exclusivo para processar o problema e resolvê-lo ao mesmo tempo, de forma simultânea? Assim, esta tese traz exatamente esse novo tipo de abordagem por meio da metodologia de Aprendizado por Reforço (um algoritmo de Machine Learning para tomar decisões) através do algoritmo Q-Learning. Em que os elementos do Q-Learning foram adaptado para reproduzir o ambiente como a rede de distribuição, a recompensa como a maximização da carga e as ações como a troca de posição das chaves (Auto-Recuperação) e a porcentagem de reduções de carga (Descarte da carga), a interagir no sistema para determinar o próximo estado (topologia). Para provar o algoritmo desenvolvido, foi utilizado um sistema urbano real com cinco alimentadores interconectados, onde o sistema foi dividido em um caso de três alimentadores, para determinar a escolha da política (a ε-greed foi a selecionada), criar alguns casos básicos e ser comparada com outras abordagens sequenciais. O caso completo foi usado para sobrecarregar o sistema e analisar os resultados para casos complexos. Em todas as simulações, os resultados encontraram uma boa solução após o estado de isolamento para maximizar a restauração da carga, e em alguns casos em que o sistema foi acionado por uma sobrecarga, o algoritmo pode, no mesmo momento, reconfigurar o sistema para evitar a sobrecarga e aplicar a redução de carga. Portanto, este trabalho forneceu uma nova linha de estudo e contribuir com uma nova linha de pesquisas a ser aprofundado em trabalhos futuros.

Palavras-chave: ADMS. Auto-Recuperação. Descarte de Carga. Aprendizado por Reforço. Q-Learning. Rede de Distribuição. Abordagem Simultânea.

ABSTRACT

Along the year, the distribution networks are getting more intelligent and automated, consequently complex problems emerge, where these are the triggers to improve old studies or start new lines of researches. The Smart Grid is the broad concept to understand the new problems and change the traditional system behavior for a new approach, where more intelligence and intercommunication is improved to solve the several distribution problems. To contribute on the network enhancements, the main idea of this thesis is to start a new line of research to combine different Advanced Distribution Management System (ADMS) functionalities to be solved by only one algorithm at the same time. To start the studies on this line of strategy, it was selected the most usual problems that has a big impact in distribution networks, the unexpected outages and the overloads, which are solved by Self-Healing and Load Shedding algorithms respectively. The current studies focus to solve the Self-Healing problem first and after, if the system initiate or maintain an overload, executes the Load Shedding to reduce the load and keeps the system in an operative mode. However, instead of having both functionalities working in a sequential mode, why not developed a unique algorithm to process both problem and solve them at the same time? Thus, this thesis brings exactly this new type of approach through the Reinforcement Learning methodology (a Machine Learning algorithm to take decisions) using the Q-Learning algorithm. The Q-Learning elements were adapted to reproduces environment as the distribution network, the reward as the maximization of load and the actions as the switch commutation (Self-Healing) and percentual of load reductions (Load Shedding) to be selected and interact on the system to determine the next state (topology). To prove the algorithm developed, it was used a real urban system with five interconnected feeders, where the system was divided in a three-feeder case, to determine the policy choice (ε-greed was selected), create some basic cases and be compared with other Self-Healing + Load Shedding sequential approaches. The complete case was used to overload the system and analyze the results for complex cases. In all simulations the results could find a good solution after the isolation state to maximizes the load restoration, and some cases where the system was trigger by an overload the algorithm could at the same moment reconfigure the system to avoid the overload and apply the load curtailment. Thus, this work provided a new line of study and contribute for new researches on this area to go deeper and improve ADMS algorithms.

Keyworlds: ADMS. Self-Healing. Load Shedding. Reinforcement Learning. Q-Learning. Distribution Network. Simultaneous Approach.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ADHDP	–	Action-Dependent Heuristic Dynamic Programming
ADMS	–	Advanced Distribution Management System
ADP	–	Adaptive Dynamic Programming
AMI	–	Advanced Meter Infraestructure
ANEEL	–	Agência Nacional de Energia Elétrica
BA	–	Branch Agent
BESS	–	Battery Energy Storage Solutions
BPSO	–	Binary Particle Swarm Optimization
CEQ	–	Correlated Equilibrium Q-Learning
CIM	–	Common Information Model
CNN	–	Covolutional Neural Network
DEC	–	Equivalent Duration of Interruption per Consumer Unit (DEC);
DER	–	Distributed Energy Resources
DIC	–	Duration of Individual Interruption per Consumer Unit (DIC);
DICRI	–	Duration of Individual Interruption Occurred on a Critical Day per Unit Consumer
DMIC	–	Maximum Continuous Interruption Duration per Consumer Unit (DMIC);
DMS	–	Distribution Management System
DoD	–	Department of Defence
DOE	–	Department of Energy
DP	–	Dynamic Programming
DRL	–	Deep Reinforcement Learning
EA	–	Equipment Agent
EMS	–	Energy Management System
EPRI	–	Electric Power Research Institute
ERAC	–	Regional Load Mitigation Scheme

FA	–	Feeder Agent
FDIR	–	Fault Detection Identification and Restoration
FEC	–	Equivalent Interruption Frequency per Consumer Unit (FEC).
FIC	–	Frequency of individual interruption by Consumer Unit (FIC);
FLISR	–	Fault Location Isolation and Service Restoration
FMQ	–	Frequency Maximum Q-learning
GICUR	–	Government-Industry Collaborative University Research
GIS	–	Geographical Information System
ICCP	–	Inter Control Center Communications Protocol
IEA	–	International Energy Agency
IED	–	Intelligent Electronic Devices
IGA	–	Infinitesimal Gradient Ascent
k-NN	–	k-Nearest Neighbors
LS	–	Load Shedding
MC	–	Monte Carlo
MDP	–	Markov Decision Process
ML	–	Machine Learning
OAL	–	Optimal Adaptive Learning
OMS	–	Outage Management System
ONS	–	National Electric System Operator
OPF	–	Optimal Power Flow
PCA	–	Principal Component Analysis
RL	–	Reinforcement Learning
RTU	–	Remote Terminal Unit
S3VM	–	Semi-Supervised Suport Vector Machines
SA	–	Substation Agent
SAIDI	–	System Average Interruption Duration Index
SAIFI	–	System Average Interruption Frequency Index

SARSA	–	State-Action-Reward-State-Action
SCADA	–	Supervisory Control and Data Acquisition
SG	–	Smart Grid
SH	–	Self-Healing
SL	–	Supervised Learning
SVD	–	Singular Value Decomposition
TD	–	Temporal Difference
TMAE	–	Average Emergency Response Time (TMAE).
TMD	–	Average Displacement Time (TMD);
TME	–	Average Execution Time (TME);
TMP	–	Average Preparation Time (TMP);
UL	–	Unsupervised Learning

LIST OF SYMBOLS

$P^{a}{}_{ss'}$	–	is the transition probability when choose in the current time t an action a in state s that guide to state s' in time $t+1$.
$R^{a}{}_{ss'}$	–	is the expected reward taken when changing the state based on the action chosen.
$E_{\pi}\{\}$	–	the expected value given when the agent follows the policy.
h	–	limit.
E	–	represents the expected result from the equation described inside.
Q	–	value function for a pair variable.
R	–	reward.
T	–	temperature.
$V(s)$	–	value function.
a	–	current action, $a \in A$.
a'	–	future action, $a' \in A$.
$p(a)$	–	transition probability.
r	–	reward function.
$rand$	–	random real values between $[0-1]$.
$randi$	–	random integer values between $[1-\mathrm{sizecolunm(Q)}]$.
s	–	current state, $s \in S$.
s'	–	future state, $s' \in S$.
t	–	time.
α	–	is the learning rate, $0 < \alpha \leq 1$.
γ	–	discount factor, $0 \leq \gamma < 1$.
ε	–	predefined variable between $[0-1]$.
π	–	policy.

SUMMARY

1 INTRODUCTION

1.1 CONTEXT

The energy is an essential resource in human life that affects directly in society, transports, security, life support, and all these points together, having a significant economic impact in a nation. Since years ago, researches tried to count how much the cost impact of an interruption is for the consumers, industrial, commercial, and residential (SULLIVAN et al., 1997; WACKER; BILLINTON, 1989; YAMASHITA et al., 2008). With time, the energy is growing in importance, and nowadays, how much more power can be distributed, more development occurs, factories can keep their production, the commerce can maintain the trading, new ideas can emerge from the house garages, and many other uncountable good things can happen.

Over the last decades the networks have become complex, due to the number of devices installed, the number of consumers, the environmental laws, new regulations, aging of the equipment, among others (MOMOH, 2012). In line to have better developments in power system area, a new concept emerged at the beginning of 21st century called Smart Grid. This concept was given to invest in new technologies to the power system infrastructure to solve the problems occurred in U.S.A. after the wave of blackouts around the country. The most important blackout was in the late 90's and the other in the summer of 2003, due to power system vulnerability that stated a cascade effect of outages (LIU et al., 2018; MA et al., 2018). Thus, some researches in this area began to be developed in order to become the system secure, agile, robust and capable to adapt in unexpecting events, so the idea of Smart Grid arisen. There are many and many definitions for Smart Grid, in summary, it is a system based on new technological applications to control, monitor and manage all the power system, it should be efficient through intelligent techniques to change a passive grid in an active grid, where not only the utility but also the consumer can be part of the system (AMIN, 2005; CECATI et al. 2010; CARVALLO; COOPER, 2011).

In an international context and according to International Energy Agency (2011), there are countries as China, United States, Italy, Japan, South Korea, Spain, Germany, Great Britain and France that already are implementing the system automation to Smart Grids, including some functionalities as intelligent metering and technologies focused in renewable energy. According to Brown (2008), there are other projects in this area, such as, Electric Power Research Institute (EPRI) IntelliGrid, EPRI Advanced Distribution Automation, Modern Grid Initiative by U.S. Department of Energy (DOE), GridWise by DOE, Advance Grid Application Consortium by Concurrent Technologies Corporation, GridWorks by DOE and Distribution Vision 2010. In Brazil, there are some pilot projects already in progress or in order to be implemented, such as the CEMIG future city, Búzios Intelligent city, Smart Grid Light, Eletropaulo Digital Project, InovCity of EDP Bandeirante, Paraná Smart Grid and others (PROJETOS PILOTO NO BRASIL, 2015).

Besides the researches on this area, one possible investment, as a tool to work in a Smart Grid environment is the Advanced Distribution Management System (ADMS), which is a software platform that includes many functionalities to guide the distribution system to be more resilient, in other words, to have the capacity to recover from disaster, for instance, overload, outages, less of voltage and increase of losses, be reliable and efficient in its operations (ENERGY, 2015; DEVANAND et al., 2020; PILO et al., 2009). A complete ADMS is composed of algorithms for automated fault location, fault isolation, service restoration, conservation voltage reduction, peak demand management, volt/var control optimization, microgrid operation, and electric vehicle support.

For each one of these features, there is a unique algorithm that will run independently when the correspondent trigger occurs. In general, it is not possible to reach an optimal global solution, resolving only a unique systemic problem; because the algorithm tries to find the best case to solve an isolated and unique issue. For instance, in the self-healing problem, there are some articles showing techniques to solve it (SUDHAKAR; SRINIVAS, 2011; TORRES et al., 2018), by centralized, decentralized, or distributed systems, using graph theory or artificial intelligence. But, in all cases, they solve an issue per time. Furthermore, other ADMS functionalities can have independent techniques for Volt/Var (SALLES et al.,2016), microgrid (RESENDE et al., 2011), load management (ZHOU et al., 2016), and so on. Beyond the self-healing problem, there is another important trigger of outages, which is the overload, as pointed out in Yamashita et al. (2008), so in this case, there are techniques to mitigate the issues related to the sudden growth of load in the system as showing in (MORTAJI et al., 2017; FERNANDES et al., 2008; XU et al., 2017). A combination of self-healing and load shedding can be seen in Ferreira et al. (2014) and Lin et al. (2011), where more than one problem occurs to be solved using the same technique, but it should address one single trouble first and after another.

As demonstrated in the paper above, there are a lot of intelligent algorithm used to optimize the solution and make the power system better during the operation. As punctuated by Venayagamoorthy (2009), the Computational Intelligence is way of future, the next step of Artificial Intelligence and will be the most potential to evolve the actual network for a Smart Grid solution. Thus, leaded to improve more intelligence during the operation decision, this work shows a new way to act through different triggers (outages and overloads).

1.2 MOTIVATION

Currently, there are two processes to solve the faults into the distribution grid. The first tries a simple service restoration where if a source is available to be used, a set of loads might be transferred to this feeder, and the process ends up at this moment. The second process considers the ADMS functionalities designed to solve each problem per time, i.e., in a sequential way. For example, in a case where the self-healing solution does not have an alternative source to transfer the load without causing an overload, the service restoration will not be executed a priori. However, a second functionality might be triggered to avoid this overload. So, load shedding can cut off part of the load, before the self-healing actions. Then, self-healing is enabled after load shedding.

This second process is more complex, and it will require operational experience or automatic triggering of functionalities to develop it. So, the motivation of this thesis is based on the idea of solving complex problems by several functionalities simultaneously. The problem in this approach is that the search space is increased exponentially, and simple techniques cannot find a solution or solve in a short time. Then, the contribution of this paper is to call on Reinforcement Learning, which is prepared to handle complexity environments.

1.3 OBJECTIVES

The main goal of this thesis project is to develop an intelligent control technique to optimize multiples Smart Grid functionalities, considering, spatial and temporal analysis.

The specific goals include:

a) To evaluate nonlinear, stochastic and time-variant systems;
b) To analyze how it is possible to model the electric power network, aiming Smart Grid functionalities influence on the grid;
c) To develop the intelligent control technique to determine the best result for the multi Smart Grid Functionalities;
d) To realize simulations to validate the technique developed and to be used on the power system model for the integration of several functionalities.

1.4 ORIGINAL CONTRIBUTIONS

The original contribution in this thesis is related with a new and unique method to solve two independent issues that occurs on the distribution network: self-healing and load-shedding. Instead of wait the solution of the self-healing to act in some shed to put the system in a normal operative scenario, the algorithm can calculate at the same time, using one process to find the better solution of reconfiguration and shedding on the network.

The solution for this method lies on the machine learning area, where it’s used the Reinforcement Learning (RL) method to create the action (switching and shedding) and interact with the environment (distribution system). For each iteration it’s verified if the action taken has a good reward (system load, operative constrains and no parallelism) to find the better solution for the current problem.

To work properly the Q-Learning algorithm selected, some adaptations was been considered in the standard process, as the value function to be modeled according the necessity of the distribution system and the reward function, where the reward is guide to minimize the exploration and maximize the exploitation.

1.5 DOCUMENT STRUCTURE

This document is divided in six main chapters, where the first chapter presets the introduction of thesis with the objectives and motivation that guide the author to start the studies to solve the related problem. The second chapter discourse about the state of art, bringing the concepts of ADMS, Self-Healing, Load Shedding, Machine Learning and Reinforcement Learning. The chapter three talks about the bibliographic research, presenting the contribution in Self-Healing, Load Shedding and Reinforcement Learning with Self-Healing areas.

From the chapter four is discussed about the Materials and Methods to develop the algorithm and provide the tests to prove the capacity of the new approach proposed. The chapter five shows the scenarios and results, besides a comparison with other techniques that uses the sequential approach. The last chapter presents the conclusion and proposes the new lines of researches from the implementation discussed in this work.

2 STATE OF THE ART

This chapter brings the initial theoretical concepts about the mainly topics for this work, such as, Advanced Distribution Management System, Self-Healing, Load Shedding, Machine Learning and Reinforcement Learning. The last topic will be open in detail to explain the technique called Q-Learning that was used to conclude the objectives.

2.1 ADVANCED DISTRIBUTION MANAGAMENT SYSTEM

Before to discuss about the Advanced Distribution Management System (ADMS), should be cleared the difference between the Smart Grid (SG) and ADMS. For this work the Smart Grid is a concept to show the problems nowadays in the distribution network, and how these problems can be solutioned and what is the future to be guided (MOMOH, 2009). For example, today the system is based on passive and centralized control, the SG brings the vision to be active and consider a mix of centralized and distributed control (DJAPIC et al., 2007). It should happen because the number of automated devices (sensors, switches, regulators, transformers, capacitor banks), advanced metering infrastructure and more renewable resource penetration installed on the network. These actuators contribute to optimize the power flow and the costs of losses and generation, regulates the voltage, acts on the reactive flows through the capacitor banks, reconfigures the distribution network and manage with stores and distributed generation (PILO et al., 2009). However, the ADMS is the framework to contribute with the SG aims, where, in a simple way, it’s composed by algorithms (functionalities) to process the inputs, find the best answers for the current problem and acts via the outputs. The Smart Grid is ample concept, while the ADMS is one tool to reach the big objective.

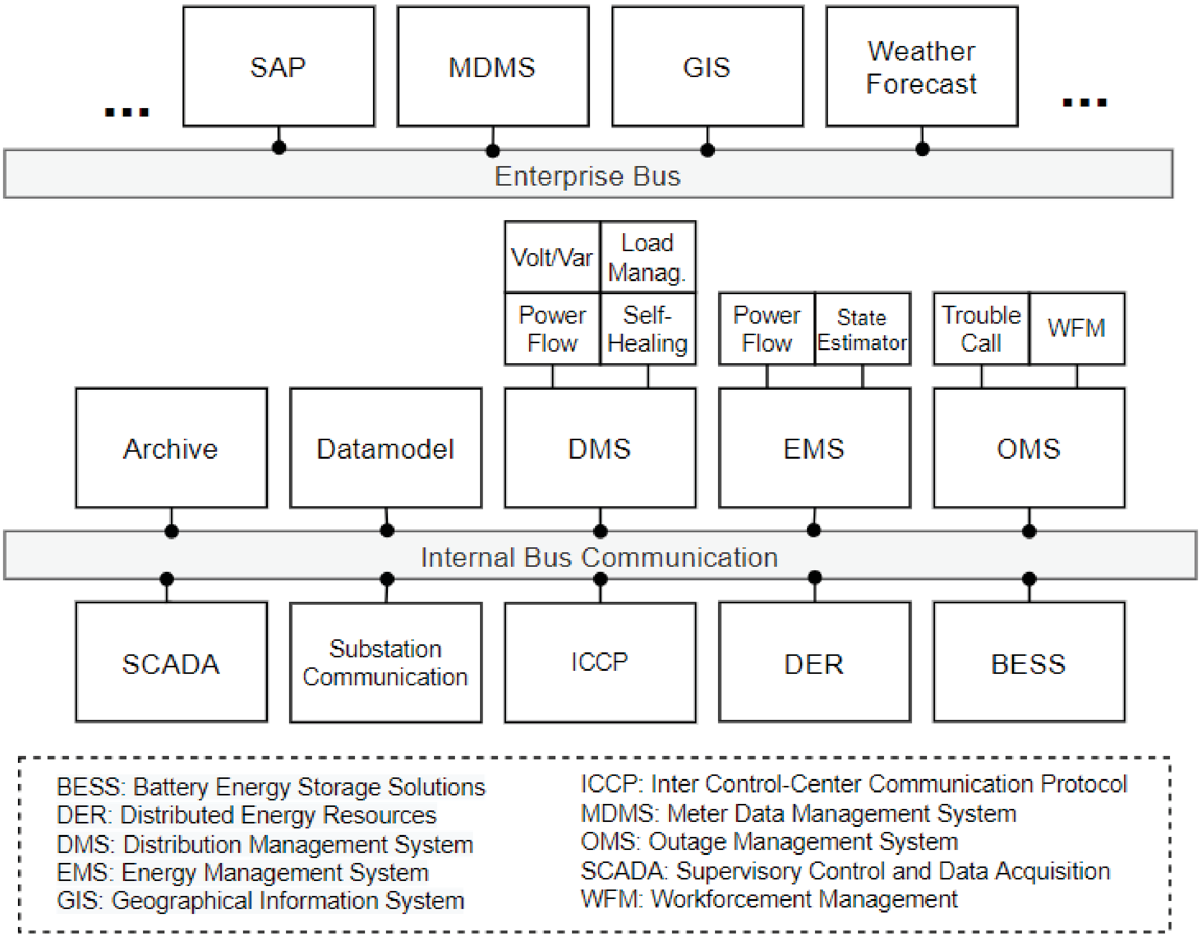
The Advanced Distribution Management System is an integrated platform for Supervisory Control and Data Acquisition (SCADA), once the data is inside there are many advanced applications to create and take decisions (DEVANAND et al., 2020). As shown in FIGURE 1, the ADMS is composed by internal and enterprise bus communication, the former is to change information between the applications and the latter to communicate with application outside of the ADMS to complement with the information provided by the field (e.g., RTUs) and other control centers via Inter-

Control Center Communications Protocol (ICCP) for instance, where these information is processed by the SCADA module to treat and send for the respective application. The application might be:

a) Distribution Management System (DMS): responsible to take the decisions in the distribution side in the Self-Healing, volt/var and load management area. This application can use power flow and other techniques to support better choices;
b) Energy Management System (EMS): responsible to take the decisions on the transmission and generation side. The same way the DMS, they can use many other techniques to support the choices as the optimal power flow and state estimator;
c) Outage Management System (OMS): responsible to manage the outages opened spontaneously by a consumer or an intelligent electronic device, or planned, when the field crew should do any maintenance on the field. This module has the importance to calculate the SAIDI, SAIFI and estimated time for restoration;
d) Distributed Energy Resources (DER): as more and more resources came from the customer or small generation installed in the medium voltage, new techniques should be created to monitor and control these types of resources that can be renewable or not;
e) Battery Energy Storage Solutions (BESSs): responsible to monitor and control the energy store system.

Other important modules should be discussed, one is the Geographical Information System (GIS), where the geographical information used in old system today should be imported in an ADMS for the applications understand the topology. Moreover, the GIS is can be converted, usually, in the Common Information Model (CIM) that is a standard to model the networks components (distribution and transmission) in order to make understandable by programs. The last component to be cited is the archive, where all information from the field and processed by the application are stored for a pre-defined period (small, medium or long term) to be consulted or reprocessed in any study.

FIGURE 1 – ADMS STRUCTURE



SOURCE: Adapted from SIEMENS INDUSTRY INC (2019).

The integration of all modules that compose the ADMS guides the distribution system to be more (U.S. DEPARTMENT OF ENERGY, 2015; BROWN,2008):

a) **resilient**: the capacity to recover from any disaster;
b) **renewable**: to be possible in implement large amount of distributed energy resources;
c) **replacement**: to change the old system in a way to keep with the same functionality;
d) **regulation**: to accept changes that improves reliability and efficiency;
e) **quality improvement**: always provide better energy distribution with less interruption frequency and duration;
f) **resistant to cyber attacks**: the ADMS must communicate with other outside applications and other Intelligent Electronic Devices (IEDs), so it

has ports opened and possible to be accessed remotely or inserted any malware, the ADMS should be prepared to avoid any external interference;

g) **minimizes operations**: old control center software depended all the time of the operator to take actions, and these should take many “clicks” until reach the desire command, so the ADMS provide less “clicks” and more intuitive decision to be done.

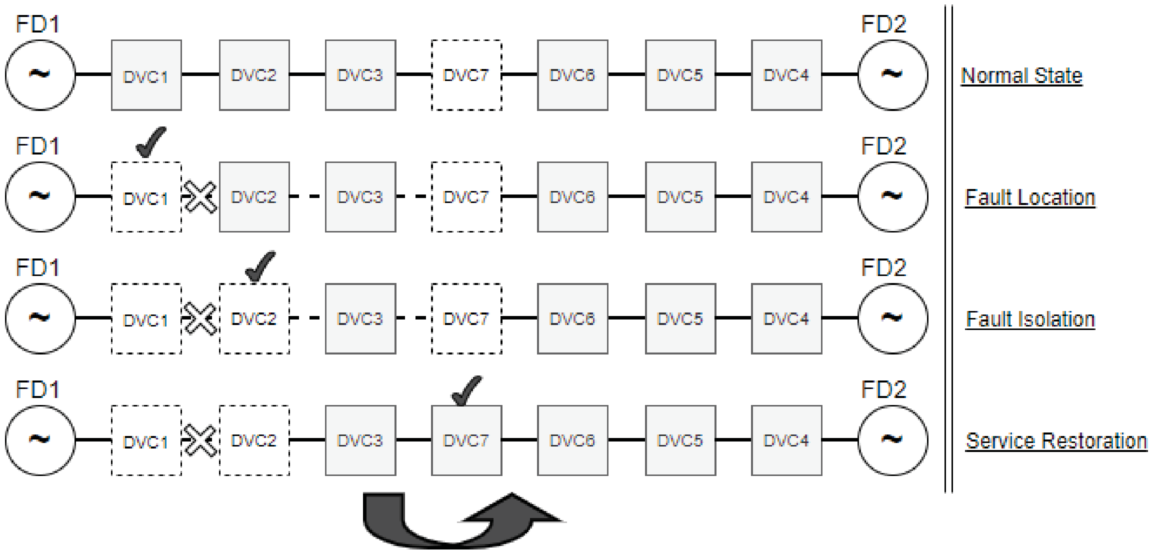
2.2 SELF-HEALING

Algorithms or techniques to solve an outage on the network automatically can have many names, as Self-Healing (SH), Serf-Recovery, Service/System Restoration, Fault Detection Identification and Restoration (FDIR), Fault Location Isolation and Service Restoration (FLISR), where in this thesis will be most used the term Self-Healing. In summary, SH can be determined as a system that consults the topology via the GIS data to understand the feeders and adjacencies. Moreover, the SH utilizes the information from the field and treated by SCADA to obtain the device status and measurements, beside to be triggered when an outage occurs.

All information is processed in any SH algorithm to determine the fault location, which are the switches to isolate the outage and find the switch sequence to restore the out-of-service healthy part of the feeder to another one that could sustain without exceed any limit of load and voltage. The switches can be manually or automatic depending of the strategy determined by the user and the algorithm selected, always thinking to maximize the load restoration and minimizing the number of switching and losses. To find the better restoration solution, some other techniques can be used as the Power Flow to determine what can happen with the new feeder, once some load will be transferred. The process should be fast enough according the energy quality standards provided in each region (OUALMAKRAN et al., 2012; TOUNE et al., 2002).

The FIGURE 2 shows the steps for Self-Healing problem for two feeders and seven switches. In a normal distribution network state, a fault occurs downstream of DVC1, so the first step is to locate the fault and determine the switches to be opened and to isolate (step 2). The last step is to find a normally open switch (DVC7) to close and transfer the de-energized segments to a healthy feeder, noticed that all constraints should be respected for a complete service restoration.

FIGURE 2 – SELF-HEALING STEP-BY-STEP



SOURCE: The author (2020).

The Self-Healing, as understandable in Smart Grid approach, was initially developed as a concept of an intelligent flight control system by the University of Washington with the objective of assisting a pilot response in critical situations that may occur to the airplane as conditions of failure or sudden damage to the airplane (AMIN, 2005). The initial researches for power system was in Amin (1998) through Government-Industry Collaborative University Research (GICUR) program that was contemplated the union of Electric Power Research Institute (EPRI) and the U.S. Department of Defense (DoD) to execute the project. The program aimed to develop new tools and techniques that would encompass large infrastructures to "self-recover" in response to threats, material failures and other destabilizers. Nowadays, there are a list of different techniques to solve the Self-Healing problem, as pointed in Tang et al. (2014).

Studies about service restoration came from the blackout occurred in the past, which originate researches to avoid other blackouts mainly in the transmission network, where it was the first material for studies and sketches of algorithms and process, before the focus start on the distribution network, which is more complex due to number of switches and instability. However, the complexity was not a trammel to start the development in old year, as demonstrated in Aoki et al. (1987, 1989), Clelland et al. (1987) and Liu et al. (1988), where the authors uses more analytical approaches to find the restoration solution, and today the studies is guided by the intelligent algorithms based on the Artificial Intelligence area.

To solve the service restoration problem the strategies might be considered as centralized, decentralized and distributed, according where is executed the logic. The centralized approach converges all information from the field in the same point and all decisions are taken this centralized controller, here the controller must be robust and efficient to support a huge amount of transfer data, for example, ADMS is centralized controller. The other way is the descentralized approach, which the decision maker is presented in a substation retrieving information of the respective devices and changing the information among other substations to determine the better reconfiguration, in this case there is less communication data, but it is not possible to cover all distribution network. The last approach is the distributed, where the logic is presented in the IEDs and take actions based in the communications with other IEDs, the Multi-agent system is one method to be applied in this case (ZIDAN et al., 2017).

Thus, development of techniques to solve the Self-Healing problem either in transmission or in distribution are important to avoid the final consumers to be de-energized by a long time, making the network more resilient, reliable, rapid-recovery, intelligent, and for the utility, they can minimized the losses and penalties caused by an outage.

2.2.1 Brazilian Standards and Quality Index for Self-Healing

Module 8 of the Electricity Distribution Procedures in the Agência Nacional De Energia Elétrica (2012) (ANEEL) refers to the Brazilian electric energy quality, which was created from the Resolution No. 395 in 2009. As presented in Module 8, the verification of the electric energy quality in the distribution network is carried out through service continuity index, and are listed below:

a) Duration of Individual Interruption per Consumer Unit (DIC);
b) Frequency of individual interruption by Consumer Unit (FIC);
c) Maximum Continuous Interruption Duration per Consumer Unit (DMIC);
d) Duration of Individual Interruption Occurred on a Critical Day per Unit Consumer (DICRI);
e) Equivalent Duration of Interruption per Consumer Unit (DEC);

f) Equivalent Interruption Frequency per Consumer Unit (FEC).

And the indicators of time to respond to emergency events:

a) Average Preparation Time (TMP);
b) Average Displacement Time (TMD);
c) Average Execution Time (TME);
d) Average Emergency Response Time (TMAE).

Another important point to be considered for SH and mentioned in Module 8, are the limits of the nominal voltage range, which are established for the voltage profiles greater than 1 kV and less than 69 kV, as shown in Table 1.

Table 1 – VOLTAGE RANGE LIMITS DEFINED BY ANEEL

Service Voltage	**Measure voltage variation range (MV) in reference voltage (RV)**
Adequate	$0.93 \times RV \leq MV \leq 1.05 \times RV$
Precarious	$0.90 \times RV \leq MV < 0.93 \times RV$
Critic	$MV < 0.90 \times RV \; or \; MV > 1.05 \times RV$

SOURCE: Adapted from Agência Nacional De Energia Elétrica (2012).

2.3 LOAD SHEDDING

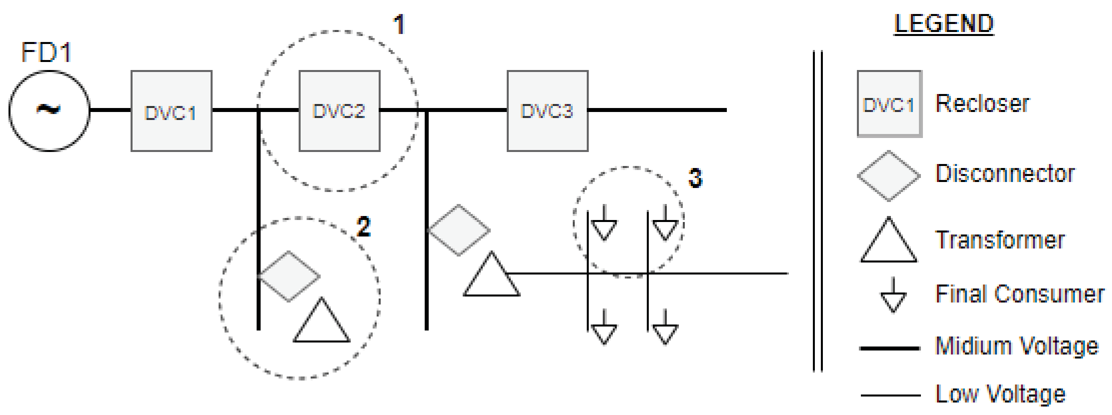
Another point that can be associated with Smart Grid and also to contribute to solve the problem generated by the blackouts is the load shedding technique, where in this case, it is the system alleviating part of the power demand to stabilize the load and generation balance, so it is possible to avoid a big blackout from the unbalance of the system. One of the most common technique used to load shedding is the under-frequency verification, once a disbalance happens, a perceptible variation in the frequency creates a trigger to shed the load (FARANDA et al., 2007).

The preoccupation with the blackout made some programs began to be implemented around the world, as the Regional Load Mitigation Scheme (ERAC, Esquema Regional de Alívio de Carga do Português), which is a standard, in Brazil, from the National Electric Energy Agency (ANEEL) to configure the protection relay to act in underfrequency and curtail parts of the distribution networks according a pre-defined percentual by National Electric System Operator (ONS) for each utility (OPERADOR NACIONAL DO SISTEMA ELÉTRICO, 2009).

In this work the Load Shedding is applied in a complete final consumer which will be curtailed and in cases related with the system restoration, once a simple switching sequence cannot restore a complete block of load, the load curtailment should be applied to try an energization in part of the out-of-service consumers; however, the consumer won't be always the most affected, there are studies where the smart appliances[1] can be turned off in some houses to reach the percentual to be reduced (SIEBERT et al., 2014). Furthermore, in a worst case should be shed some load blocks, which is a cluster of consumers to be out-of-service.

Depending of the technique to be utilized in the load shedding, the load to shut off can be done in three different ways, as shown in FIGURE 3. The first, indicate by the number 1, is a segment shed, where a switch should be opened, and all loads downstream will be de-energized. The number 2, means a shed on the transformer, so a small region will be affected. The last one, number 3, is the case when the consumer has intelligent meter and the curtail can be done in the end of the process, some case even the smart appliances can be turned off to reduce the load.

FIGURE 3 – TYPES OF LOAD SHEDDING



SOURCE: The author (2020).

[1] Intelligent house equipment that can be controlled remotely to be turned on/off when desired.

2.4 MACHINE LEARNING

In the past, Artificial Intelligence (AI) was the trend word and it was a term to include many methods to solve a problem, for example, problems in areas as economics, electrical, robotics, etc., based on an objective optimization.

The mainstream word is the Machine Learning (ML), which is a branch inside of the Artificial Intelligence and tries to create programs to learn based on the experience and information provided previously when possible. According to Shalev-shwartz; Ben-david (2014), a difference between AI and ML is about the learning ("intelligence") process, once the AI is an imitation of the intelligent behavior the other seeks for the positive points that a computer can provide to support the human intelligence.

The ML approach is being increasing in electric power area, where many techniques is solving complex problems and having a different understanding to help the power system be more robust and resilient, once the consumers are more active on the network and more information should be analyzed by the utilities to provide a better energy quality. According to Cheng; Yu (2019), at this moment the world is in the AI 2.0 from the Machine Learning uses in power system, the article provides a good discussion among smart energy, electric power system and Machine Learning.

For this chapter, it was used as basis to describe the Machine Learning concepts the research of Mitchell (1997) and Shalev-shwartz; Ben-david (2014) works. The former presents the base concepts to understand the Machine Learning, such as, statistics, estimation theory, artificial intelligence, information theory and computational learning theory. The latter talks about the key concepts bellow the Machine Learning theory and it presents some main algorithms to be used in this theme.

Besides the books, there are goods blogs that talk about the Machine Learning and can help to introduce some terms, concepts and examples that supports the book information. As a huge area and many information bring up every time, interesting blogs may be used as a complement for the traditional form of reference. In this case, to support the traditional references the website Maini; Sabri (2017) and Brownlee (2013) were consulted to describe some concepts introduced in this work.

Basically, Machine Learning (ML) is not a simple algorithm and is not a unique concept, the ML comprehends many other disciplines as showed in FIGURE 4 and there are several algorithms used to solve the problem, FIGURE 5. Furthermore, each algorithm has the purpose to manipulate the data input to obtain the correspondent result. These algorithms are not a set of simple rules, they are programmed to take actions and decisions based on the data input, and so they try to learn and to work dynamically over the time.

FIGURE 4 - MACHINE LEARNING AND THEIR DISCIPLINES

Artificial Intelligence		
Machine Learning		
DATA SCIENCE	STATISTICAL	GRAPH
PROBABILITY	CONTROL THEORY	PATTERN RECOGNATION
NEURAL NETWORK	DATA MINING	COMPUTATIONAL LEARNING

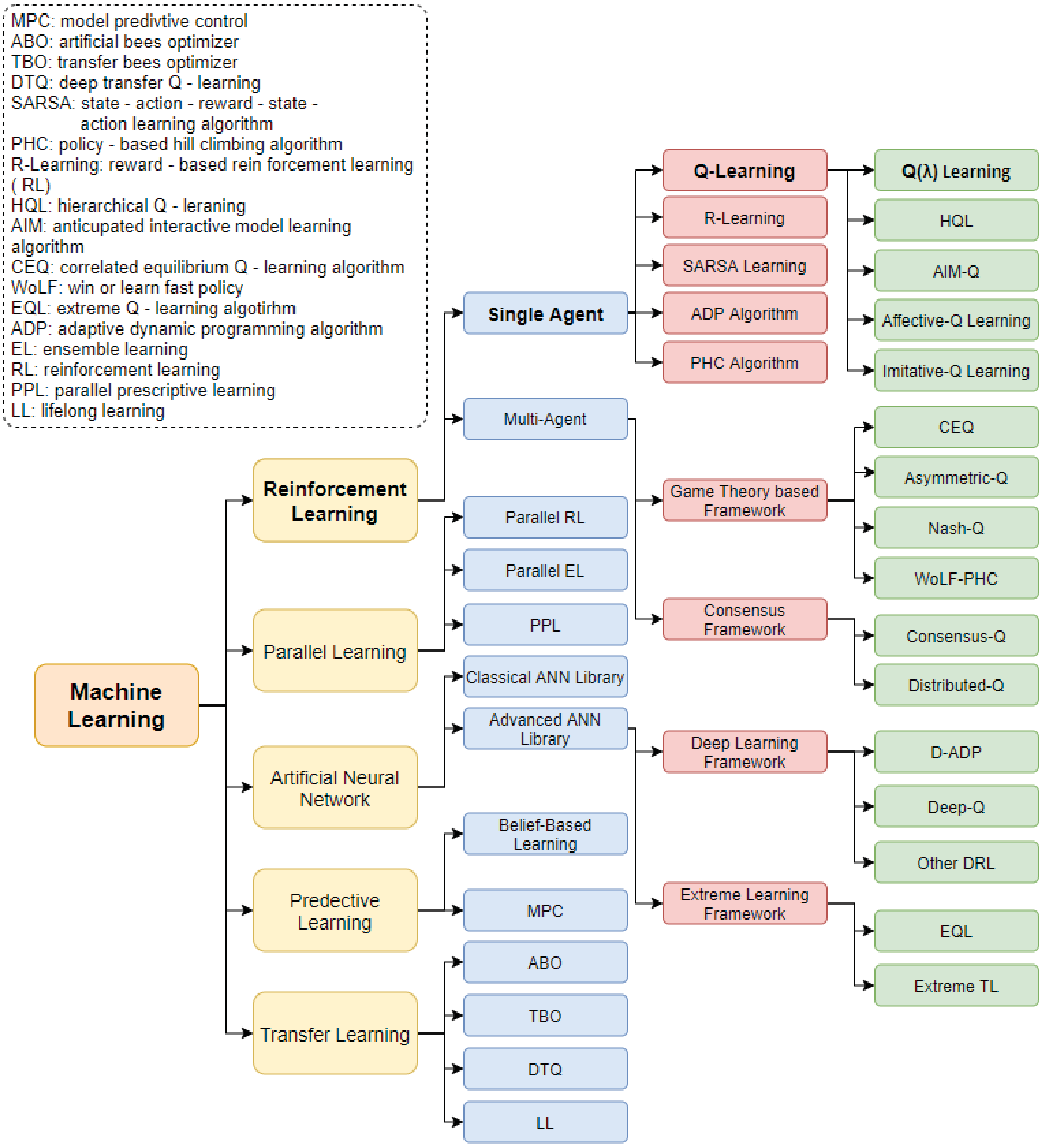
SOURCE: The author (2020).

In ML the choice of the best algorithm is fundamental for the best results or the possible mix of algorithms can solve the problem more efficiently than a simple algorithm, this involves in the first moment the data analysis to comprehend the type of problem and what is the expected result. In addition, other important part is the training because the learning process to obtain a correct answer, considers a set of data applied to adjust the algorithm (equations and parameters).

Machine Learning may be applied in diverse areas and in distinct problems, such as, to control robots, autonomous cars and power generation, to classify images text and e-mails, to identify credit card violations, weather prediction, to analyze the genomic data and to convert medical archives into medical knowledge.

For control robotics some examples can be seen in Kouppas et al. (2018), where the paper brings the comparison for different machine learning techniques to control a bipedal robot from the interferences of electric motors and actuators. The author, in Nandi (2018), came with the idea to use the Deep Reinforcement Learning (DRL) and Convolutional Neural Network (CNN) to able the robot to grasp objects. Another application with the DRL is shown in Taitler; Shimkin (2017) to control a robot in the air hockey game.

FIGURE 5 - MACHINE LEARNING ALGORITHMS



SOURCE: Cheng; Yu (2019).

About the autonomous vehicle it’s possible to see in Boroujeni et al. (2018), Thammachantuek et al. (2018) and Roncancio et al. (2013) use of vector approach to see images, like pedestrian, on the road and decide the vehicle position. In Olgun (2018), is applied the DRL to control the car to do some tasks as tracking the lanes, following vehicle and stop in some abnormal conditions.

In power system area some article can be seen in forecast generation in Akhter et al. (2019)is done a review of machine learning techniques and other metaheuristics to compare the forecast of photovoltaic power based on several forecast horizon. The other paper Fleming (2019) brings the prediction for seasonal water in US West about the arid region, the author shows a comparison with eight different methods. For wind generation, three different algorithms based on machine learning and also an ensemble with the techniques can be verified in Du (2019).

Besides the forecast, another case of study in power system with machine learning is related with the microgrid. In Alam et al. (2017) is applied the SVM to classify the island detection, the same approach is presented in Mishra; Rout (2017), where a combination of techniques (Hilbert-Huang transform and Machine Learning) to classify the fault events.

In the subsection is demonstrated how the Machine Learning can be classified based on the learning styles: supervised, unsupervised, semi-supervised learning and reinforcement learning, this classification is the most common used on the literature. According to Shalev-shwartz; Ben-david (2014), in addition to the classification cited, they classify the Machine Learning as Active versus Passive Learners, Helpfulness of the Teacher or Online versus Batch Learning.

2.4.1 Classification by Learning Styles

In this section is discussed about the existents learning styles inside of machine learning.

2.4.1.1 Supervised Learning (SL)

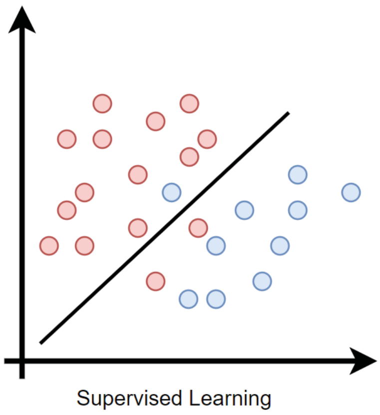
The concept of supervised learning are algorithms that need a set of data to training the model and it is based on applied rules. This training improves the “intelligence” of the model to determine the correct output, for instance a prediction.

Moreover, this data must have a label for each information because this label helps to understand what the correct result from the input data.

To apply this approach a good dataset should be provided in a way to prepare a properly training of the algorithm. In some cases, should be necessary to use all possible data, "brute-force", to training (MUHAMMAD; YAN, 2015). As expected, this type of data and analysis can have noises or incomplete information, so other techniques should be applied to correct and have better results on the model (HODGE; AUSTIN, 2004).

One example to understand the supervised learning is in FIGURE 6 based on regression type, where can be defined the better division to split the two model of data that is similar between them.

FIGURE 6 – GRAPHIC EXAMPLE OF SUPERVISED LEARNING



SOURCE: The author (2020).

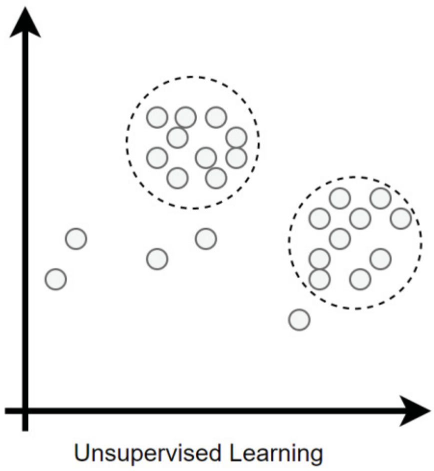
The supervised learning may be divided in classification and regression types, and below some algorithms based on this classification:

- Linear Regression;
- Ordinary Least Squares Regression;
- Backpropagation Neural Network;
- Decision Trees;
- Support Vector Machines;
- k-Nearest Neighbors (k-NN);
- Bayesian.

2.4.1.2 Unsupervised Learning (UL)

In unsupervised learning there is no distinction between the training and test data, the reason is because the objective is based on a massive data input, which helps the algorithm to find an output of subsets that represents one set of data. This data should be the same type of information and will represent something more useful to be understood, i.e., to find clusters of data in a mix of different information, where is not possible to be done by human hand. For this type of learning, there isn't labeled data to comprehend what is the correct output after the training (GHAHRAMANI, 2003). The FIGURE 7 is one example of data cluster with similar behavior.

FIGURE 7 – GRAPHIC EXAMPLE OF UNSUPERVISED LEARNING



SOURCE: The author (2020).

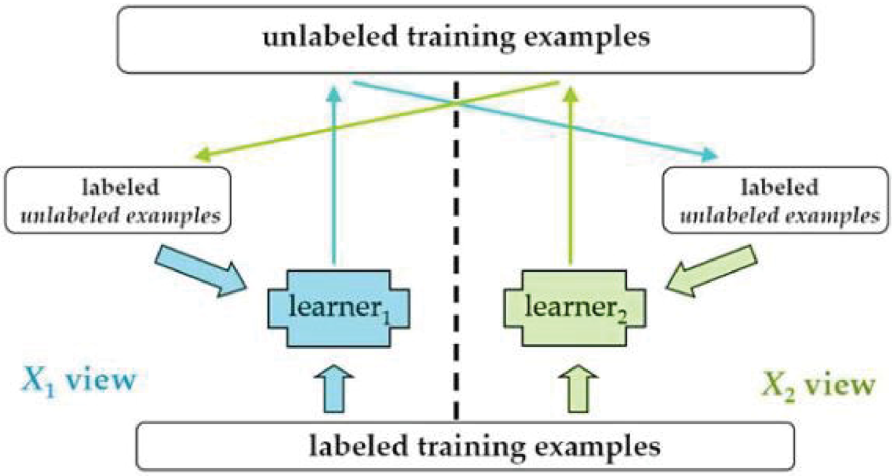
The unsupervised learning is a descriptive model and can be divided in summarization, association and clusterization. Following some algorithms used for unsupervised learning:

- Apriori;
- k-Means;
- Hierarchical Clustering;
- Principal Component Analysis (PCA);
- Singular Value Decomposition (SVD);
- Mixtures of Gaussians.

2.4.1.3 Semi-supervised Learning

The semi-supervised learning has the same objective of classification, as the supervised model. However, in this case, the dataset is a mix of labeled data and unlabeled to train and create better models. The unlabeled data has the meaning of modify or reprioritize hypotheses earned from the labeled data (ZHU, 2005). As explained in FIGURE 8, there is the learner based on the labeled data and the unlabeled data influencing the learning to improving the model.

FIGURE 8 – PROCESS EXAMPLE OF SEMI-SUPERVISED LEARNING



SOURCE: Zhou; Li (2010).

In some application the semi-supervised learning is useful because the labeled data is expansive, so it is necessary to spend more space to store the information instead of the data for unsupervised methods. Possible algorithms that works on this mode are (ZHOU; LI, 2010):

- Generative Methods: Expectation-Maximization;
- Semi-Supervised Suport Vector Machines (S3VMs);
- Graph-Based Methods;
- Disagreement-Based Methods.

2.4.1.4 Reinforcement Learning (RL)

This learning is different from the other learning models because in this case, the **agent** finds the correct output through the trial and error (**actions**), for each action an interaction is executed in the **environment** and it is calculated a reward that will represent whether the action taken was positive or not. The Reinforcement Learning is composed by three important components:

a) **agent**: is the decision maker;
b) **environment**: is where the agent interacts;
c) **actions**: are the possibilities that the agent can do.

In Kaelbling et al. (1996) are listed some important points that can explicit separate the Reinforcement Learning and Supervised Learning Methods based on their modeling and application. This work brings up some important points as the main differences that there is no correlation between the input and output; the on-line performance for RL, where is fundamental. The supervised methods have interest in having a future predictive accuracy or statistical efficiency applied in its algorithms different of the RL and another point is the RL execute a better exploration of the environment, where the supervised cannot process in the same way.

For this work to solve the Self-Healing problem was used the Reinforcement Learning, especially the Q-Learning algorithm, to take the decision and find the best switching action and load shedding, simultaneously, to restore the distribution system with the maximum load possible. A better explanation about the reinforcement learning is demonstrated in the next chapters.

2.5 REINFORCEMENT LEARNING

This section provides an introduction about the Reinforcement Learning technique and the algorithm used to solve the problem presented in the first part. The principal reference to describe RL was Sutton; Barto (1998), which is an introductory book to understand how the reinforcement learning born and grow up beyond the years.

A useful survey is Kaelbling et al. (1996), this paper was used to understand some approaches in reinforcement learning and also to comprehend some points about the history until to reach the final understanding that is known in Reinforcement Learning concept.

Another important reference used to understand the concepts and other references was Khan et al. (2012). This article brings the initial authors about this area, complement with applications to enhance the reinforcement learning and finishes the article with an application in robotics.

2.5.1 A Briefly History

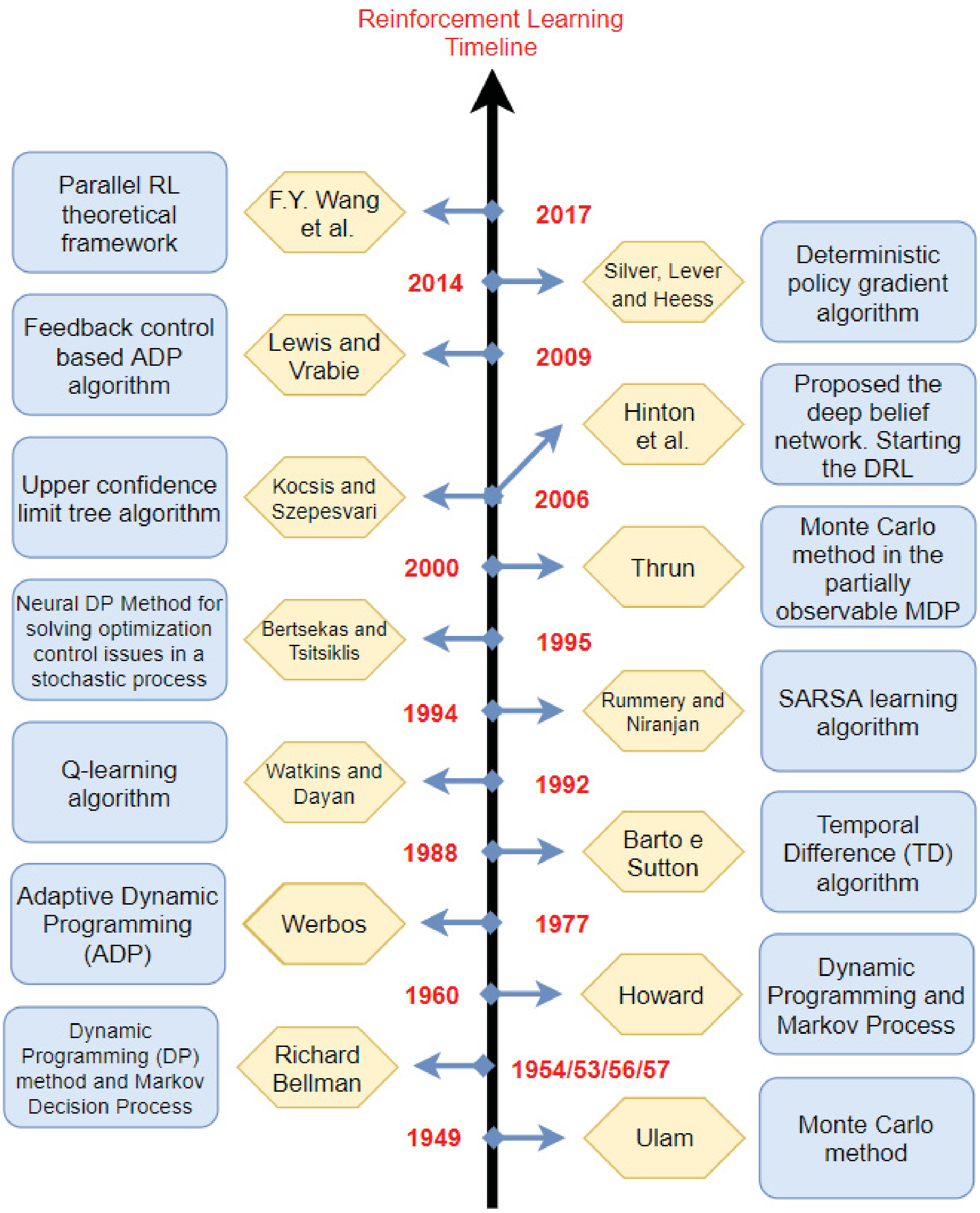
Reinforcement Learning is a technique which was born around 80s with diverse types of research until be called a reinforcement learning. According to Kaelbling et al. (1996), Sutton; Barto (1998) and Cheng; Yu (2019) there are some important references in the during the RL construction method, such as, Richard Bellman with the Dynamic Programming and Markov Decision Process in 1950s (BELLMAN, 1953, 1954, 1956, 1957). Howard (1960) worked with the policy iteration in 1960, beside this the book brings a compiled about the dynamic programming and Markov Decision process. In Werbos (1977), the author demonstrated method called Adaptive Dynamic Programming (ADP). Figure 9 easier shows the RL timeline.

In the 1970s and 1980s, Barto and Sutton contributed with a lot of researches on this area to improve the reinforcement learning as known today, one important reference is Sutton (1988) about the Temporal Difference algorithm. In the early 90s, after the TD algorithm definition, Watkins and Dayan proposed the off-policy Q-Learning approach (WATKINS; DAYAN, 1992). Some years latter Rummery and Niranjan created the on-policy method called State-Action-Reward-State-Action (SARSA) Learning (RUMMERY; NIRANJAN, 1994). Keeping in this decade, Bertsekas (1995) create the neural dynamic programming to solve optimization in control issues through stochastic process.

In 2000s, Thrun (2000) shown a Monte Carlo Method in a partially observable Markov Decision process. The Deep Reinforcement Learning start the studies in 2006 with Hinton et al. (2006), in the same year Kocsis; Szepesv (2006) developed the upper confidence limit tree algorithm. The last relevant studies to be related here are Silver et al. (2014) and Li et al. (2017), where the former talk about a deterministic policy gradient algorithm and the latter about a parallel RL theoretical framework.

Some main concepts should be understood first for a complete comprehension of the Reinforcement Learning, these concepts are presented in the next subsections and comprehend the elements ("the language") that compose the reinforcement learning. After this, there is a discussion about the finite Markov Decision Process, which is the base for a RL working, the methods to solve RL, such as, Dynamic Programming, Monte Carlo and Temporal Difference. Inside of Dynamic Programming there are two important formulation, value and policy iteration, that are used to model the Q-Learning algorithm, where this algorithm is the final concept explained in this section.

Figure 9 - The Reinforcement Learning evolution diagram



SOURCE: adapted from Cheng; Yu (2019).

2.5.2 Fundamental Elements of Reinforcement Learning

To have a better understanding about the Reinforcement Learning (RL) technique is there are some fundamentals concepts to be introduced, such as, the policy, environment model, value function and reward. As a learning algorithm, it should handle with two distinct objectives, the exploration and the exploitation. The former would like to explore complete the environment, but this spend time to execute. The latter tries to find the minimize the costs, i.e., tries to find the better location to take advantage about that was learned to determine the action. The trade-off between them brings the maximization of the learning effected, where can learn in less time with less effort (THRUN, 1992).

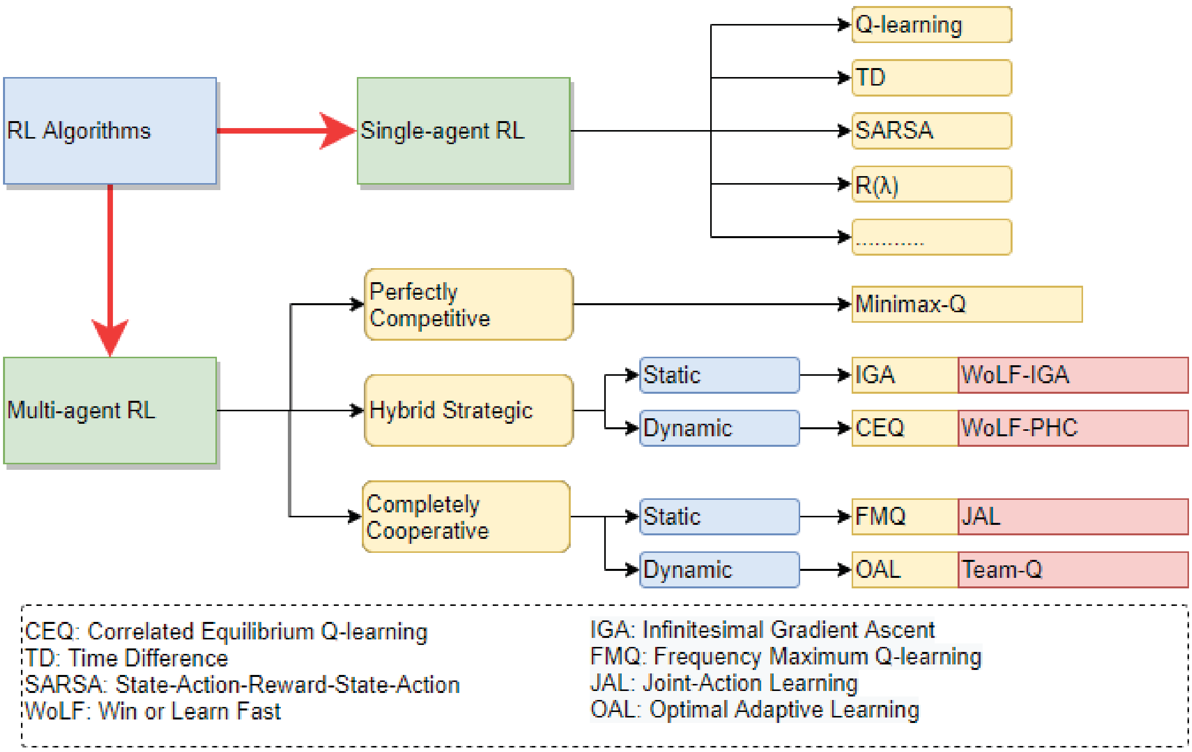
According to Kaelbling et al. (1996) there are two main strategies to solve the reinforcement learning problems: explore the search space to find the best result for the correspondent environment and the second is to use statistical and dynamic programming to estimate which action is the best inside of the complete environment.

Basically, the Reinforcement Learning is composed by three sets:

a) A discrete set of environment states, $s \in S$;
b) A discrete set of agent actions, $a \in A$;
c) A set of scalar reinforcement signals; $r \in R$.

Another important aspect in the Reinforcement Learning, where the algorithm can be divided in the number of agents, the single-agent with the traditional methods (Q-Learning, SARSA, TD) and the new techniques based on the multi-agents (IGA, CEQ, FMQ, OAL) as demonstrated in the FIGURE 10.

FIGURE 10 - THE CLASSIFICATION OF REINFORCEMENT LEARNING



SOURCE: Cheng; Yu (2019).

2.5.2.1 Model of Optimal Behavior

When it's started the algorithms about learning, should understand about the models of optimally, where according to Kaelbling et al. (1996) there are three major relevant types in this study. The first one is the finite-horizon model. This model (1) just cares about the limit (h) to calculate the reward (r) optimization in the time (t), so in this case it's not possible to explore more than its limit.

The second model, (2), is the infinite-horizon discounted model, where there isn't the time limit and a new factor (γ) is included to geometrically discount the reward on the future values. The last model, (3), commented is the average-reward model, where the action to take better optimization comes from a long-run average reward. As an average, the application of two different policies (explained in the next chapter), one applied to find the best result initially and the other for long-run will be mixed as a unique result the first policy will be masked at the end.

$$E(\sum_{t=0}^{h} r_t) \quad (1) \qquad E(\sum_{t=0}^{\infty} \gamma^t r_t) \quad (2) \qquad \lim_{h\to\infty} E(\frac{1}{h}\sum_{t=0}^{h} r_t) \quad (3)$$

where:

E – represents the expected result from the equation described inside.

t – time.

r – reward.

h – limit.

γ – discount factor.

2.5.2.2 Policy

The objective of a policy is to maximize the total reward over the process iteration. It is comprehended as a mapping of probability transition of states from an action selected, $\pi(s, a)$.

There are two classifications for **policies**, on-policy or off policy. The on-policy has dependence between the policy value and the value for control, the same estimations for a policy is used to control the system. One example of on-policy is the SARSA (State – Action – Reward – Start – Action) algorithm (RUMMERY; NIRANJAN, 1994).

In the meanwhile, the off-policy has the estimation policy and the control value separate, and according Sutton; Barto (1998), it comes up with two new concepts: the behavior policy, which is to generate the system behavior and can retrieve all possible action; while, the other is deterministic and can be called as estimation policy, and is not linked with the policy which is evaluated and improved. In the other words, the idea is to estimate the value function using hypothetical actions that have not been tested. One example of off-policy is the Q-Learning method.

Comparing the equations below can see the difference between an on-policy approach, SARSA equation (4), and an off-policy, Q-Learning equation (5). The equation (4) utilizes the future action (a') to calculate the value function (Q), i.e., the policy influences directly the final equation, instead of equation (5) that utilizes the action selected in current iteration.

$$Q(s,a) = r(s,a) + \gamma max_a Q(s', \boldsymbol{a'}) \tag{4}$$

$$Q(s,a) = r(s,a) + \gamma max_a Q(s', a) \tag{5}$$

where:

Q – value function for a pair variable.

r – reward function.

max – maximum value of Q for the current state when varying the actions.

s – current state, $s \in S$.

s' – future state, $s' \in S$.

a – current action, $a \in A$.

a' – future action, $a' \in A$.

γ – discount factor, $0 \leq \gamma < 1$.

Some policies functions are discussed next, the **Greedy technique**, equation (6), tries to choose the best action based on the best payoff from the value function. The idea of the greed algorithm is simple, and its processing is faster than compared with other methods. In some cases, the greed algorithm can be scant, i.e., it can find in the first moment the best action to follow, but it is not safe because the first choice cannot lead to better final solution.

$$\pi'(s) = argmax(Q^{\pi}(s,a)) \tag{6}$$

A variation of the greed approach is the **ε-greedy technique**, equation (7), which include a random variable ε to improve the learning exploration instead of all iteration only the exploit side. In place to retrieve the maximum value of Q-Matrix in each iteration (exploit) to go deeper in the action selected, the ε-greed tries to avoid a possible wrong decision selected in the past to choose a new action (explore) and verify if the new decision is better or not.

$$\begin{cases} rand > \varepsilon, & \pi^{'}(s) = argmax\big(Q^{\pi}(s,a)\big) \\ rand \leq \varepsilon, & \pi^{'}(s) = Q^{\pi}(s, randi) \end{cases} \tag{7}$$

where,

ε – predefined variable between $[0 - 1]$.

$argmax$ – obtain the better action (index) for the current state in the Q matrix.

$rand$ – random real values between $[0 - 1]$.

$randi$ – random integer values between $[1 - \mathrm{sizecolunm(Q)}]$.

The **randomized technique** (simulated annealing), equation (8), picks up an action random from a distribution function (8). In this case the objective is to explore in a larger area in the initial iteration based on the temperature T and over the time, T should be decreased to reduce the exploration area. For a better working, the system should demonstrate the best action less related with the other options.

$$P(a,s) = \frac{e^{\frac{Q(a,s)}{T}}}{\sum_{a' \in A} e^{\frac{Q(a',s)}{T}}} \tag{8}$$

where:

$P(a)$ – transition probability.

Q – value function for a pair variable.

s – current state, $s \in S$.

a – current action, $a \in A$.

a' – future action, $a' \in A$.

T – Temperature.

The **Interval-based techniques** is recommended in empirical trial and aims to save a statistical value for each action and after choosing the best action based on the highest upper bound according to the equation (9). One example is the Kaelbling's interval estimation algorithm (Kaelbling (1990)).

$$100 \times (1 - \alpha)\% \tag{9}$$

where:

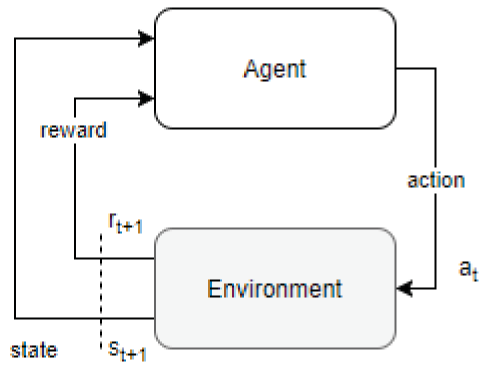
α – discount factor.

2.5.2.3 Agent and Environment Model

The first two concepts to be understood in RL and very important for the rest of the comprehension is the Agent and the Environment. The **Agent** has the objective to find the best policy π to maximize the reward function, it is the learning and decision-maker. The **Environment** is the rest outside of the agent and the aim is to answer when the agent interacts through the actions in the environment.

The interaction between these two elements is demonstrated in FIGURE 11, which each action that the agent selects, is interacted through the environment, the result is a new state and a new reward value representing as a next time step, so the agent process this new input to select a new action and repeat the process, but now with the learning from the previous iteration.

FIGURE 11 – REINFORCEMENT LEARNING FLOWCHART



SOURCE: Sutton; Barto (1998).

2.5.2.4 Value Function

The **value function** is a pair composed by the states and action, $Q(s, a)$, or just states, $V(s)$. These types of value function are related with the environment model, that, when an environment model is known, it is used the simple value function.

Another important notation is the relation between policy and value function, where the value function means the return value when starting in s following the policy π. In this case, the simple value function is $V^{\pi}(s)$, and for the pair is $Q^{\pi}(s, a)$, the formally equation for each function is presented in equation (10) and (11) respectively.

$$V^{\pi}(s) = E_{\pi}\{R_t | s_t = s\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \, | s_t = s\} \quad (10)$$

$$Q^{\pi}(s,a) = E_{\pi}\{R_t | s_t = s,\ a_t = a\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \, | s_t = s, a_t = a\} \quad (11)$$

where:

V – value function for a single variable.

Q – value function for a pair variable.

π – policy applied.

$E_{\pi}\{\}$ – the expected value given when the agent follows the policy.

R – reward.

s – current state, $s \in S$.

a – current action, $a \in A$.

γ – discount factor, $0 \leq \gamma < 1$.

t – time step.

The value function helps to choose the best action to generate a higher reward value. Then, how better is possible to estimate the value function, a better action can be chosen and consequently a better reward for the final state can be determined.

2.5.2.5 Reward

A reward can be described how the agent is performing when selects its actions to be interacted with the environment. This metric forces the agent to learn, once the reward is prepared to respond when the agent carries out a task successfully or penalizes the reward when the agent executes an action which results in a tough situation.

2.5.3 Markov Decision Process

Another important definition for RL is about Markov Decision Process (MDP), where a RL can be called finite MDP when its states and actions are finites, and this definition is very important for the reinforcement learning functionality.

The MDP was created by Bellman in 1957 and correspond a discrete stochastic way to describe the optimal control problem (BELLMAN, 1957). The MDP models problems with delayed reward, which represents a sequence of action where the reinforcement value is increasing according to this sequence to find the highest value of reinforcement. Another key point about MDP is that the environment should understand the past without interfering the next steps.

There are two important equations that defines the dynamics for finite Markov Decision Process, the transition probability (12) and next reward value (13):

$$P^{a}{}_{ss'} = P_r\{s_{t+1} = s' | s_t = s, a_t = a\} \tag{12}$$

$$R^{a}{}_{ss'} = E\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\} \tag{13}$$

The problem to be solve is to find the best policy that maximize the cumulative sum of all rewards calculate in an infinite horizon, the problem equation is presented in (14).

$$\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1}) \tag{14}$$

The algorithm that solves the problem before is divided in two parts, the first one is to find the policy for the current state, which represents the best action in the moment, equation (15). The equation (16) means the value function and contains the discounted sum of the rewards from the result of equation (15).

$$\pi(s) \coloneqq argmax_a\{\sum_{s'} P_a(s, s')\big(R_a(s, s') + \gamma V(s')\big)\} \tag{15}$$

$$V(s) \coloneqq \sum_{s'} P_{\pi(s)}(s, s')(R_{\pi(s)}(s, s') + \gamma V(s')) \tag{16}$$

Where:

$P^{a}{}_{ss'}$ – is the transition probability when choose in the current time t an action a in state s that guide to state s' in time $t+1$.

$R^{a}{}_{ss'}$ – is the expected reward taken when changing the state based on the action chosen.

E – represents the expected result from the equation described inside.

s – current state, $s \in S$, and s' is the next state.

a – current action, $a \in A$, and $a^{t} = \pi(s_t)$.

γ – discount factor, $0 \leq \gamma < 1$.

t – time step.

$V(s)$ – value function.

π – policy.

There are two modes in Dynamic Programming to solve the problem using the equations described above, one is the policy iteration and the other is value iteration, for more details go to *2.5.4.1 Dynamic Programming* chapter.

2.5.4 Methods to solve the Reinforcement Learning

To solve the problems in Reinforcement Learning is briefly presented next three possible methods: Dynamic Programming, Monte Carlo and Temporal Difference.

2.5.4.1 Dynamic Programming (DP)

The idea of Dynamic Programming (DP), basically, is to solve complex problems breaking in smaller problems, easier to solve, to find the final solution for the complex equation. Moreover, when the DP is being performed, each solution is stored to be used in a future iteration when necessary, so the method can estimate the current value using past estimations.

Bellman develops the introduction for DP in Bellman (1953) and formalized the method in Bellman (1954), which brings the Principle of Optimality to solve the problems in DP, *"an optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with the regard to the state resulting from the first decisions"*.

One of the problems for the DP method is the "curse of dimensionality" where the number of states can be exponentially bigger and computations process cannot execute successfully.

Below is described two most popular methods in Dynamic Programming, the policy iteration and the value iteration.

POLICY ITERATION

Due to in RL there is the property about finite Markov Decision Process, it is possible to converge the policy inside of this constrained space. The idea of policy iteration is presented by the pseudocode #1, which presents two parts to solve the problem. The first part is the policy evaluation, where is determined in each state the value function based on the previous policy, and a new delta is calculated to verify whether the new delta is smaller than the pre-defined value θ.

Then, the second part is the policy improvement, which correspond the part to determine the new policy in the case the problem was not solved. The policy is determined using the value function and when the policy is the same that the previous solution, the algorithm converged, in other case it is not stable yet and another round to find the policy evaluation should be executed.

One disadvantage of policy iteration is the number of iterations to determine the policy evaluation, which depending of the problem, it can be necessary a long processing to find the result.

Pseudocode #1 Policy Iteration

Initialize: $V(s) \in \Re$ and $\pi(s) \in A(s)$

Policy Evaluation:

  Repeat

    $\Delta \leftarrow 0$

    For each $s \in S$:

      $v \leftarrow V(s)$

      $V(s) \leftarrow \sum_{s'} P^{a}{}_{ss'}[R^{a}_{ss'} + \gamma V(s')]$

      $\Delta \leftarrow \max(\Delta, |v - V(s)|)$

  Until $\Delta < \theta$

Policy Improvement

  $policy\_stable \leftarrow true$

  For each $s \in S$

    $b \leftarrow \pi(s)$

    $\pi(s) \leftarrow argmax_{a} \sum_{s'} P^{a}{}_{ss'}[R^{a}_{ss'} + \gamma V(s')]$

    IF $b \neq \pi(s)$ **THEN**

      $policy\_stable \leftarrow false$

  IF $policy\_stable$ **THEN**

    stop.

  ELSE

    GO TO Policy Evaluation

SOURCE: Sutton; Barto (1998).

VALUE ITERATION

The value iteration tries to solve the problem about the policy evaluation, where in this case, the idea is to break the policy evaluation after one sweep. The new value function equation is demonstrated in (17) and represents a combination of policy improvement and truncate the policy evaluation steps the find the value function.

$$\begin{aligned} V_{k+1}(s) &= \max_{a} E\{r_{t+1} + \gamma V_k(s_{t+1}) | s_t = a, a_t = a\} \\ &= \max_{a} \sum_{s'} P^{a}{}_{ss'}[R^{a}_{ss'} + \gamma V(s')] \end{aligned} \tag{17}$$

Pseudocode #2 Value Iteration

Initialize: V arbitrarily
Repeat
  $\Delta \leftarrow 0$
  For each $s \in S$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \max_{a} \sum_{s'} P^{a}{}_{ss'}[R^{a}_{ss'} + \gamma V(s')]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
Until $\Delta < \theta$

$\pi(s) = argmax_{a} \sum_{s'} P^{a}{}_{ss'}[R^{a}_{ss'} + \gamma V(s')]$

SOURCE: Sutton; Barto (1998).

The value iteration algorithm is presented in Pseudocode #2, and in one sweep can be determined the policy evaluation and policy iteration at the same time, according to the equation (17). The algorithm just stops when the variation between the initial value function with the new is less than the pre-defined value θ.

2.5.4.2 Monte Carlo (MC)

Monte Carlo (MC) methods was initially thought in 40s, when the nuclear bomb was being created. A formal definition for MC methods is presented in Metropolis; Ulam (1949).

In summary, the Monte Carlo methods is different of the DP methods described above, the MC methods don't learn from a model because it can learn using samples to understand the problem, in each iteration, random samples, based on a probability distribution, are used to execute a deterministic simulation with the inputs to give a final result.

Furthermore, another significant difference between Dynamic Programming is MC doesn't change the current estimation based in other estimations from previously iteration, i.e., it doesn't bootstrap.

2.5.4.3 Temporal Difference Learning (TD)

Temporal Difference Learning (TD) is a model-free[2] and the objective is to use insights from the value iteration to adjust the estimated value of a state, based on the immediate reward and the estimated value of the next state. For a complete comprehension about the temporal difference method, in Sutton (1988), brings the first formal definition about the theme.

Basically, TD is a mix of Monte Carlo and Dynamic Programming concepts, where the TD retrieves from MC the objective of learning from raw experience, i.e., it is not necessary to have an environment model because MC Learning from the samples. Moreover, the DP part is important for TD because to approximate the current value is necessary to use, in this case, the information from the past.

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \quad (18)$$

Another key point to understand TD is the $TD(0)$ algorithm, which was the base for the Q-Learning method. The algorithm is demonstrated in pseudocode #3 and it is similar with the pseudocode #4 about the Q-Learning. This algorithm is a specific case of $TD(\lambda)$, but just analyzing one step ahead when estimating the equation (18), which is the rule for $TD(0)$, notice that the equation just considers one position in the future ($t + 1$).

[2] Model-Free means to learn a controller without learning a model (the complete environment) before start the learning process. (Kaelbling et al., 1996)

Pseudocode #3 $TD(0)$ algorithm

Initialize: $V(s)$ arbitrarily, π to the policy to be evaluated
Repeat for each episode:
  Initalize s (state)
  Repeat (for each step of the iteration)
    $a \leftarrow$ action given by π for s
    Take an action a
    Observe reward, r, and next state, s'
    $V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$
    $s \leftarrow s'$
  until s is terminal

SOURCE: Sutton; Barto (1998).

2.5.5 Q-Learning

The Q-leaning algorithm was originated in a Ph.D thesis by Christopher J. C. H. Watkins, Watkins (1989). The article from Watkins; Dayan (1992) was used for a better understanding because this paper brings the mainly points to comprehend the Q-leaning and also to prove the method presented in Watkins (1989), it is possible to converge.

In Kaelbling et al. (1996) is presented the definition summed up and with the focus on Reinforcement Learning, this paper brings comparison with other similar techniques and the essential to understand the Q-Learning definition.

Q-Learning is classified as model-free and belongs to off-policy Temporal Difference Methods which was explained earlier. Before enter in the Q-Learning method, should be cited another research based on the adaptive critics that was developed by Werbos' team in Werbos (1992), under the name of Action-Dependent Heuristic Dynamic Programming (ADHDP) and has the same objective of estimate the output based the inputs of states and action, which for Werbos the states is represented by $R(t)$ and the action $u(t)$ in a continuous time domain.

The Q-Learning idea is to act in the Markovian domains where the agents can learn based on their experience when taking actions, so according to equation (19), the actual Q value is determined in two parts, where the first term means how much the value function in position (s, a) will be decreased based on the learning factor α, and after, in the second term, the value function is increased based on the reward and the maximum value for the next better action (a') from the next state (s'), so the result means if this position will be interesting for future states. Notice that the values for Q are based on the load. As it is possible to see, the Q-Learning definition comes from the value iteration concept to solve the problem.

It is important to highlight some variables that were previously treated, such as the learning factor and the discount factor. The former means the amount of information the new value will override the old information, and latter represents how the future values will change the current reward. Both values impact directly in $Q(s, a)$ and consequently for the choice of new actions, since the policy uses this value to determine future action.

The formal definition for the Q-Learning function is:

$$Q(s, a) \coloneqq (1 - \alpha)Q(s, a) + \alpha(R(s, a) + \gamma max_{a'}Q(s', a')) \tag{19}$$

where:

Q – is the matrix which represents the value function.

s – is the current state.

s' – is the future state when taking the action a.

a – is the current action taking.

a' – is the future action from s'.

R – is the reward function.

α – is the learning rate, $0 < \alpha \leq 1$.

γ – discount factor, $0 \leq \gamma < 1$.

The Q-Learning algorithm is simple and consists in the first moment initialize the value function $Q(s, a)$ arbitrary, and for each episode initialize the state that will be used as initial point for the next sweep, which is chosen one action based on the policy selected, after should be determined the new state from the action selected to calculate the reward. Then, this reward determines the new value for $Q(s, a)$, and in the end, the future state s' becomes the current state for the next cycle.

The use of Q-Learning is due to some factors as, simple implementation to generate the first analysis of the problem to be solved in thesis, a technique that can be used on-line and it is a method that can be enhanced to be more robust to avoid some problems as the dimensionality curse.

Pseudocode #4 Q-Learning algorithm

Initialize: $Q(s,a)$ according to the problem
Repeat in each episode:
  Initalize s (state)
  Repeat (for each step of the iteration)
    Choose a (action) from s (state) using the policy derived from Q
      e.g., greed policy
    Take an action a
    Observe r, s'
    $Q(s,a) \coloneqq (1-\alpha)Q(s,a) + \alpha(R(s,a) + \gamma max_{a'} Q(s',a'))$
    $s \leftarrow s'$
  until s is terminal

SOURCE: Sutton; Barto (1998).

2.6 FINAL DISCUSSION

Due to the complexity of the problem related in this thesis, it is hard to find a single strategy that can handle different approaches at the same time, so, the idea is to use the method inside the Machine Learning area. As pointed in Brown (2008) the ADMS should be integrated with a list of functionalities to control and monitor a complete distribution system, and not independent methods to solve the network issues isolated. Thus, to create more this integration, the idea for the present work is to prove, using the usual problems (SH and LS), the development of an unique strategy to take care about the cited issues and select the correct action to act on the distribution system. Despite of all benefits, to develop an ADMS takes a long time, investment and a great hardware due to its complexity of inputs to be analyzed and the process to communicate with all algorithms into this framework. Therefore, if it's possible to minimize the number of algorithms and communication process, it's a gain to construct the ADMS in terms of maintenance and understanding of the workflow.

The selected algorithm to be developed is the Reinforcement Learning, where it is a viable technique to solve the problem of multi-functionalities. The distribution network can be modeled as the environment, and the switches/shedding the actions to be taken by agent to interact with the environment. Furthermore, as discussed in Cheng; Yu (2019) the RL has a strong adaptability to handle with unexpected issues, and already has many applications in the power system area.

The algorithm to be utilized will be the Q-Learning where can be possible to calculate the rewards based on the distribution network load in each system configuration according with the action selected. Moreover, the value function matrix (Q) can store all the environment information in relation to the changes on the network to choose the best option in further iterations.

The way to model the reward function and the Q-matrix will be introduced in chapter 4, as the complete process and adaption to work in line with the distribution constrains (feeder voltage and current limitation), in order to avoid any wrong configuration (switching position) during the procedure to recovery the system.

3 BIBLIOGRAPHIC RESEARCH

This chapter brings the contributions about the mainly areas of this work, the first on is demonstrated works about the Self-Healing, the second references about the Load Shedding applied on Power System and the last is the Reinforcement Learning to solve the Self-Healing problem.

3.1 SELF-HEALING

This chapter presents the most varieties of methodologies to solve the Self-Healing problem since centralized approaches until decentralized, as integer linear programming, meta-heuristics and multi-agent system. Together with the SH some others approach might be considered as the Load Shedding and Microgrids.

In Li et al. (2010) the goal is to execute a self-healing action to avoid the disturbances propagation on the grid and the unbalance between generation and demand that may occur after an action to minimize the active and reactive power. The network topology is represented by a weighted graph to use the partitioning graph method, where the idea is to minimize the active and reactive power. The article concludes that a load shedding should help the self-healing logic to make the method more robust and to minimize the effects of possible failure cascade. Another article that brings the Self-Healing approach together with the Load Shedding is Cavalcante et al. (2016), which call the centralized method as two-stage procedure. The SH is solved using the Mixed Integer Linear Programming (via CPLEX solver), and the LS via a Nonlinear Programming (via KNITRO solver) after the first stage found the switching action to be executed.

Another mixed approach between SH and LS can be seen in Ferreira; Siebert; Aoki; et al. (2014), where the Self-Healing is solved via the Binary Particle Swarm Optimization (BPSO) and, in case there isn't any possible reconfiguration without trigger an overload, the next step is to start the Load Shedding, through the Optimal Power Flow technique, to decrease the load and make possible the reconfiguration.

Using the graph theory, in Kosťálová; Carvalho (2011), the principal objective is to facilitate the distribution network radial model through a bipartite graph, where the focus is to highlight the switches and abstract the other network elements. Making the network simple, the methodology utilized for the reconfiguration is a simple search on the bipartite graph seeking for the better sequence of switching, where the better path is based on the less loaded path. In case the solutions are overloaded, it is decreased the segment to be restored.

The article presented by Botea et al. (2012) talks about a distribution system based on graph theory to use the A*. The heuristic proposed for the method is the load maximization that don't belongs for the isolation region and that were de-energized, and the number of switching action to be executed when reconfiguring the system.

The self-healing problem is solved in Zidan; El-saadany (2012) by the multi-agent method, which are created layers called zone and feeder, to change information between them. The first layer is the zone and has the objective to monitor, to execute simple equations and control actions. In the second layer, the feeder, is the negotiation. The methodology constrains are the voltage and current limits and the radial network configuration. Another multi-agent system is demonstrated in Liu et al. (2012) for self-healing. In this case the model is based on the five operation states (emergency, restorative, alert, insecurity and security) and in the four controls (emergency, restorative, corrective and preventive). The agents created are separated into three layers of action, the first layer is the response layer, the second layer is the coordination layer and the third layer is the organization layer.

In Leite; Mantovani (2017) came up with the Multi-agent system to execute the system restoration, in this case the agents are modeled as the Agent Communication Protocol, Switching Local Agent, Analysis Agent for State Estimation and Self-Healing Coordinator Agent. The Communication Agent is constructed based on the IEC61850 standard protocol to be the link between the Switching and SH Agents. The Switching Agent is responsible to change information with other agents to obtain the network measurement status and for this analysis it's used a fuzzy controller. To obtain a better information from the field the State Estimator Agent has the objective to provide reliable data to load forecast, which it is used in the normal, short-circuit and restorative operation condition. The SH Agent is responsible by the power flow analysis, the shot-circuit simulator and the decision maker for isolation and restoration.

Continuous in the multi-agent topic, the paper Sampaio et al. (2016) presents the agents defined as Substation (SA), Feeder (FA), Branch (BA) and Equipment (EA). The Equipment Agent is responsible to report the information from the field and communicate with the FA and SA. Then the Feeder Agent is responsible to change information with other FA to determine the available power, beside the BA to determine the constraints and receive information from SA related with the restoration stage. The Branch Agent measures what will happen with the action taken and report the FA. At the end, the Substation Agent is owner to process the isolation logic and start the restoration process to FA. In the distribution approach, but outside of the multi-agent area, the author in Torres et al. (2018) developed a new distributed strategy to pass through the location, isolation and service restoration. The algorithm brings a new type of approach, based on groups (set of one or more switches), to understand the field, communicate among the groups and find the best solution according the environment conditions.

The methodology developed in Arefifar et al. (2013) presents two stages, the first stage is the planning, in which is obtained the best network configuration creating microgrids in the distribution system. The second stage is the operation, where after the configuration should be necessary adequate the microgrids created. Moreover, a Tabu Search is included to solve the self-healing problem, some constrains were thought to the search, such as, the auto-adequacy for microgrids, the loss minimization and the load maximization. The algorithm created also realizes the load shedding to support the self-healing action and calculate the better dispatch for the generation spread across the grid. Another article that uses the Tabu Search for self-healing is Mori; Muroi (2011), where the difference the author applied probabilistic samplings into the Tabu Search, where the main idea is to reduce the computational processing to find the switching actions.

In Chen et al. (2015), it is considered for the restoration problem the uncertain of load and Distributed Generation installed on the distribution network. The method utilized is the Information Fap Decision Theory based on the envelope-bound model. The idea behind this methods is to avoid the self-healing solution using the load as a constant value, so the methodology is divided in two steps, the first one is called Determined Restoration Optimization, which carry out the network constrain analysis based on a constant load. After, is executed the Robust Restoration Optimization to maximize the solution from the uncertain of load and the distributed generation.

In Li et al. (2014) is applied the minimum spanning tree method to determine the best configuration after a fault on the distribution system; furthermore, this algorithm also take in account the microgrid influences. The methodology aims to reduce the number of switching actions and restore the load de-energized without any current and voltage violations.

The next papers present the Self-Healing operation considering microgrids and renewable energy. In Wang; Wang (2015) is proposed a self-healing method considering dispatchable (micro turbines) and nondispatchable (wind turbines and photovoltaic) distributed generators to create microgrids self-supplied. The technique comprehends a rolling-horizon optimization to schedule the dispatchable distributed generators and a stochastic rolling-horizon for nondispatchable. When the system is in normal operation, the objective is to minimize the operation cost and when the fault occurs in the system, the objective is to supply the maximum possible of customers using the microgrids.

A distributed Self-Healing and Microgrid logic is modeled in Wang et al. (2016) which in the normal operation follows the objective before, where the algorithm should schedule dispatchable distributed generator, energy storage system and controllable loads to minimize the operational costs and maximize the supply for each microgrid. In a case where a fault occurs, each microgrid can communicate with the neighborhoods to request for more supply if the current microgrid can’t energized all loads.

A mixed approach of Self-Healing, Microgrid and Load Shedding is presented in Wang; Wang (2017). The SH is solved using the Advanced Metering Infrastructure (AMI) to determine the outage location and the mixed-integer quadratic programming for the service restoration. The method takes in account the distributed generators installed on the microgrids and the possibility of load curtailment, where each approach is activate when the traditional reconfiguration cannot be performed, so the mixed-integer quadratic programming calculates the better solution according to the constraints setted.

The strategy applied in Hosseinnezhad et al. (2018) is to predicted one day ahead the system operation considering normal and emergency conditions. In normal condition the idea is to minimize the operational costs and in emergency the planned the cases to island the segments to supply the maximum load possible and adjust the generation to balance the new system. The article also considers the possibility to shed when it is not possible to island the load.

Based on Dynamic Programming (DP) is presented the article Pérez-guerrero et al. (2008) and Riahinia et al. (2018). The first article enhanced the DP to reduce the number of states and the optimization starts to find the better path (necessities of the feeder to be energized) among the stages of the process. Moreover, in each stage it’s optimized the better state to be chose in the end, when the DP scans the entire process to determine the final solution. The second article also applied the reduced state approach and considers the following topics: uncertain from transmission networks, load priority, integration of distributed generation and storage units.

3.2 LOAD SHEDDING APPLIED IN POWER SYSTEM

In this subsection is listed some articles that comprehends some techniques to shed the load in power system to maintain the generation and load balanced. The first article brings a combination of nonlinear mathematical programming and discrete differential equation from the system to estimate the optimal value for the load shedding, this method also supports the operation to know what it is the better time to execute the shedding. The limits used for the methodology are the maximum power flow in the cable, voltage, angle and the maximum load shedding (APONTE; NELSON, 2006).

In Lopes et al. (2006) is presented a load shedding algorithm to stabilize the micro-grid system when an island occurs. The first step, in this method, is to model the load because each load modeled can be verified the frequency range deviation to determine the amount of load to be shed and it is also considered a priority for each load. Another approach is done in Solanki et al. (2007) which is proposed a multi-agent system to solve the self-healing problem and the load shedding is applied inside of the agent to execute the action when necessary.

One different consideration is executed by Faranda et al. (2007) that considers the load shedding inside of the consumer, where the smart appliances can be shed to keep the system in a normal operation. It is helpful when the numbers of participants are bigger because the numbers of appliances are less than the program has few adopters. Other important paper to discuss is Fernandes et al. (2008), the authors utilize the Optimized Power Flow (OPF) with relaxation to determine which loads should be shed. The load prioritization is also done in this work to avoid possible hospitals, for example. In Ghaleh et al. (2011) is combine one analysis of frequency and voltage to apply on the protection curves inside of one relay. The model works in transmission system, which is more often to detect the system instabilities.

One more strategy is demonstrated in Tang et al. (2013), which is developed one centralized algorithm to execute an adaptive load shed, based on the low frequency and voltage. The differential in this case is to use inside the equations the consideration of active and reactive power. Basically, the methodology utilized one system of low order of frequency response, together with load models that has voltage dependence to determine the total difference between active and reactive power to be shed.

3.3 REINFORCEMENT LEARNING ON SELF-HEALING

Based on a mix of techniques, in Ye et al. (2011), Ghorbani et al. (2014) and Ghorbani et al. (2016), execute the self-healing by the Multi-Agent System together with the Reinforcement Learning algorithm for power system. The first article focuses on the transmission system, when occurs loss of generation, so it’s modeled three types of agents (Generator, Switch and Load) to communicate among them to find the better solution. The Q-Learning algorithm is implemented on Switch Agent, which is responsible to receive the information from the Load and Generator agent to determine which switches should be turned on/off.

The second article focuses on the distribution grid and it is composed by Zone Agents, Feeder Agents and Substation Agents. The Q-Learning algorithm is located in each Feeder Agent and learns the option action individually, based on the Zone Agents information and communicate with other Feeders and Substation agents. Another point to highlight is the creation of Q-matrix which is modeled in the number of zones (comprehend the segment between the switches in a feeder) and the adjacent feeders with a tie-switch. The third article is a continuous work of the article Ghorbani et al. (2014), and according to the author the contribution is related with the Fault Location and Isolation approach in the Feeders agent together with the learning methodology. With this approach could reduce the communication messages and computational process.

In Ribeiro et al. (2017) is proposed a tool to simulate the SH problem, where it’s used the standard Q-Learning algorithm with ε-greed policy to solve the system reconfiguration, and the constrains are calculated through the Newton-Raphson method.

In Pal et al. (2010) and Das et al. (2013) is applied the Reinforcement Learning to solve the Self-Healing problem in a Naval shipboard. The former article introduces the problem inside of the shipboard, where it may be related with the distribution power system and its necessities as vital load and constrains, such as, maximum and minimum of power generation, voltage and current. The algorithm used was the Q-Learning, the policy was the ε-greed and the reward function was the variation between the previous topology with the actual after the action. The second article includes more detail about the test and improves the reward function to calculate the time response of the generator when a switch is commuted. In both works the problem was solved fast and find the best result after the fault, in addition, the algorithm also provides the switch sequence to be executed.

3.4 FINAL DISCUSSION

As demonstrated in this chapter the Self-Healing problem can be solved in many ways (multi-agent system, meta-heuristics, integer programming, reinforcement learning, etc.) with different strategies, centralized, decentralized or distributed. To support the solution for SH problem, another technique can be used in sequence to support the final decision selected, as the Load Shedding, where the distribution network does not have enough capacity to transfer load, and a partial load reduction can be executed to restore many load as possible.

In the case of the Load Shedding, also has many methodologies to apply and where should be executed, as a complete segment delimited by switches, or open a transformer with a quantity of consumers, or, when the distribution presents the Advanced Meter Infrastructure, the final consumer might be disconnected. Furthermore, in any approach the priority selection must be considered because of the vital customer that cannot be turned off, for instance hospitals.

The newest Self-Healing algorithms currently should start to think about new technologies and concepts of operation applied in the distribution network, such as, distributed generator, storages and microgrids. The microgrids, can be used in an abnormal network state to produce self-sufficient energized cells to mitigate partial or complete load curtailment.

There isn’t a huge amount of papers considering the reinforcement learning in SH problems into the distribution area. However, as the idea for naval ship and distribution can be similar, some papers in both areas was presented in this chapter, where the article Das et al. (2013) was the trigger to understand the method and how could be adapted for this thesis.

As punctuated in Cavalcante et al. (2016) that the LS is solved after the SH, the idea here is to walk against the common way to prove that is possible to find the switching actions and load curtailment at the same moment, and not firstly define the actions sequence and after the amount of load to reduce. Furthermore, even though some works are aiming microgrids, the fundamental idea here is to look for a methodology where the self-healing and the load shedding can be solved at the same time through the same algorithm, which is complex enough to increase with more items to be considered. Another important point to highlight is instead of having an outage, but an overload the same algorithm can be triggered to determine the best system reconfiguration. Thus, the present thesis starts a new area of researches about the convergence of ADMS functionalities in a common decision maker.

4 MATERIALS AND METHODS

In this chapter, it will be presented the material to validate the algorithm developed to solve the problem of integration of ADMS functionalities. After it’s presented how the methodology was created.

4.1 MATERIALS

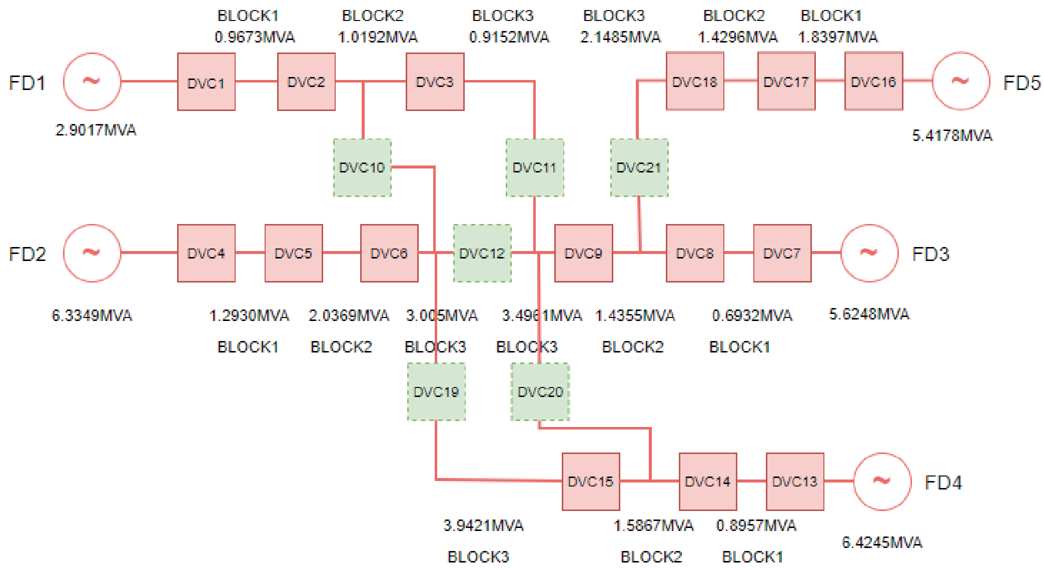
To develop and validate the methodology is presented the materials that are composed by the distribution system data, following by the computational platform and the respective framework to develop the idea from this thesis. For the tests were utilized two difference system, one is the complete system to validate the simulations. The second system, it’s a reduced scheme of the first to test the results into the limited time of three minutes and which represents most of the cases in Brazil.

4.1.1 Distribution system data: five feeders

To perform all tests and to validate the initial methodology proposed, it was used a real distribution system. The called five-feeder system is shown in FIGURE 12, and it composed by five feeders, 15 normally closed switches and six normally opened switches (tie-switches). Each feeder has at the least one or more interconnections, where feeder 3 presents connections among the other four feeders. To highlight the feeder 1 and feeder 4 are interesting because the tie-switch allocated in block2, which provide more resource for the respective feeder and more possibilities for restoration. The switches DVC1, DVC4, DVC7, DVC13 and DVC16 are automatic reclosers that are usually inside the power substation and called as Feeder Head.

Each combination of two switches creates the load concentration which is called as block. The first block starts near of the source and the last block is delimited by a closed switch and the tie-switches. The power and current for each feeder is presented in Table 2, the feeder 2 and feeder 4 are the most loaded and feeder 1 is the most available feeder to be used in restorations approaches depending of the constrains configured.

FIGURE 12 – TEST SYSTEM WITH FIVE FEEDERS



SOURCE: The author (2020).

Table 2 – POWER AND CURRENT SYSTEM CONFIGURATION FOR FIVE-FEEDER

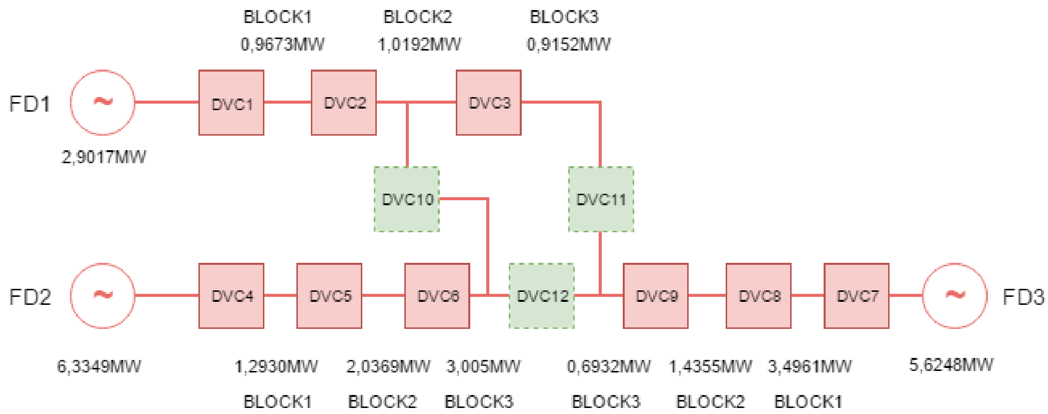
Feeder 1		Feeder 2		Feeder 3		Feeder 4		Feeder 5		
Power [MVA]	Current [A]	Power [MVA]	Current [A]	Power [MVA]	Current [A]	Power [MVA]	Current [A]	Power [MVA]	Current [A]	
0.9673	74.4077	1.2930	99.4615	0.6932	53.3231	0.8957	63.9	1.8397	141.5153	Block 1
1.0192	78.4000	2.0369	156.6846	1.4355	110.4231	1.5867	122.0538	1.4296	109.9692	Block 2
0.9152	70.4000	3.0050	231.1538	3.4961	268.9308	3.9421	303,2384	2.1485	165.2692	Block 3
2.9017	**223.2077**	**6.3349**	**487.3000**	**5.6248**	**432.6769**	**6.4245**	**489.1922**	**5.4178**	**416.7537**	**Total**

SOURCE: The author (2020).

4.1.2 Distribution system data: three feeders

A subset of the complete topology is created to be used in some special cases, besides the normal tests. This topology is comprehended by the feeders 1, 2 and 3. The one-line diagram is shown in FIGURE 13 and contemplates three distinct sources, with three normally closed switches in each feeder and three normally open switches that connect one feeder with the others. The feeder head are DVC1, DVC4 and DVC7. The power and current for each feeder is presented in Table 3, the feeder 2 and feeder 3 are the most loaded and feeder 1 is the most available feeder to be used in restorations approaches depending of the constrains configured.

FIGURE 13 – TEST SYSTEM WITH THREE FEEDERS



SOURCE: The author (2020).

Table 3 – POWER AND CURRENT SYSTEM CONFIGURATION FOR THREE-FEEDER

Feeder 1		Feeder 2		Feeder 3		
Power [MVA]	Current [A]	Power [MVA]	Current [A]	Power [MVA]	Current [A]	
0,9673	74,4077	1,2930	99,4615	0,6932	53,3231	Block 1
1,0192	78,4000	2,0369	156,6846	1,4355	110,4231	Block 2
0,9152	70,4000	3,0050	231,1538	3,4961	268,9308	Block 3
2,9017	**223,2077**	**6,3349**	**487,3000**	**5,6248**	**432,6769**	**Total**

SOURCE: The author (2020).

4.1.3 Computational Platform

In this part is presented the computational platform and software used to the development of algorithms and to realize the tests. To simulate the tests, it was used a desktop with operational system Windows 10 Professional 64 bits, with a hardware configuration AMD phenom II of 2.8GHz, 8GB of memory and hard disk of 1TB.

All methodology was developed in MATLAB (MATrix LABoratory) of Mathworks. Furthermore, to calculate the power flow was utilized the OPENDSS of EPRI, which is a program of power flow simulation focused on distributed system, where there are two methods to calculate: current injection and Newton for iterative methods (DUGAN, 2013). For raw data was used the Microsoft Office Excel (Office365).

4.2 METHODS

The methodology developed in this thesis was motivated from Pal et al. (2010) and Das et al. (2013), both article present the same problem, but each one complements the method developed to solve the reconfiguration on shipboard after an outage occurs. Bringing the main concept that is the application of reinforcement learning to solve the SH problem in a distribution network.

The algorithm proposed, in the FIGURE 14, is divided in two parts, the first is called **initialization phase** and the second is the **learning phase**. The initialization phase is comprehended by the first part of the Self-Healing, identifies and isolate the fault, where the logic used is in according to Ferreira (2015) methodology. After the isolation, it's created the list of actions based on the switches that was not isolated are selected to be able to interact with the system, together with the preselected percentual of load shedding. In a case of an overload, it's not necessary to identify and isolate a fault, so the algorithm can jump to create the list of actions with all switches.

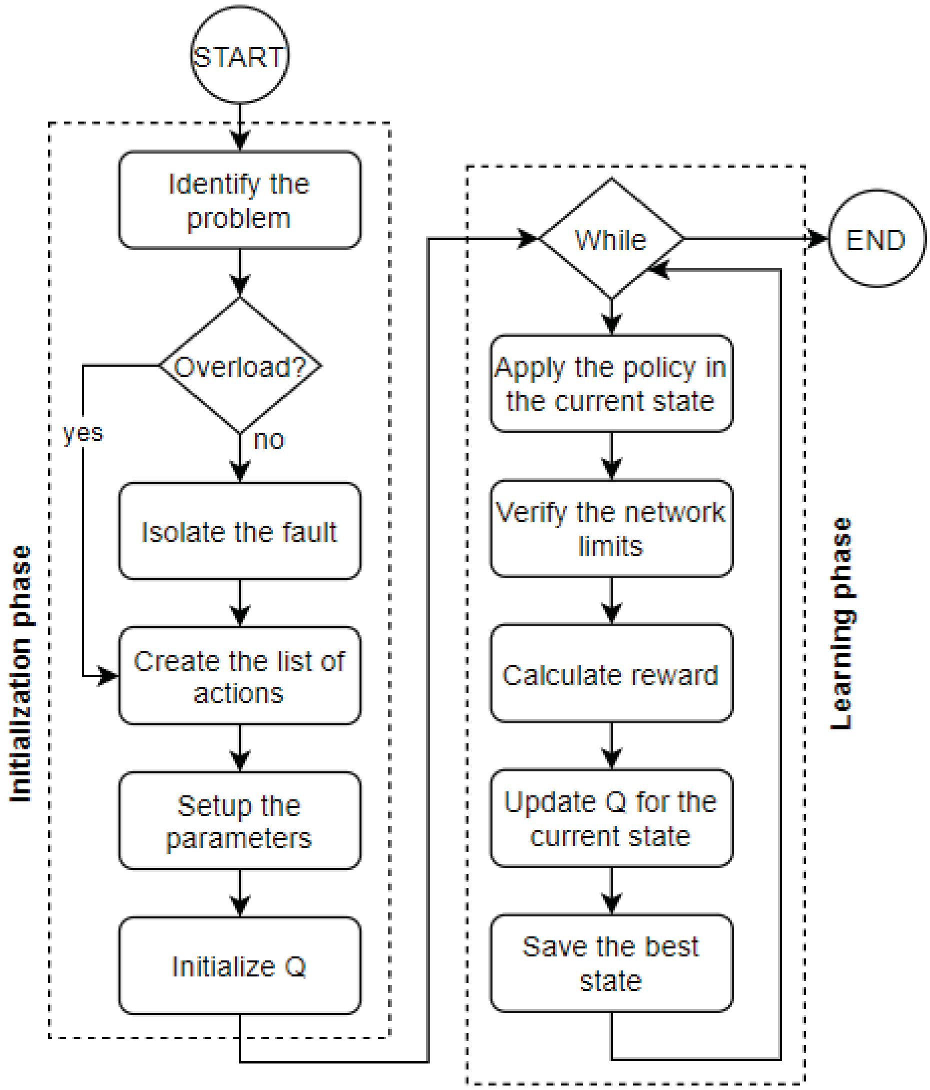
The Q-Learning and policy parameters, as well the number of iterations should be setup before the loop start. The QMatrix initialization might be done in two ways, starts with zeros and according the iterations are processed the QMatrix is filled, or initialize with the total load according the topology (lines) and action (switches) to be taken. After this initialization, the algorithm can run and find the best actions to be executed and solve the problem. Note that for each new state selected, the limits to avoid any new failure in the system is verified.

Before to introduce the details of the algorithm, it is necessary to understand what QMatrix represents and associates some terms about reinforcement learning with the system built for this program.

a) **agent**: switch or load;
b) **environment**: Distribution System and its constraints;
c) **action**: Open/Close switch, or Load Shedding [%];
d) **state**: System topology;
e) **reward**: delta load;
f) **QMatrix**: matrix to save the accumulate knowledge in each iteration.

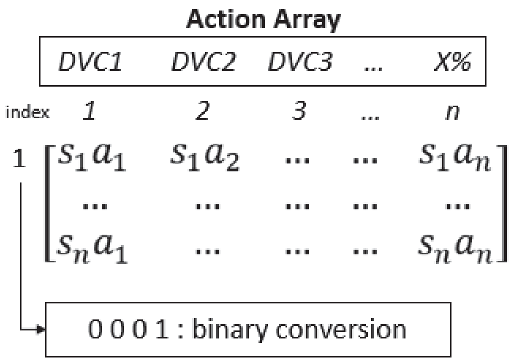
The QMatrix is one of the important points developed in this thesis, where the knowledge stored is related with the rows – the index represents the distribution topology in decimal that should be converted in binary, and the columns – the actions to be performed, each index represents a possible device to commute or a perceptual of shed. For a visual detail look in FIGURE 15, that represents a topology with four devices and one level of shed.

FIGURE 14 – LOGIC FLOWCHART



SOURCE: The author (2020).

FIGURE 15 – UNDERSTANDING THE QMATRIX



SOURCE: The author (2020).

4.2.1 Identify the problem

The algorithm can be triggered by two different problems, the first is an overload, where some equipment has a current limit and when the system has an increase of load that exceed this limit, the control system should take an action to avoid the overload.

The second trigger is related with the unexpected outages, for example a car crashes on a pole, or some storm that de-energized parts of the system, etc. Then, the IEDs can process the over current and send to the control center to about the problem and the intelligent algorithms (or the operators) must take any action to correct the issue on the field.

4.2.2 Identification and Isolation

The fault identification and isolation are based in the same principle as done in Ferreira (2015) using the graph theory. The fault is associated with a load block (a node in a graph), so all edges (switches) connected with this node should be isolated. Notice that the distribution system topology approach used in this thesis is different, where the normal representation is the switches as nodes and the loads as edges.

4.2.3 List of Actions

Once the system found the problem and reduced the search space in the cases of unexpected outages, the list of action can be created. The list will impact directly in the number of states, i.e., in the number of lines in Q-matrix, once the Q-matrix size is exponentially based on all possible topological configurations (2^<number of switches>). The columns are also affected because the it's composed by the total number of switches to open as an action, plus, the number of switches to close as an action, plus, the amount of selected percentual to shed. Moreover, the feeder head is removed from the actions, once there isn't any sense to turn off a complete feeder from the beginning.

To exemplify, a system with 12 switches, three feeders, four different level of load shedding and two switches were used to isolate the fault, the QMatrix sizes is 2^7 lines x 7+7+4 columns, where the number seven represents the amount of switches to be commuted.

4.2.4 Parameterization and Initialization

There are some parameters to be configured before the Q-algorithm starts, the number of maximum iterations, this avoid the case when the logic can't find a final solution for the problem presented for itself. The learning rate and discount factor, those are used to calculate the value function in each iteration. The last parameter related with the learning process is the policy balance between explore and exploit from the ε-greed approach. For more details go to chapter two. In the power system side, there are the system maximum of capacity and the maximum and minimum of voltage in each feeder segment.

The next step is to initialize the values of Q, as commented before the QMatrix can be completed with zeros or using the load equation. The second approach is related with the idea of the environment (distribution system), the function proposed was based on the load in each state. It is executed the reward idea from Reinforcement Learning, where the algorithm will run for each column and row and calculate the reward based on the delta variation between the previous state and the current state, according to equation (20).

4.2.5 Learning Process

The beginning of the learning process considers as input the state of the distribution system after the isolation, so the algorithm will find the best action to be selected by the ε-greed policy, as explained on the chapter about reinforcement learning and demonstrated in equation (7). Then, the topology will change if the action is related with a device or will change the feeder load when a shedding action.

As the algorithm was design to consider the ε-greed policy, so it's generate a random value between [0-1], if less than ε value, the actions to be selected from the QMatrix will be random (explore concept); otherwise, it returns the QMatrix column index which represents the action array to be taken (exploit concept). In a case where the action is a device position, the new state is obtained converting the row index minus one (because Matlab doesn't have index zero) in binary, after updating the binary vector with the correspondent action and reconverting in decimal plus one to be now the current state (topology) and the row index. However, if the action chosen is the load shedding, it is just executed a shed on loads that belongs to the transferred feeder.

Determined the new state (topology), the next step is to verify three constraints to avoid any new failure in the system. The first constraint is the permanent parallelism, where it is executed a graph search for each feeder and the same point (device) cannot appear twice. The second verification is the voltage limits, which is used the OPENDSS software to calculate the power flow and obtain the voltage profile. The last limit is the maximum capacity of current on the equipment, in this case is used a general capacity for the whole distribution system. If one of these limits is exceeded the reward for the new state is minus one. Notice, all values are normalized considering the total power calculated when the system is in normal operation mode.

$$R(s,a) = \Delta L = L(s') - L(s) \tag{20}$$

If no limits were exceeded, the reward can be calculated according equation (20), which represents the variation between the load from previous state and the load for the current state. The values can be positive, that represents an improvement or negative, where the new state has less load than the previous and indicating a bad action choice. Therefore, with a reward value for the current state, the Q value can be obtained based on the equation (19).

To determine the best solution the algorithm saves in the end the best state when there isn't limit violation and has a load improvement. The algorithm just stops when the number of iterations reach the end or there isn't more variation for the policy choice in 10 consecutive iterations.

4.2.6 A brief example of the application of the RL in the Self-Healing context

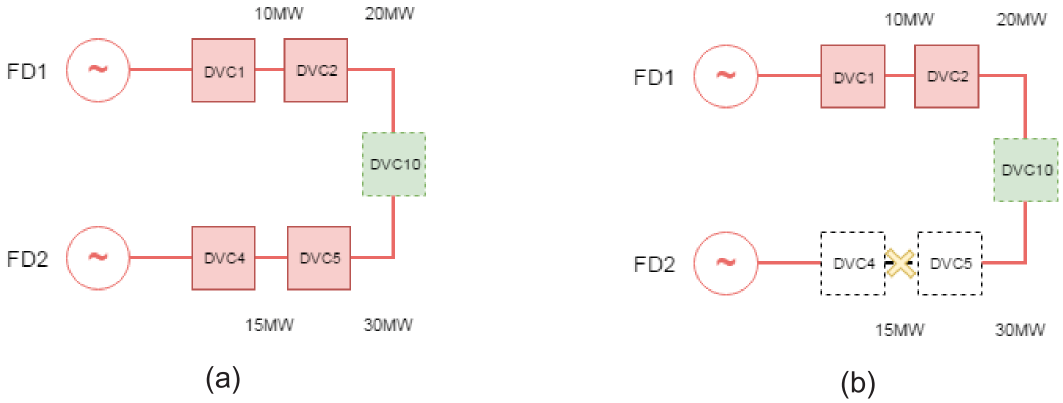
To have a better comprehension about the method developed, it is created a small system to apply in a few steps the complete idea of the algorithm. FIGURE 16 shows the small system in a normal (a) and isolated state (b) and the equation (21) is the initialization for Q matrix after the isolation, notice that there are three possible simple actions, which are Close or Open for DVC1, DVC2 and DVC10. The initial state, that is represented in FIGURE 16 (b) comprehends the line four according to the Table 4.

The first step of the algorithm is to determine the action using the greedy technique, so the maximum value in line four is 1 (60/60). This action is to close DVC10, and this carries for a new state where all devices is closed and represent the line eight.

This new state should be analyzed to verify all limits described above, if there aren't any limit exceed the new value for Q should be calculate based on the reward equation (20). The Q-matrix is update, if this state comprehends a better result from the previously iteration, the state is saved to be used as a solution at the end of process.

A new iteration starts, now with the new state (all devices closed) and the action should be chosen again, in this case the best solution is to open DVC10, but the result return in the initial state, so there isn't better solution than all devices closed, in this case the algorithm can stop and show the best solution stored in the process.

FIGURE 16 – EXAMPLE SYSTEM WITH 5 RECLOSERS. (A) NORMAL SYSTEM AND (B) SYSTEM AFTER ISOLATION



SOURCE: The author (2020).

$$Q_{initial} = \begin{bmatrix} 10/60 & 0 & 0 \\ -10/60 & 30/60 & 0 \\ 30/60 & 0 & 0 \\ -30/60 & -20/60 & 60/60 \\ 10/60 & 0 & 0 \\ -10/60 & 60/60 & 0 \\ 60/60 & 0 & 0 \\ -60/60 & -50/60 & -30/60 \end{bmatrix} \tag{21}$$

Table 4 – RELATION BETWEEN Q-MATRIX AND SYSTEM TOPOLOGY

Q-matrix index	DVC1 position	DVC2 position	DVC10 position
1	0	0	0
2	1	0	0
3	0	1	0
4	1	1	0
5	0	0	1
6	1	0	1
7	0	1	1
8	1	1	1

SOURCE: The author (2020).

4.3 FINAL DISCUSSION

The proposed method was based on the same approach for a Self-Healing in a naval ship operation. As the environment are different the process was changed and included new steps and analysis to be in accord with the distribution system standards, together with the load shedding analysis. Furthermore, as it's not simple to understand the process, it was included a specific section to talk about how the RL method works and applied in a small network, the idea is to facilitate the understanding in the results and conclusion chapter.

To prove the RL method it was selected a real distribution network system with five feeders. The load used to simulate the faults is a snapshot of a determined moment of the day, and to simulate the overload in the system, it was proposed to decrease the capacity in the switches. The five-feeder system was decreased for a three-feeder to analyze some time process and to adjust RL parameter. One of the reasons to consider the three-feeder case is that in Brazil, the distribution network has not yet a high level of automatism compared with North America and Europe. However, there is a specificity that differs Brazil with other countries, as the bigger amount of load concentrated and the distance between the remote commanded devices. Furthermore, as the idea of this thesis is to begin the discussion between the traditional way, where each problem is addressed to one technique, and the new approach, the size of the grid does not interfere with demonstrating the real purpose.

The computer used to simulate the case is old and cannot perform according the currently computer, also the MATLAB is not the better framework to have good performance results. However, as the objective of this thesis is to prove the concept to have two different techniques being solved at the same moment in a single algorithm, the computer and the framework demonstrated enough performance.

5 RESULTS AND DISCUSSION

In this chapter is presented the results to test the developed algorithm. The chapter was divided in five main sections, first the discussion how the policy was selected to be used in the simulations for the next sections, where it was divided in test with three-feeder topology and the other with the five-feeder topology. Furthermore, the fourth section is a comparison with other two different techniques and at the end the final discussion of this chapter.

The parameters for the testes using the three-feeder case were, maximum number of iteration equal 1000, and for the five-feeder scenarios depending according the complexity of the system, the number variates between 20000~100000. In both cases the learning rate and discount factor respectively 0.9 and 0.6. The percentual for load shedding were 5%, 10%, 15% and 20%.

5.1 DEFINING THE POLICY

To define the policy to simulate the scenarios, first it was simulated two different cases to select the better policy to take action during the learning process. Scenario one is a single fault in DVC1 and the other is a double fault in DVC4 and DVC8, both cases using the three-feeder topology. Each case was run 100 times for three different policies: Greed, ε-Greed and Randomize. For the first scenario the objective is to keep at the least 11.96 MVA after the reconfiguration, and for the second scenario a better solution should be more than 3.59 MVA, both cases represent the total load after the isolation step.

The TABLE 5 brings the results for the single fault in a Greed policy, where 14% reached the global solution with 12.76 MVA. In an overview, 63% keeps the topology after the isolation and 35% found a good solution from a mixed reconfiguration and load shedding. Moreover, 2% the algorithm found a bad system reconfiguration because the final load is less than keep with the final distribution topology after the isolation switching. When using the ε -Greed approach, TABLE 6, the solution demonstrated 4% of bad solution, where the load is less than the isolation topology. For a good service restoration, 48% of the cases could reconfigure the topology and apply a load reduction to increase the load. Moreover, 48% keep the load system as the same after the isolation, where 22% open a non-necessary switch (DVC3) in the final reconfiguration. Compared with the Greed policy, the ε -Greed approach has reached in 18% against 14%.

TABLE 5 – GREED POLICY FOR A SIMPLE FAULT

FAULT DVC1	Greed					
DVC3	1	0	0	1	0	0
DVC5	1	1	1	1	1	1
DVC6	1	1	1	1	1	0
DVC8	1	1	1	1	1	1
DVC9	1	1	1	1	1	1
DVC10	0	0	0	0	1	1
DVC12	0	0	0	0	0	0
DVC11	0	1	0	1	0	0
Amount of solution times	34	21	29	14	1	1
Total Power [MVA]	11.96	12.548	11.96	12.76	11.508	8.955
Shed [%]	0	5	0	15	20	0

SOURCE: The author (2020).

TABLE 6 – E-GREED POLICY FOR A SIMPLE FAULT

FAULT DVC1	ε -Greed							
DVC3	0	0	1	1	0	0	0	1
DVC5	1	1	1	1	1	1	1	1
DVC6	1	1	1	1	0	0	1	0
DVC8	1	1	1	1	1	1	1	1
DVC9	1	1	1	1	1	1	1	1
DVC10	0	0	0	0	1	0	1	0
DVC12	0	0	0	0	0	0	0	0
DVC11	1	0	0	1	0	1	1	0
Amount of solution times	30	22	26	18	1	1	1	1
Total Power [MVA]	12.221	11.96	11.96	12.76	8.955	9.54	11.81	8.955
Shed [%]	10	0	0	15	0	5	15	0

SOURCE: The author (2020).

The randomize policy presented 2% of inadequate solution, where 57% increase the load in a final solution and the other 41% just keep the load after the isolation. Different of the previous cases, the randomize solution cannot find in any moment the better configuration of 12.76 MVA, the complete analysis is shown in TABLE 7.

TABLE 7 – RANDOMIZE POLICY FOR A SIMPLE FAULT

FAULT DVC1	**Randomize**					
DVC3	1	1	0	0	0	0
DVC5	1	1	1	1	1	1
DVC6	1	1	1	1	0	1
DVC8	1	1	1	1	1	1
DVC9	1	1	1	1	1	1
DVC10	0	0	0	0	1	1
DVC12	0	0	0	0	0	0
DVC11	1	0	1	0	1	0
Amount of solution times	22	24	35	17	1	1
Total Power [MVA]	12.382	11.96	12.221	11.96	9.543	11.508
Shed [%]	20	0	10	0	5	20

SOURCE: The author (2020).

For the double fault scenario, the Greed policy, TABLE 8, found 10 different solution, where 1% a bad solution, 1% keeps the isolation state, 23% with reasonable result and 75% a good solution when this type of fault occurs. The ε-Greed approach, TABLE 9, found 1% of inadequate solution, 3% in isolation state, 8% a reasonable solution and 88% a good solution. The randomize policy, TABLE 10, with the numbers 0% for a bad solution, 3% in the isolation state, 16% a reasonable solution and 81% a good solution.

TABLE 8 – GREED POLICY FOR A DOUBLE FAULT

FAULT DVC4 and DVC8	**Greed**									
DVC2	1	1	1	1	1	1	1	0	1	1
DVC3	1	1	0	0	1	1	0	1	1	0
DVC5	0	1	0	1	1	0	1	1	1	0
DVC6	0	1	1	1	0	1	1	1	0	1
DVC12	0	0	0	0	0	0	0	0	0	0
DVC11	1	0	1	1	1	0	0	0	0	0
Amount of solution times	33	2	4	2	42	10	4	1	1	1
Total Power [MVA]	7.09	7.04	5.68	6.66	7.09	6.59	7.01	1.66	3.50	5.68
Shed [%]	0	20	0	15	0	0	10	0	5	0

SOURCE: The author (2020).

TABLE 9 – E-GREED POLICY FOR A DOUBLE FAULT

FAULT DVC4 and DVC8	ε-Greed							
DVC2	1	1	1	1	1	0	1	1
DVC3	1	1	1	1	1	1	0	0
DVC5	1	0	1	0	1	1	0	1
DVC6	0	0	0	1	1	0	1	1
DVC12	0	0	0	0	0	0	0	0
DVC11	1	1	0	0	0	0	1	0
Amount of solution times	50	38	3	4	2	1	1	1
Total Power [MVA]	7.091	7.091	3.5949	6.599	7.048	1.6605	5.6846	7.0187
Shed [%]	0	0	15	0	20	0	0	10

SOURCE: The author (2020).

TABLE 10 – RANDOMIZE POLICY FOR A DOUBLE FAULT

FAULT DVC4 and DVC8	Randomize								
DVC2	1	1	1	1	1	1	1	1	1
DVC3	1	1	1	0	1	0	0	0	1
DVC5	1	0	0	1	0	1	0	0	1
DVC6	0	0	0	1	1	1	1	1	0
DVC12	0	0	1	0	0	0	0	0	0
DVC11	1	1	0	1	0	0	1	0	0
Amount of solution times	41	40	1	4	3	5	3	1	2
Total Power [MVA]	7.0911	7.0911	3.595	7.0187	6.5999	6.6673	5.6847	5.6847	3.595
Shed [%]	0	0	0	10	0	15	0	0	10

SOURCE: The author (2020).

According the comparison between the simulations, the ε -Greed shown a better policy to solve the self-healing and load shedding because in the first case this policy could reach more results than the other and the main reason is related with the second case, where the number of inadequate solutions was less and the algorithm with this policy could reach more times the better system reconfiguration.

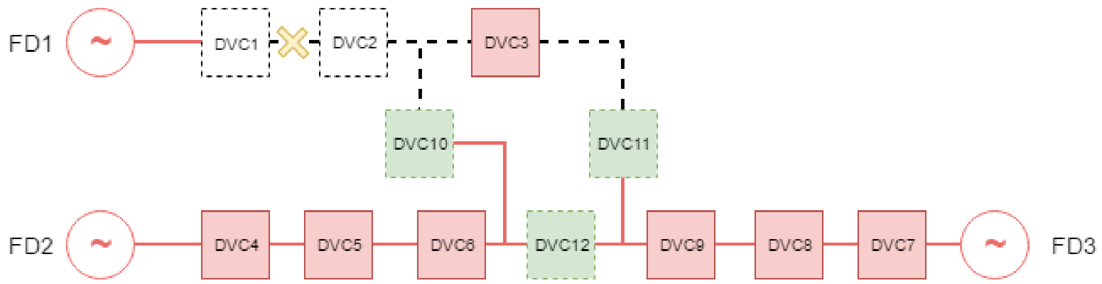
5.2 SIMULATION WITH THREE-FEEDER TOPOLOGY

The first step to prove the efficiency of the algorithm developed it to test in a topology most common in Brazil with three feeders interconnected, where in this case the process time is relevant as the reconfiguration result. To analyze the behavior, it will be shown four different fault cases, in three/four distinct capacity configurations according to each scenario, 400 A, 500 A, 560 A, 600 A or 700 A.

5.2.1 CASE1.1: Fault DVC1 – DVC2

In the first test case, it is simulated a fault between the switches DVC1 and DVC2, where the loads in front of DVC2, with 1.9344 MVA, is de-energized. It was simulated the problem with three levels of maximum capacity on the feeder (500 A / 560 A / 600 A). The isolation state is demonstrated in FIGURE 17.

FIGURE 17 – CASE1.1: TOPOLOGY AFTER ISOLATION

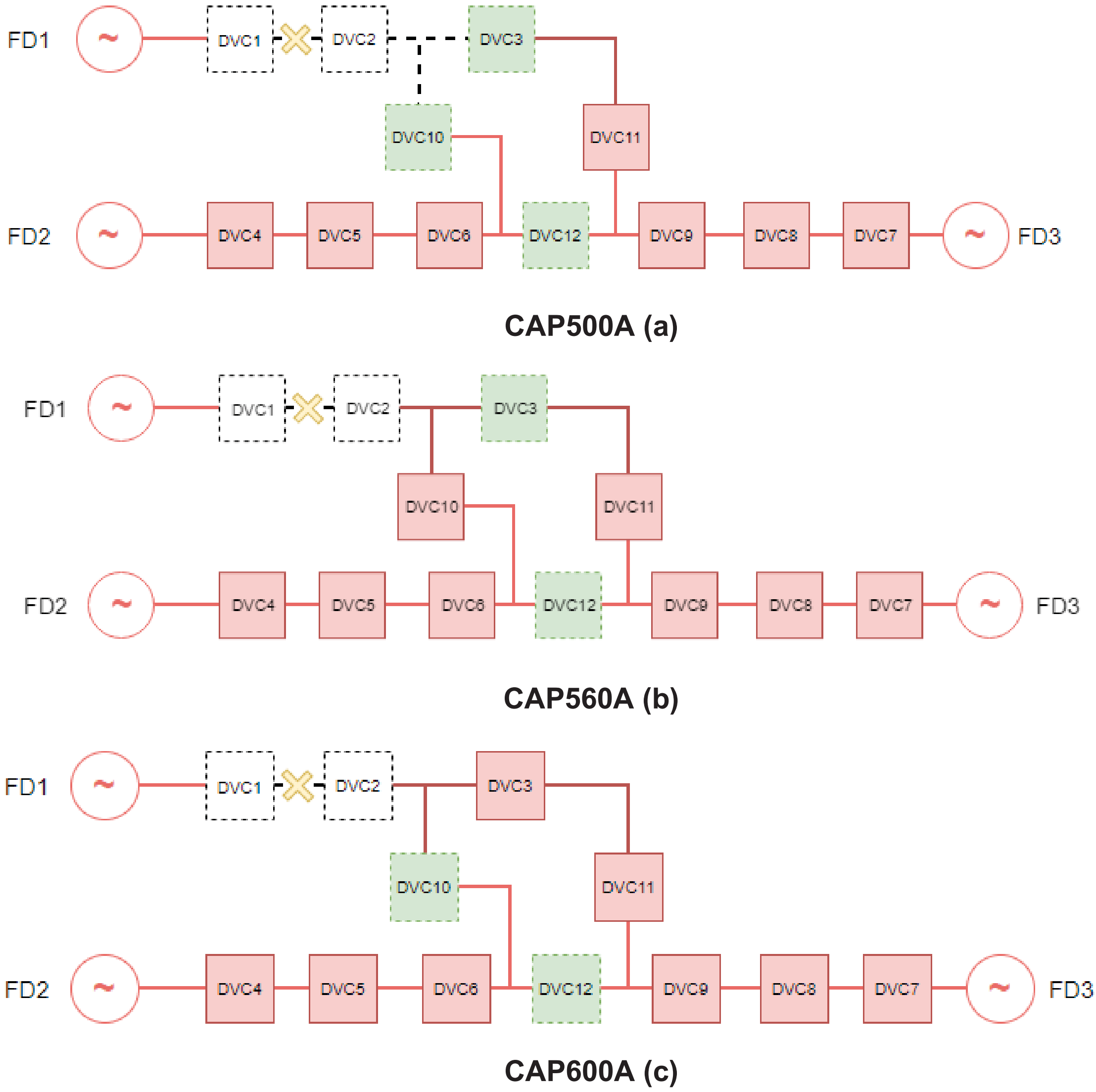


SOURCE: The author (2020).

When the capacity is 500 A, the algorithm selected the DVC3 to open (load shedding via block) and DVC11 to close (service restoration) because the feeder 3 is more available (67.32 A) than the feeder 2 (12.7 A). Then, just the block3 (70.4 A) from feeder 1 can be restored and a small shed (5%) should be executed to keep the current into the limits. For the scenario with 560 A the both healthy feeders are available to restore the load blocks from the feeder 1, so to avoid any overload for one feeder, the algorithm selects a mode to distribute the load for both feeders, so at the end the just a 5% of load reduction in feeder 2 was necessary as Table 12.

The last case with the system configured for 600 A of capacity, it was just necessary to close DVC11 because the feeder 3 had enough capacity to support the transfer. As no load shedding was required a single switch action could be performed, in other words, the algorithm could consider that in this case increase the number of switch action will not change the final load situation. The final topology for each capacity scenario can be seen in FIGURE 18.

FIGURE 18 – CASE1.1: TOPOLOGY AFTER RECONFIGURATION. (A) 500A, (B) 560A AND (C) 600A



SOURCE: The author (2020).

The Table 11 details the process time, as the problem to be solved is the same, so the only time difference is related with system capacity, where more restrictive is the system more complicate is to find a solution. Based on the capacity levels, the Table 12 shows the total load of the system after the final reconfiguration, which it was possible to keep 84% of the total load for 500 A, 91% for 560 A and 93% for 600 A, where the pos-isolation represents 80% of the total. The final solution didn't extrapolate any current limit that was imposed as a parameter, according to the Table 13, and the final solution could avoid also the permanent parallelism and the voltage limits.

Table 11 – CASE1.1: TIME RESULTS AND NUMBER OF ITERATIONS

Scenario	Learning Time [s]	Number of iteration	Position found the best result
CAP500	62.0739	1000	1000
CAP560	47.4579	1000	668
CAP600	37.0105	1000	511

SOURCE: The author (2020).

Table 12 – CASE1.1: SHED AND LOAD RESULTS

Scenario	Best Shed [%]	Shed Feeder	Total power restored [MVA]	Total Power w/ SH [MVA]
CAP500	5	3	12.548	11.96
CAP560	5	2	13.526	11.96
CAP600	0	0	13.894	11.96

SOURCE: The author (2020).

Table 13 – CASE1.1: FINAL CURRENT RESULTS

Scenario	Current Feeder 1	Current Feeder 2	Current Feeder 3
CAP500	0	487.2990	477.9315
CAP560	0	537.4126	503.0858
CAP600	0	487.2990	581.4842

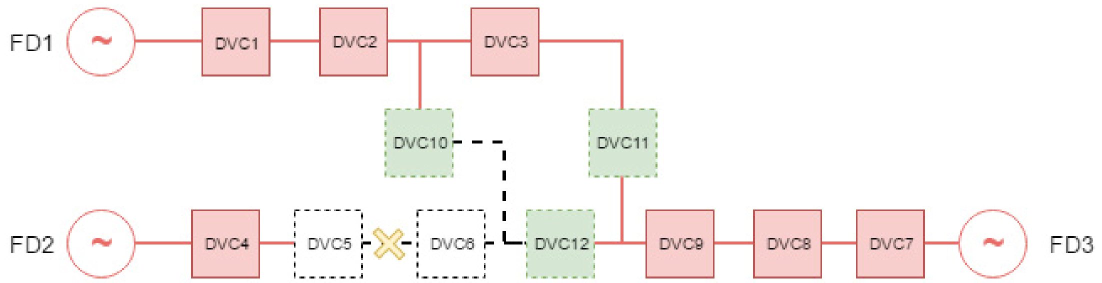
SOURCE: The author (2020).

5.2.2 CASE1.2: Fault DVC5 – DVC6

The second test case simulates a fault between the switches DVC5 and DVC6, where the segment in front of DVC6, with 3.0050 MVA, is de-energized, where in this case the most loaded feeder presents a fault. The capacity analysis follows the previous scenario (500 A / 560 A / 600 A). The isolation topology is shown in FIGURE 19.

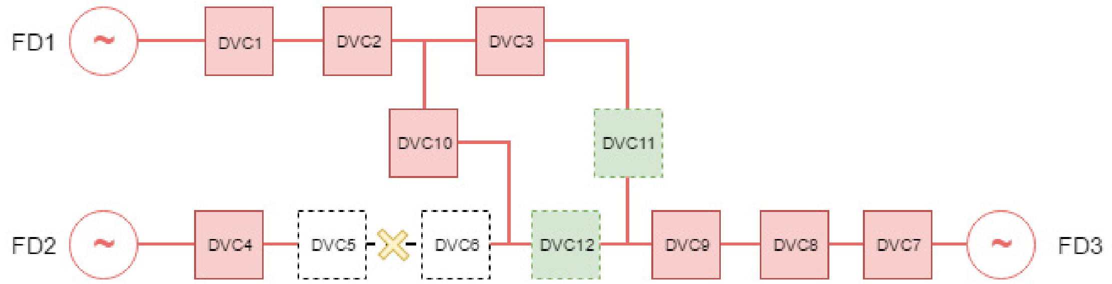
For all capacity limit, the reconfiguration, as demonstrated in FIGURE 20, was done closing DVC10 to energize the last block in feeder 2 because the feeder 1 is more available (276.79 A) than feeder 3 (67.32 A) for the worst capacity case (500 A). In this situation, it wasn’t necessary to shed loads to execute a better reconfiguration, for all capacity levels feeder 1 could receive the last block of feeder 2.

FIGURE 19 – CASE1.2: TOPOLOGY AFTER ISOLATION



SOURCE: The author (2020).

FIGURE 20 – CASE1.2: TOPOLOGY AFTER RECONFIGURATION



SOURCE: The author (2020).

The Table 14 details the process time, as the feeder 1 has enough capacity for all levels, the process time is similar and there isn't any consideration based on the complexity caused by the system restriction. Based on the capacity levels, the Table 15 shows the total load of the system after the final reconfiguration, which for all cases the final load represents 86% of the total load, for comparison the pos-isolation represents 66% of the total. The final solution didn't extrapolate any current limit that was imposed as a parameter, according to the Table 16, and the final solution could avoid also the permanent parallelism and the voltage limits.

Table 14 – CASE1.2: TIME RESULTS AND NUMBER OF ITERATIONS

Scenario	Learning Time [s]	Number of iteration	Position found the best result
CAP500	26.2447	1000	123
CAP560	30.7247	1000	147
CAP600	23.1874	1000	98

SOURCE: The author (2020).

Table 15 – CASE1.2: SHED AND LOAD RESULTS

Scenario	Best Shed [%]	Shed Feeder	Total power restored [MVA]	Total Power w/ SH [MVA]
CAP500	0	0	12.825	9.819
CAP560	0	0	12.825	9.819
CAP600	0	0	12.825	9.819

SOURCE: The author (2020).

Table 16 – CASE1.2: FINAL CURRENT RESULTS

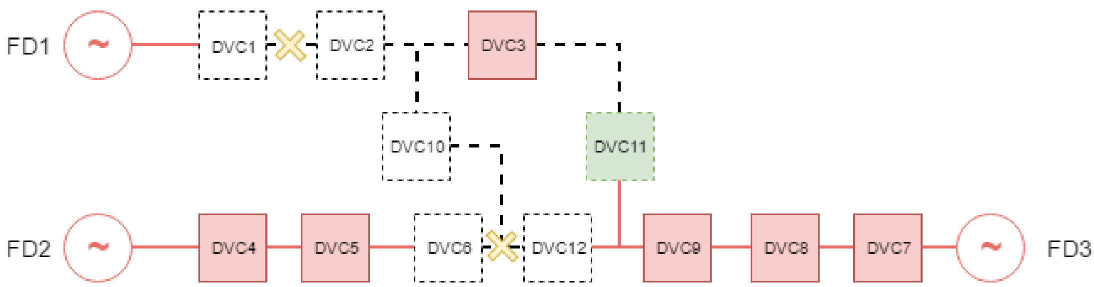
Scenario	Current Feeder 1	Current Feeder 2	Current Feeder 3
CAP500	454.36	99.4644	432.6819
CAP560	454.36	99.4644	432.6819
CAP600	454.36	99.4644	432.6819

SOURCE: The author (2020).

5.2.3 CASE1.3: Fault DVC1 – DVC2 and DVC10 – DVC6 – DVC12

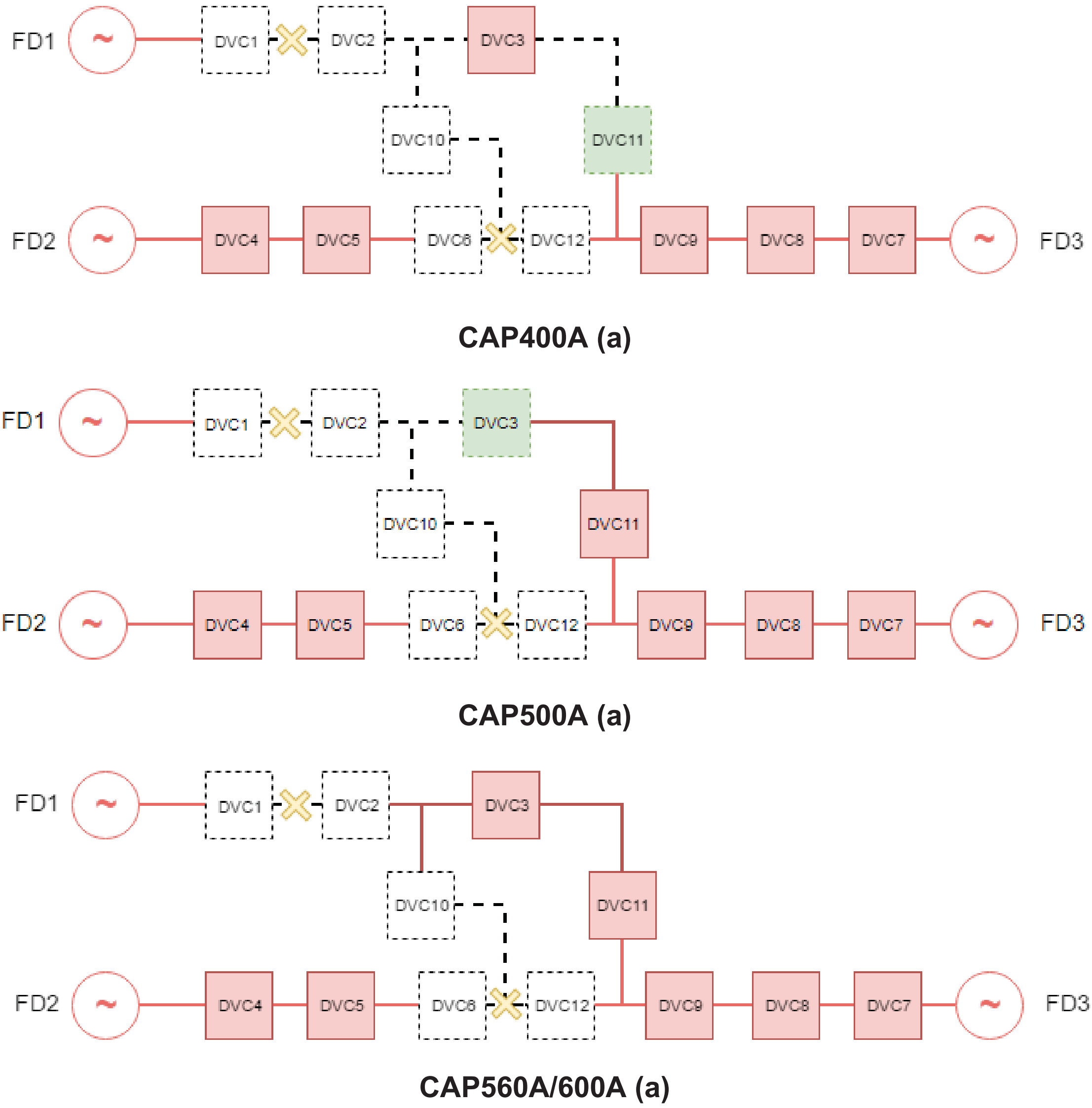
The third test case simulates two simultaneous faults between the switches DVC1 and DVC2, and the other fault is among DVC6, DVC12 and DVC10. For this scenario the capacity constraint was diversified in four different levels, 400 A, 500 A, 560 A and 600 A. When the capacity is 400 A an overload happens at the same time of the fault, so the algorithm should respond from two different problems, unexpected outage and overload. The isolation state is shown in FIGURE 21.

FIGURE 21 – CASE1.3: TOPOLOGY AFTER ISOLATION



SOURCE: The author (2020).

FIGURE 22 – CASE1.3: TOPOLOGY AFTER RECONFIGURATION. (A) 400A, (B) 500A AND (C) 560A/600A



SOURCE: The author (2020).

As expected, for the 400 A case the algorithm kept the topology after the isolation and only execute the load shedding of 10% on feeder 3, without consider the system losses a shedding of 5% should be enough; however, when analyzed the losses together the shed should be increase for 10%. When the capacity is 500 A the result closes DVC11 (service restoration) and provides a complete shed in feeder 1 block 3 (open DVC3), besides a shed of 5% on feeder 3. For 560 A and 600 A the final topology was the same, just closing DVC11, the difference is related with the load reduction (5% - feeder 3) in 560 A case, and not necessary for the 600 A. All final topologies are presented in FIGURE 22

The Table 17 details the process time, for this case the number of switches was dramatically reduced because the fault position, so all times was small. With a difference explanation than the case 1.1, where the worst case was because the system becomes more restrictive with less capacity, in this case because the short search space in QMatrix the algorithm kept in the same location, repeating the same result, the quantity of times to stop the algorithm earlier than the 1000 iterations.

Table 17 – CASE1.3: TIME RESULTS AND NUMBER OF ITERATIONS

Scenario	Learning Time [s]	Number of iteration	Position found the best result
CAP400	4.4682	1000	18
CAP500	8.7008	1000	76
CAP560	10.2041	1000	13
CAP600	9.3744	1000	51

SOURCE: The author (2020).

Based on the capacity levels, the Table 18 shows the total load of the system after the final reconfiguration, which it was possible to keep 56% of the total load for 400 A, 64% for 500 A, 71% for 560 A and 73% for 600 A, where the pos-isolation represents 47% of the total. The final solution didn't extrapolate any current limit that was imposed as a parameter, according to the Table 19, and the final solution could avoid also the permanent parallelism and the voltage limits.

Table 18 – CASE1.3: SHED AND LOAD RESULTS

Scenario	Best Shed [%]	Shed Feeder	Total power restored [MVA]	Total Power w/ SH [MVA]
CAP400	10	3	8.3923	6.9178
CAP500	5	3	9.5430	6.9178
CAP560	5	3	10.5110	6.9178
CAP600	0	0	10.8890	6.9178

SOURCE: The author (2020).

Table 19 – CASE1.3: FINAL CURRENT RESULTS

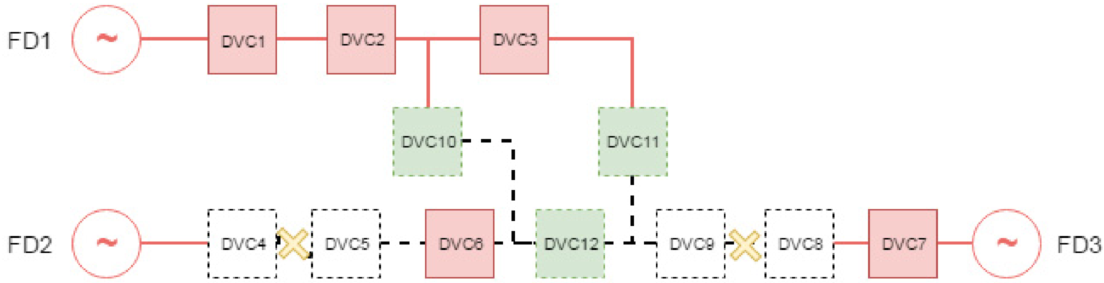
Scenario	Current Feeder 1 [A]	Current Feeder 2 [A]	Current Feeder 3 [A]
CAP400	0	256.149	389.4137
CAP500	0	256.149	477.9315
CAP560	0	256.149	552.41
CAP600	0	256.149	581.4842

SOURCE: The author (2020).

5.2.4 CASE1.4: Fault DVC4 – DVC5 and DVC8 – DVC9

The last test case simulates two simultaneous faults between the switches DVC4 and DVC5, and the other fault is between DVC8 and DVC9, creating a scenario to stress the resources from feeder 1, as there isn't any load prioritization the algorithm is free to select which block of load can be restored. To discuss on the results, it was prepared fours cases, but starting in 500 A until 700 A. The fault isolation is presented in FIGURE 23.

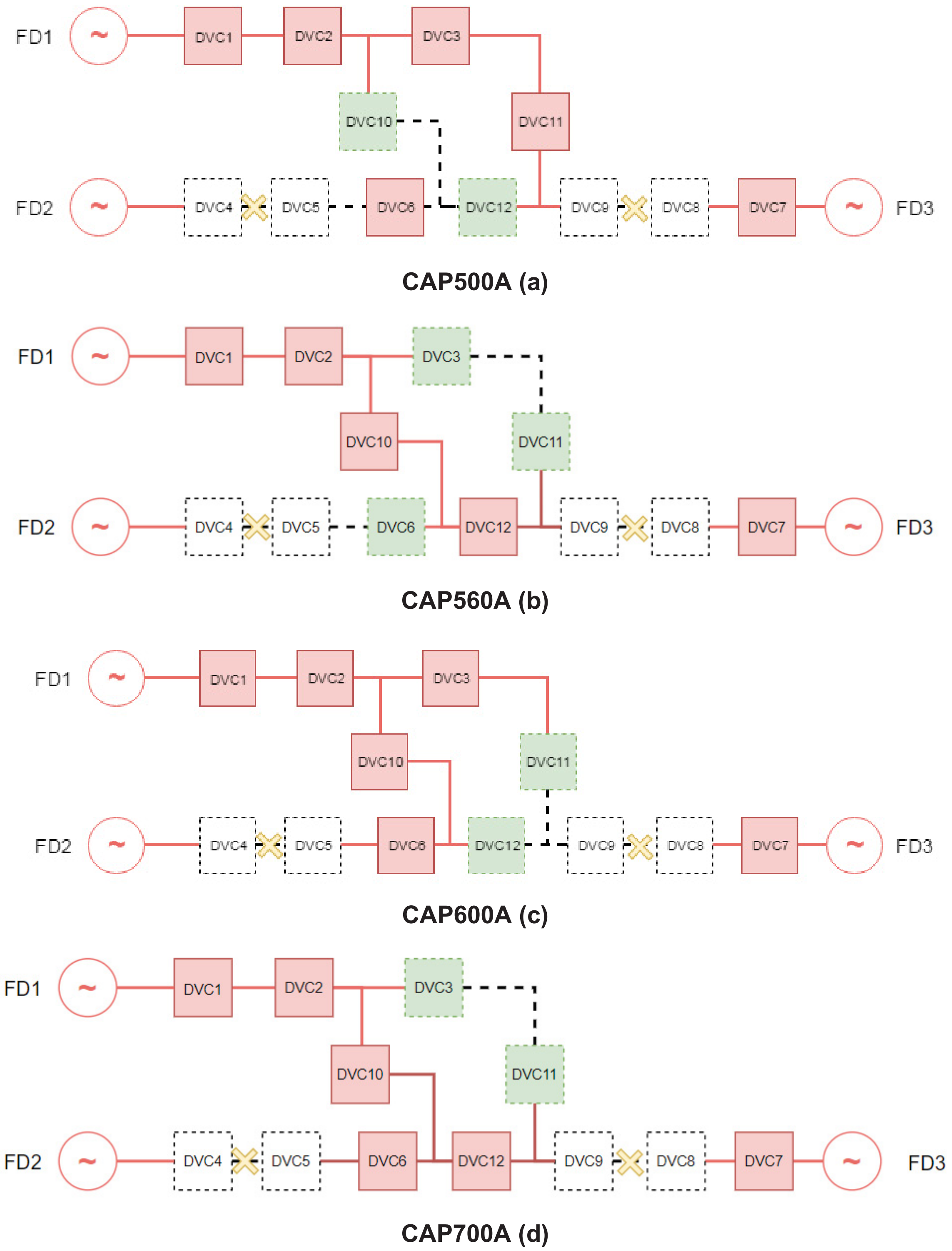
FIGURE 23 – CASE1.4: TOPOLOGY AFTER ISOLATION



SOURCE: The author (2020).

When the capacity is 500 A, the algorithm selected DVC11 to close (service restoration) because the feeder 3 block 3 has more load (3.4961 MVA) than the feeder 2 block 3 (3.005 MVA), so as in both cases it's not necessary a load reduction, the algorithm prefers the block with more load. When the capacity is increased (to 560 A) the DVC6 should be opened (load shedding) and the DVC10 and DVC12 should be closed to restore many loads is possible without exceed the capacity, and a 15% of load reduction should be applied. For 600 A to avoid a bigger shedding connecting the three de-energized feeders, the best choice was to restore all load in feeder 2 with a load shedding of 5% through the closing of DVC10. The last case with 700 A the DVC3 was opened (load shedding) and DVC10 and DVC12 was closed to restore the load of the three other blocks from feeder 2 and feeder 3. Furthermore, a load shedding should be applied to keep the system in a normative operation. All final topologies are demonstrated in FIGURE 24.

FIGURE 24 – CASE1.4: TOPOLOGY AFTER RECONFIGURATION. (A) 500A, (B) 560A, (C) 600A AND (D) 700A



SOURCE: The author (2020).

The Table 20 details the process time, all values are similar because the problem shows the same complexity for each capacity variation. One of the reasons that in 600 A case spent more time than the others, it could be because the line of search on QMatrix, where take a wrong way at the beginning and according the exploration phase the solution could be converged. Based on the capacity levels, the Table 21 shows the total load of the system after the final reconfiguration, which it was possible to keep 48% of the total load for 500 A, 53% for 560 A, 55% for 600 A and 65% for 700 A, where the pos-isolation represents 23% of the total. The final solution didn't extrapolate any current limit that was imposed as a parameter, according to the Table 22, and the final solution could avoid also the permanent parallelism and the voltage limits.

Table 20 – CASE1.4: TIME RESULTS AND NUMBER OF ITERATIONS

Scenario	Learning Time [s]	Number of iteration	Position found the best result
CAP500	10.0298	1000	191
CAP560	10.24	1000	214
CAP600	16.8714	1000	333
CAP700	6.4051	1000	134

SOURCE: The author (2020).

Table 21 – CASE1.4: SHED AND LOAD RESULTS

Scenario	Best Shed [%]	Shed Feeder	Total power restored [MVA]	Total Power w/ SH [MVA]
CAP500	0	0	7.091	3.5949
CAP560	15	1	7.9077	3.5949
CAP600	5	1	8.2396	3.5949
CAP700	15	1	9.639	3.5949

SOURCE: The author (2020).

Table 22 – CASE1.4: FINAL CURRENT RESULTS

Scenario	Current Feeder 1 [A]	Current Feeder 2 [A]	Current Feeder 3 [A]
CAP500	492.1438	0	53.325
CAP560	554.9565	0	53.325
CAP600	580.4924	0	53.325
CAP700	688.1384	0	53.325

SOURCE: The author (2020).

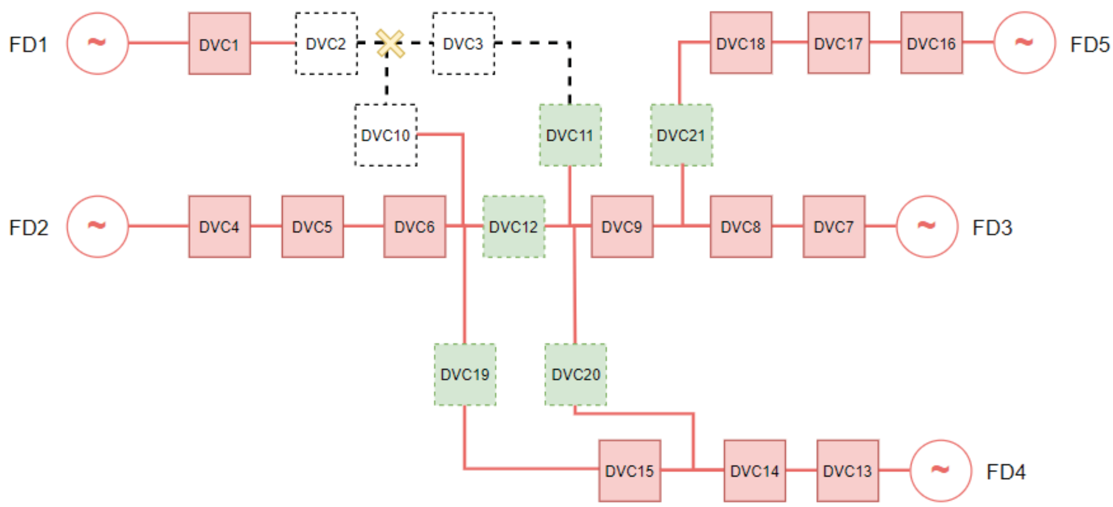
5.3 SIMULATION WITH FIVE-FEEDER TOPOLOGY

To stress the algorithm in complex scenarios, it's considered the five-feeder topology and the time process is relevant in this case, just the result. To analyze the behavior, it will be shown three different fault cases, in two distinct capacity configurations, 500 A and 600 A. A special case is created and discussed in the end of this section.

5.3.1 CASE2.1: Fault DVC2 – DVC3 – DVC10

The first test case with the five-feeder topology simulates a fault among the switches DVC2, DVC3 and DVC10, where the segment in front of DVC6, with 1.9344 MVA, is de-energized. The idea is to apply a simple fault in the smallest loaded feeder and analyze how will be the algorithm behavior to supply the feeder 1 block 3. To isolation topology is shown in FIGURE 25.

FIGURE 25 – CASE2.1: TOPOLOGY AFTER ISOLATION

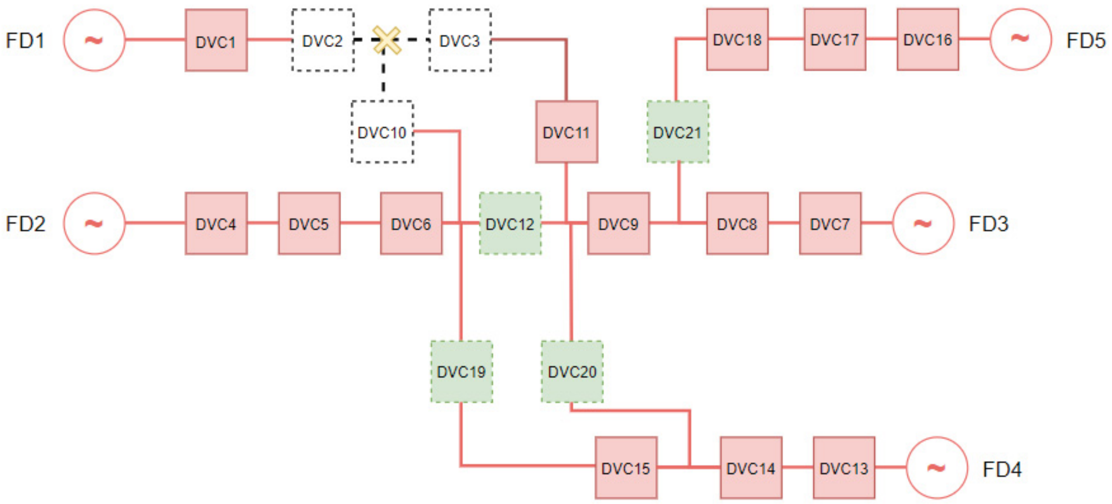


SOURCE: The author (2020).

When the capacity is 500 A, the algorithm selected the DVC11 to close (service restoration) because it's the unique route to be used in the reconfiguration, as shown in FIGURE 26. For the scenario with 560 A the same switching operation is execute, where the different is that in 500 A scenario should be executed a load shedding of 5% in feeder 3, and in the second case no load reduction was necessary.

The Table 23 details the process time, as the problem to be solved is the same, so the only time difference is related with system capacity, where more restrictive is the system more complicate is to find a solution, different of the results from the three-feeder topology, here the time consuming is much more perceptive. The 500 A case was necessary to increase the total number of iterations to find a better solution because in 10000 iteration the result was not enough, so it was increased the iterations for 50000, where the best configuration found was in position 19105.

FIGURE 26 – CASE2.1: TOPOLOGY AFTER RECONFIGURATION, WHEN CAPACITY 500A AND 600A



SOURCE: The author (2020).

Table 23 – CASE2.1: TIME RESULTS AND NUMBER OF ITERATIONS

Scenario	Learning Time [s]	Number of iteration	Position found the best result
CAP500	2273	50000	19105
CAP600	791	10000	1357

SOURCE: The author (2020).

Based on the capacity levels, the Table 24 shows the total load of the system after the final reconfiguration, which it was possible to keep 95% of the total load for 500 A and 96% for 600 A, where the pos-isolation represents 93% of the total. The final solution didn't extrapolate any current limit that was imposed as a parameter, according to the Table 25, and the final solution could avoid also the permanent parallelism and the voltage limits.

Table 24 – CASE2.1: SHED AND LOAD RESULTS

Scenario	Best Shed [%]	Shed Feeder	Total power restored [MVA]	Total Power w/ SH [MVA]
CAP500	5	3	25.358	24.7693
CAP600	-	-	25.6850	24.7693

SOURCE: The author (2020).

Table 25 – CASE2.1: FINAL CURRENT RESULTS

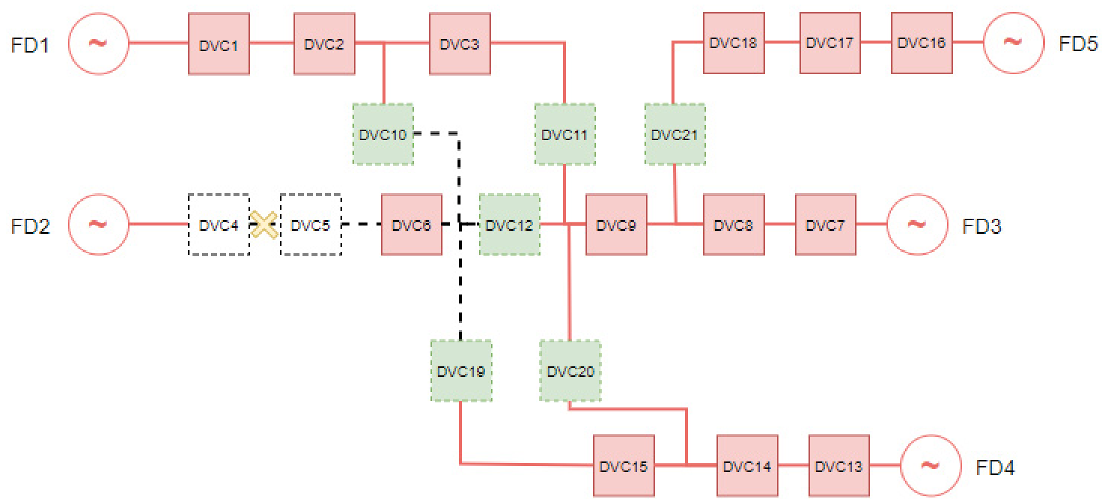
Scenario	Current Feeder 1 [A]	Current Feeder 2 [A]	Current Feeder 3 [A]	Current Feeder 4 [A]	Current Feeder 5 [A]
CAP500	74.0477	487.2990	452.7772	494.2075	416.7613
CAP600	74.0477	487.2990	503,0858	494.2075	416.7613

SOURCE: The author (2020).

5.3.2 CASE2.2: Fault DVC4 – DVC5

Different of the case above, now the idea is to prove how the algorithm can handle with more possibilities, where the fault (DVC4) in feeder 2 might be supported by three other feeders. Possibility one is through feeder 1 (DVC10) which is the most available feeder (425.59 A / 525.59 A), second option is the feeder 3 (DVC12) and the third is feeder 4 (DVC19), but in the last to options the availability is less than the first option (67.32 A / 167.32 A) (5.80 A / 105.80 A). The isolation state is presented in FIGURE 27.

FIGURE 27 – CASE2.2: TOPOLOGY AFTER ISOLATION



SOURCE: The author (2020).

When the capacity is 500 A, the algorithm selected the DVC6 to open (load shedding via block) and DVC10 to close (service restoration) because the feeder 1 is more available (276.79 A) than the feeder 3 (67.32 A). Then, just the block3 (231.15 A) from feeder 2 might be restored without any load shedding. For the scenario with 600 A to supply all de-energized load a special configuration was executed, where the feeder 1 block 3 was transferred to feeder 3 (without any load reduction), so the feeder 1 could have enough capacity to supply the two blocks from feeder 2 and also without any load shedding. Both reconfiguration topology can be seen in FIGURE 28.

The Table 26 details the process time, as the problem to be solved is the same, so the only time difference is related with system capacity, where more restrictive is the system more complicate is to find a solution, the same explanation from the previous scenarios is applied in this case, the difference is in the 600 A simulation, where the number of iterations to satisfy a good solution could be increased for 20000.

Based on the capacity levels, the Table 27 shows the total load of the system after the final reconfiguration, which it was possible to keep 88% of the total load for 500 A and 95% for 600 A, where the pos-isolation represents 76% of the total. The final solution didn't extrapolate any current limit that was imposed as a parameter, according to the Table 28, and the final solution could avoid also the permanent parallelism and the voltage limits.

Table 26 – CASE2.2: TIME RESULTS AND NUMBER OF ITERATIONS

Scenario	Learning Time [s]	Number of iteration	Position found the best result
CAP500	2012	50000	45566
CAP600	1484	20000	17413

SOURCE: The author (2020).

Table 27 – CASE2.2: SHED AND LOAD RESULTS

Scenario	Best Shed [%]	Shed Feeder	Total power restored [MVA]	Total Power w/ SH [MVA]
CAP500	-	-	23.3741	20.3688
CAP600	-	-	25.4110	20.3688

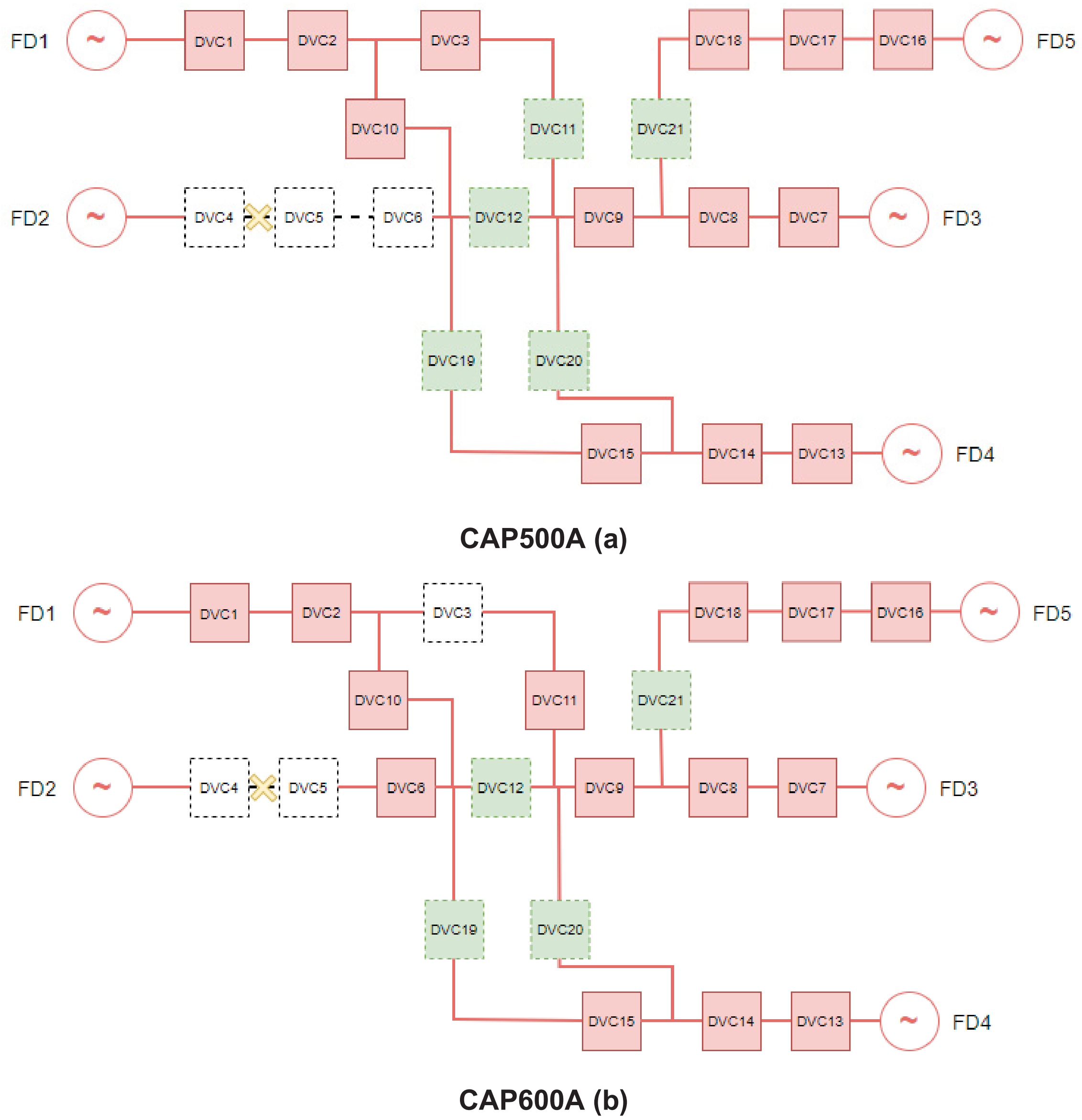
SOURCE: The author (2020).

Table 28 – CASE2.2: FINAL CURRENT RESULTS

Scenario	Current Feeder 1 [A]	Current Feeder 2 [A]	Current Feeder 3 [A]	Current Feeder 4 [A]	Current Feeder 5 [A]
CAP500	454.3600	0	432.6819	494.2075	416.7613
CAP600	540.6408	0	503.0858	494.2075	416.7613

SOURCE: The author (2020).

FIGURE 28 – CASE2.2: TOPOLOGY AFTER RECONFIGURATION. (A) 500A AND (B) 600A



SOURCE: The author (2020).

5.3.3 CASE2.3: Fault DVC1 – DVC2 AND DVC7 – DVC8

The third test case simulates two simultaneous faults between the switches DVC1 and DVC2 (de-energizing a total of 2.9017 MVA), and the other fault is between DVC7 and DVC8 (de-energizing a total of 5.6248 MVA). The objective is to provide a complexity situation for the algorithm, where the system has less possibilities to restore all load, once two feeders are lost. The isolation state is shown in FIGURE 29.

When the capacity is 500 A, the algorithm selected the DVC9 to open (load shedding via block) and DVC21 to close (service restoration), losing the most available feeder the second feeder if more capacity is the feeder 5 (83.24 A), which just have the possibility to support the feeder 3 block 2 (110.42 A) and to avoid the overload, a shed of 10% should be applied. For the scenario with 600 A the switching action was the most complex than the other cases, where the DVC9 was opened to split the load and to create a route between feeder 4 and feeder 1. The second switch opened was DVC15 to provide a load reduction in feeder 4, which receive the feeder 3 – block 3 and feeder 1 – block 3. The switches closed were, DVC21 to supply the feeder 3 – block 2, and DVC20 and DVC11 to supply the last blocks from feeder 3 and feeder 1. The final topology for both cases is shown in FIGURE 30.

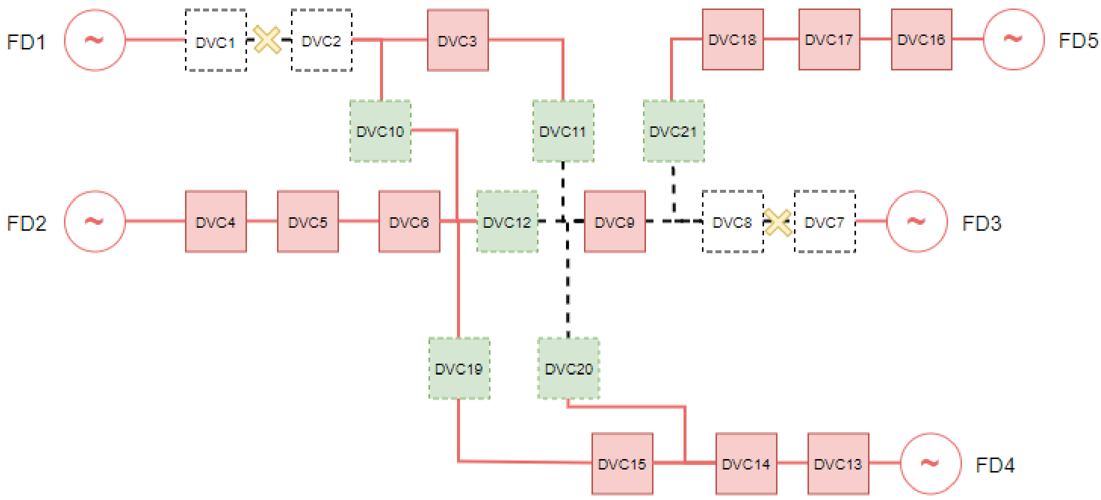
The Table 29 details the process time, as cited above this is the worst scenario created to test the algorithm, so when the capacity was 500 A, the number of iterations to run the process should be increased to 100000 and the position that found the best solution was 79556. Otherwise, when the capacity was 600 A the time was near by the previous case.

Table 29 – CASE2.3: TIME RESULTS AND NUMBER OF ITERATIONS

Scenario	Learning Time [s]	Number of iteration	Position found the best result
CAP500	4719	100000	79556
CAP600	1742	20000	4156

SOURCE: The author (2020).

FIGURE 29 – CASE2.3: TOPOLOGY AFTER ISOLATION



SOURCE: The author (2020).

Based on the capacity levels, the Table 30 shows the total load of the system after the final reconfiguration, which it was possible to keep 71% of the total load for 500 A and 79% for 600 A, where the pos-isolation represents 68% of the total. The final solution didn't extrapolate any current limit that was imposed as a parameter, according to the Table 31, and the final solution could avoid also the permanent parallelism and the voltage limits.

Table 30 – CASE2.3: SHED AND LOAD RESULTS

Scenario	Best Shed [%]	Shed Feeder	Total power restored [MVA]	Total Power w/ SH [MVA]
CAP500	10	5	18.9280	18.1772
CAP600	-	-	21.1013	18.1772

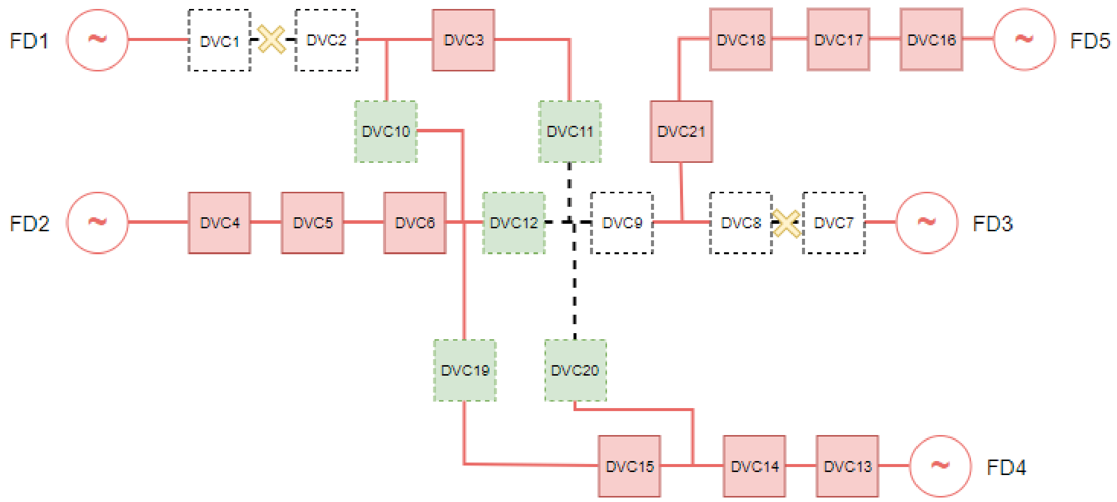
SOURCE: The author (2020).

Table 31 – CASE2.3: FINAL CURRENT RESULTS

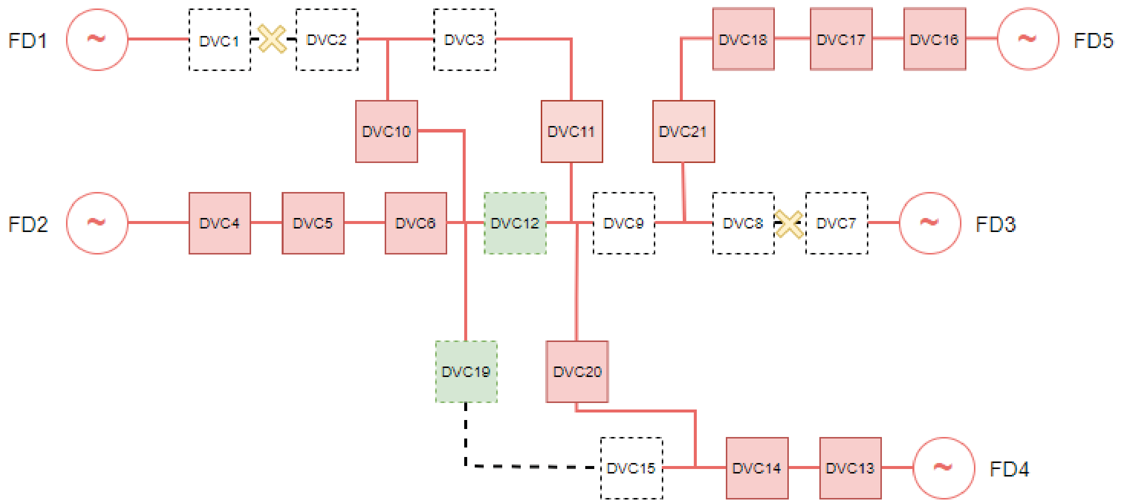
Scenario	Current Feeder 1 [A]	Current Feeder 2 [A]	Current Feeder 3 [A]	Current Feeder 4 [A]	Current Feeder 5 [A]
CAP500	-	487.2990	-	494.2075	474.4660
CAP600	-	565.6975	-	530.2973	527.1844

SOURCE: The author (2020).

FIGURE 30 – CASE2.3: TOPOLOGY AFTER RECONFIGURATION. (A) 500A AND (B) 600A



CAP500A (a)



CAP600A (b)

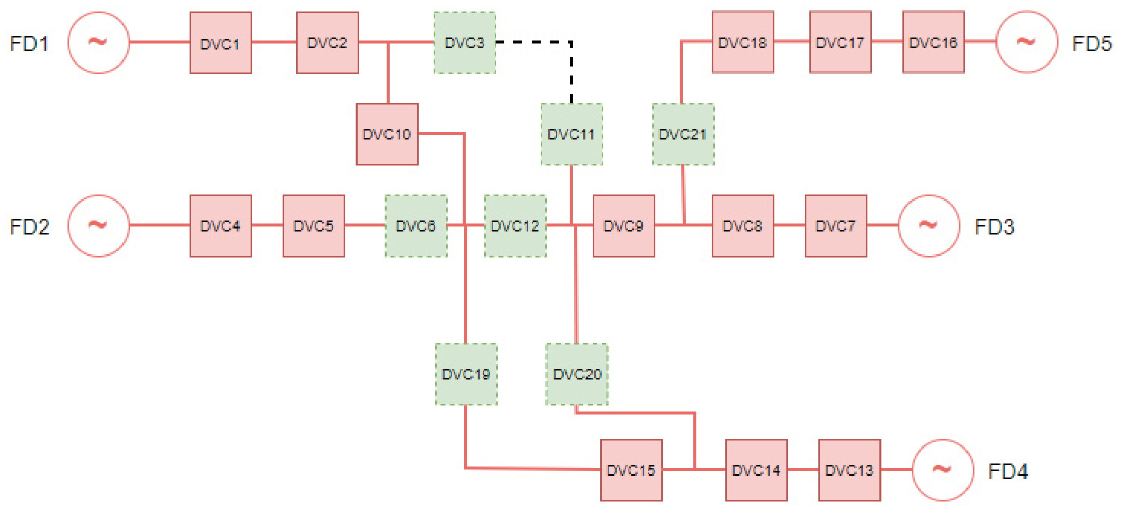
SOURCE: The author (2020).

5.3.4 CASE2.4: Overload on feeder 2 and feeder 4

In a different approach, the case in this sections is related with an overload on feeder 2 and feeder 4 instead of an unexpectable outage, where the idea is a sudden increase of load on the system for both feeders and the algorithm should reconfigure the system or apply the load shedding to avoid the distribution system operates out of the normative limits.

The final topology is presented in FIGURE 31 where the algorithm selects the small reconfiguration between feeder 1 and feeder 2, to resolve the overload on feeder 2. As the feeder 1 is the most available feeders in this network, the decision maker understood that is preferable to discard the block3 (0.9152 MVA) and restore the feeder 2 – block 3 (3.005 MVA). In feeder 4, the idea was a load shedding of 10% instead of transferring the load, once the feeder 2 and feeder 3 has not any capacity to support the overload in feeder 4.

FIGURE 31 – CASE2.4: TOPOLOGY AFTER OVERLOAD, WHEN CAPACITY 450A



SOURCE: The author (2020).

The Table 32 details the process time, where the QMatrix is considered complete, once there isn't any fault to decrease the search space. The total time was not worse than the previous case because of the problem complexity, where two faults in restrictive system is more complicated than two overloads. Based on the capacity levels, the Table 33 shows the total load of the system after the final reconfiguration and load shedding, which decreased 6% of the load to return the distribution network operative again. The final solution didn't extrapolate any current limit that was imposed as a parameter, according to the Table 34, and the final solution could avoid also the permanent parallelism and the voltage limits.

Table 32 – CASE2.4: TIME RESULTS AND NUMBER OF ITERATIONS

Scenario	Learning Time [s]	Number of iteration	Position found the best result
CAP450	3781	100000	23489

SOURCE: The author (2020).

Table 33 – CASE2.4: SHED AND LOAD RESULTS

Scenario	Best Shed [%]	Shed Feeder	Total power restored [MVA]
CAP450	10	4	25.146

SOURCE: The author (2020).

Table 34 – CASE2.4: FINAL CURRENT RESULTS

Scenario	Current Feeder 1 [A]	Current Feeder 2 [A]	Current Feeder 3 [A]	Current Feeder 4 [A]	Current Feeder 5 [A]
CAP450	383.9562	256.1490	432.6819	444.7867	416.7613

SOURCE: The author (2020).

5.4 COMPARATIVE ANALYSIS

To compare the solution with other techniques developed and already publish, the Table 35 shows the results of two different methods: one centralized and another distributed. The strategy of the first one is based on a Binary Particle Swarm Optimization (BPSO) for service restoration and an Optimum Power Flow (OPF) for load shedding, in a sequential way the use of these algorithms (Ferreira; Siebert; Aoki, 2014). The second, a decentralized solution, each agent, installed in the switches, has operational rules-of-thumb (TORRES et al., 2018). For the analysis comparison, it was selected two different cases (1.1 and 1.4) with the three-feeder system and the same capacity for both cases (500 A). The first case is applied a single fault in DVC1 and the second case a double fault in DVC4 and DVC8 bringing on just the feeder1 as resource to transfer the load, a summary of the results can be seen Table 35.

In the first scenario the BPSO+OPF choose the feeder2 as resource, but compared with feeder3, the feeder2 has 11% more load, so it's not a good choice once the feeder3 has more capacity and the load reduction could be smaller, against the 23% applied. To prove this statement, it's possible to see the results for Distributed and RL method, where both selected the feeder3 as resource and the load curtailment was 14% and 15% respectively, resulting an increase of load in 6%.

Table 35 – RESULTS OF OTHER METHODS FOR COMPARISON

Case	Functionality	Capacity [A]	Trigger	Switches changed	Best Shed [%]	Feeder Selected to Shed	Total Final Load [MVA]
BPSO + OPF Method							
1.1	Self-Healing	500	DVC1	DVC10 - CLOSED	23	2	11.99
1.4	Self-Healing	500	DVC4 and DVC8	DVC11 - CLOSED DVC12 - CLOSED	47	1	6.20
Distributed Method							
1.1	Self-Healing	500	DVC1	DVC11 - CLOSED	14	3	12.83
1.4	Self-Healing	500	DVC4 and DVC8	DVC10 - CLOSED DVC11 - CLOSED	47	1	6.20
Reinforcement Learning Method							
1.1	Self-Healing	500	DVC1	DVC11 – CLOSED DVC2 - OPENED	5	3	12.548
1.4	Self-Healing	500	DVC4 and DVC8	DVC11 - CLOSED	0	0	7.091

SOURCE: The author (2020).

The second scenario evidence the benefits to have the Self-Healing and the Load Shedding solved at the same moment. As the BPSO+OPF and Distributed techniques first find the best switching sequence and after the load to be reduced, both tried to restore the maximum load possible with the service restoration, but once the system was limited with 500 A a big shed must be executed (47%) to put the system in a normal operation. However, with a different perspective, the RL method just close one tie-switch (DVC11) because during the process the algorithm understood the disadvantage to close DVC10. Thus, closing just one device it was not necessary to execute any load reduction and the final system load was 12.5% more than the other two methods.

5.5 FINAL DISCUSSION

The first discussion is about the result from the policy analysis, to have a better comprehension the difference for bad solution between the greed and the others is related with the concepts of explore and exploit, the greed solution focus in a exploit approach, where selecting the first choice the greed approach will try to extract the maximum value all the time for the path selected without see other possibilities. However, the other two approaches consider randomize influence, so in some iterations the algorithm change the exploit concept for a explore idea, to change the direction and try to find another good solution, where in this last case, it’s open more possibilities to find solution.

Considering the articles Ribeiro; et al. (2017), Das et al. (2013) and Pal et al. (2010) those applying RL in SH problems, the better approach for the policy was the one that balances the explore and exploit search, in other worlds, it was used in the development of the algorithm the ε -Greed policy to have 10% of probability in explore and 90% to exploit the selected route to find the solution. According to the results to select the policy, it is important to highlight that is not in all time a good solution could be found. For example, in a single fault the ε -greed reached 4% of wrong solution, and for simultaneous fault 1% could not find the better topology that could keep or increase the load after the isolation. Furthermore, compared with the other two policies, the ε -Greed demonstrated more consistent in its results, even in the single fault which it found 3% of worse solutions than the others.

In general, all simulated cases could be resolve satisfactorily from a solution where more load could be restored. Some of the cases as 1.1 (500 A) and 2.3 (600 A) could not find the global solution, i.e., the final load could be increased if the algorithm had selected other actions, but as the algorithm start from a random action and learns from the actions chosen, some ways cannot be the best one. Moreover, the algorithm could handle in two different scenarios, one related with three feeders, which comprehends most of the cases in Brazil, and the second scenario considered five feeders to bring more complexity for the algorithm. The five-feeder case is considered more complicated because the number of switches and the number of possibilities to transfer load between other feeders. For more details the Table 36 presents a summary of all results to be compared among them.

Table 36 – SCENARIO RESULTS IN COMPARISON ANALYSIS

Case	Functionality Trigger	Capacity [A]	Device Trigger	Switches changed	Best Shed [%]	Feeder Selected to Shed	Final Load [MVA]
1.1	Self-Healing	500	DVC1	DVC3 - OPENED DVC11 - CLOSED	5	3	12.54
	Self-Healing	560		DVC3 - OPENED DVC10 - CLOSED DVC11 - CLOSED	5	2	13.52
	Self-Healing	600		DVC11 - CLOSED	---	---	13.89
1.2	Self-Healing	500/560/ 600	DVC5	DVC10 - CLOSED	---	---	12.82
1.3	Self-Healing + Load Shedding	400	DVC1 and DVC6	---	10	3	8.39
	Self-Healing	500		DVC3 - OPENED DVC11 – CLOSED	5	3	9.54
	Self-Healing	560		DVC11 - CLOSED	5	3	10.51
	Self-Healing	600		DVC11 - CLOSED	---	---	10.88
1.4	Self-Healing	500	DVC4 and DVC8	DVC11 – CLOSED	---	---	7.09
	Self-Healing	560		DVC3 – OPENED DVC6 - OPENED DVC10 – CLOSED DVC12 – CLOSED	15	1	7.90
	Self-Healing	600		DVC10-CLOSED	5	1	8.23
	Self-Healing	700		DVC3 – OPENED DVC10 – CLOSED DVC12 - CLOSED	15	1	9.63
2.1	Self-Healing	500	DVC2	DVC11 - CLOSED	5	3	25.35
	Self-Healing	600			---	---	25.68
2.2	Self-Healing	500	DVC4	DVC6 – OPENED DVC10 - CLOSED	---	---	23.37
	Self-Healing	600		DVC3 – OPENED DVC10 – CLOSED DVC11 – CLOSED	---	---	25.41
2.3	Self-Healing	500	DVC1 and DVC7	DVC9 – OPENED DVC21 – CLOSED	10	5	18.92
	Self-Healing	600		DVC15 – OPENED DVC9 – OPENED DVC21 – CLOSED DVC11 – CLOSED DVC20 - CLOSED	---	---	21.10
2.4	Load Shedding	450	Feeder4 and Feeder 2	DVC3 – OPENED DVC6 – OPENED DVC10 - CLOSED	10	4	25.14

SOURCE: The author (2020).

To understand about the system complexity, the Table 37 shows all simulated cases in the worst capacity scenario (500 A), with the number of possible switches to commute, the average total process time for each simulation and the QMatrix size, which is responsible for the most of the time process increasing. It's possible to conclude that the problem is exponential based in the number of switches to manipulate. Event tough the case 2.4 has a bigger time, but a QMatrix smaller than the other five-feeder cases, the reason is because the number of faults applied at the same time, where with more fault more restrict the system is. Thus, the number of switches is not the unique variable that influences the total time process, the number of faults and the interconnections can directly increase the time.

Table 37 – ANALYSIS OF SYSTEM COMPLEXITY FOR EACH CASE

Case (500A)	Number of switches	Average Process Time [s]	MQ size
1.1	8	62.0739	256x20
1.2	7	26.2447	128x18
1.3	5	8.7008	32x14
1.4	6	10.0298	64x16
2.1	13	2273	8192x30
2.2	15	2012	32768x34
2.3	14	4719	16386x32

SOURCE: The author (2020).

Furthermore, as demonstrated in the articles Ferreira; Siebert; Aoki (2014) and Torres et al. (2018) the process to execute first the SH and after the LS cannot determine a good solution, once the first algorithm (SH) optimizes the switching actions to minimize the reconfiguration losses, and if there isn't any possibilities without trigger an overload, the second algorithm (LS) should be trigger to reduces the load and "correct" the reconfiguration from the first algorithm. However, when the reconfiguration and the load curtailment is applied at the same moment to take a decision, the results shows more consistent the others, it isn't in all time the proposed algorithm could find the better solution, but the results for all comparative tests were near for the best result.

As a bonus simulation to focus on the overload trigger, the cases 1.3 and 2.4 presents an exceed capacity limit besides the outage on the distribution system. The case 1.3 considers at the same moment on overload and an outage on the network to be solved and the 2.4 case only an overload in two different feeders (2 and 4) to trigger the process. The algorithm executed the load shedding in both cases, and specially for the five-feeder scenario, instead of having a load reduction, the process preferred to transfer part of the feeder 2 to feeder 1 to keep with more load.

6. FINAL REMARKS

In this chapter is discussed how the objectives were reached, highlighting the main results and the comparison with other works. Furthermore, the contributions for further studies to improve the beginning of this line of research.

6.1 CONCLUSION

The proposed method successfully solved the main objective proposed for this work to develop an intelligent control algorithm to optimize multiples Smart Grid functionalities. To satisfy the purpose of this thesis, it was studied initially about the Machine Learning area which brings a set of different methods to solve the most varied problems. As the problem is related with electric power system, the method should comprehend a way to model the distribution network and its constraints, consider the inclusion of several functionalities, learn according the actions taken and have critical analysis according the results for each iteration. The best method selected was the Reinforcement Learning through the Q-Learning algorithm which could model the Self-Healing and the Load Shedding problematic in a way to solve at the same time both functionalities.

To reach the results aimed, the Q-Learning algorithm, in special, the QMatrix was built in according to recreate a distribution network, where the columns represents the switch actions and the load reduction percentual, the lines reproduces the system topology. When the algorithm cruces the column with the lines, the result is to obtain the current topology being changed by the switching commands or load shedding, where these actions are selected by the policy. The policy selected is important to lead reinforcement learning to explore (giving more randomness of the action selection) or to exploit (forcing the algorithm to go deeper in the same line of strategy, with less variation). As explained before, depending of the problem the balance of explore or exploit should be analyzed to contributed for good results, in other words, using the standard greed approach the objective is to exploit the system at your maximum. In the other hand, the ε-greed and randomized technique mixed the explore approach to avoid some wrong lines of strategy in the middle of the process, so for the algorithm developed the ε-greed approach was chosen. The method can be triggered from a fault, overload or a mixed of both.

To test the proposed algorithm, it was used a real urban distribution network, where it's composed by five interconnected feeders, two substations, fifteen normally closed and six normally opened switches. It was created two types of approach, the first is using three feeders interconnected and the second using the complete network. The idea for the first approach is to prove the technique in a faster time and validate the policy selected. For the second approach is to validate the methodology in a bigger system, where the idea is to prove the final solution for Self-Healing and Load Shedding and not validate the process time.

It's important to highlight that the algorithm developed in this thesis has a different approach compared with other algorithm to resolve the SH and LS problems as cited in Arefifar et al. (2013), Botea et al. (2012), Cavalcante et al. (2016) and Sampaio et al. (2016), where there are works based on Multi-agent system and others in a optimization approach. Thus, this work shows a decision maker to solve the Self-Healing and Load Shedding problems based on influences from environment created. The Q-Learning algorithm was adapted to reproduces a distribution network with its constraints (voltage, current and parallelism) where the actions selected (switching and load reduction) could directly influences the distribution system in a way to exploit more the rewards. In contrast with the article Das et al. (2013), the QMatrix was modeled different, where in this works the objective is to reproduce the topology in each line and the columns are the actions to be taken to change the topology behavior (switch configuration and load).

According to the simulations in all cases was possible to find a solution for the system reconfiguration and overload. In some cases, the global solution couldn't be found, but an increase of load was perceived after the pos-isolation state. The most interesting cases were 1.3, 1.4, 2.3 and 2.4, where the first one is applied a fault and decreased the load capacity (400 A), so the system should handle with two different problems at the same time. The result presented was a system in pos-isolation topology with a load shedding on the feeder three because the overload, in line with the previous scenario, the case 2.4 is only applied an overload, so reduction was executed just on feeder four which exceeded the current limits. The 1.4 scenario brought how the capacity in the system can change the solution in terms of topology and load reduction. In four different capacities, the reconfiguration and the shedding were different. The last notable scenario is the 2.4 which considers two simultaneous faults in the five-feeder topology, so this is the most complexity test and the algorithm could find a good solution to restore the most possible load in the de-energized areas. Beside the simulations, the algorithm was tested against two other different techniques, one is a BPSO + OPF approach and the other a Distributed method. The first comparison scenario, the developed algorithm was not better than the distributed approach, but in the second scenario, the proposed algorithm could restore more load than the other two methods.

On the other hand, the proposed method takes some disadvantage, the first is that the algorithm needs from the utility a complete information from the distribution network through GIS application and a certain field automation level to run the power flow. Another point is related when the distribution network is highly interconnected or has several telecontroled reclosers installed in the network, which the complexity for QMatrix is bigger enough to enter in the “curse of dimensionality”. At the end, the three policies testes reached at the least one bad solution (among the 100 tests) that leads for a wrong configuration, it means that the final topology has less load then the pos-isolation state.

6.2 FUTURE RESEARCH

For future researches there are four possibilities to extend the methodology developed. The first is related to include more complexity in the system through the inclusion of micro-grids, distributed energy resources, tap controller, etc. The second line of research could be based on improvements in the Reinforcement Learning for a new algorithm in Machine Learning. The third enhancement is related with the load prioritization, and the last is to improve the reward equation to include other analysis than the load maximization.

As demonstrated in this thesis, the actions related on QMatrix are based on the switch position (Self-Healing) and load shedding percentual (avoid overload after a reconfiguration), so including more variable as actions, for instance, tap position or capacitor steps to act on the distribution network, the result could contribute for a different perspective as the volt/var control, together with the other two functionalities. Moreover, for distributed generation and microgrids could control the injection of power or the grid connection when a fault occurs. However, the normal reinforcement learning cannot support a big amount of actions to be selected once the system becomes more interconnected and automated, so the second line of research is to go deeper in other techniques to avoid the “curse of dimensionality”, which it possible to verify from the discussion in Table 37 that compares the number of switches, faults and QMatrix size, for further researches the Deep Reinforcement Learning could be the next step. Another possibility to include more variable is to include the equipment actions into FPO, for example, tap, and keep the RL to take of the other variable to find the best solution.

In terms of load prioritization and as explained in the chapter 2.3, once the control system can manipulate small parts of the distribution network (the fuse of transformer installed in medium voltage to low voltage) or the final consumers (via AMI) the algorithm could be improved to select the transformer or consumers to be turned off. This work selects the percentual to be reduced, but it’s not taking in account if this percentual is enough based on important consumer, so this might impact in the action selected at the end.

About the reward equation, where in this work considers just the increase of load, it should be enhanced to consider the costs to operate a switch determined from the equipment life cycle. The algorithm could select the better switch action when both solutions was guided for a same load maximization instead of a random selection. Moreover, other load analysis could be done, one in terms of the cost of load shedding, or the impact (in profit) that the utility is not receiving money from the energy not distributed. The second is the losses from the system, once some reconfiguration can become the feeder long enough to provide more losses.

6.3 PUBLICATIONS

6.3.1 Peer-reviewed journals

FERREIRA, Lucas R.; AOKI, Alexandre R; LAMBERT-TORRES, Germano. A reinforcement learning approach to solve service restoration and load management simultaneously for distribution networks. **IEEE Access**, v. 7, p. 145978-145987, 2019.

TORRES, B. S.; FERREIRA, L. R.; AOKI, A. R. Distributed Intelligent System for Self-Healing in Smart Grids. **IEEE Transactions on Power Delivery**, v. 33, n. 5, p. 2394–2403, 2018.

6.3.2 Conference Proceedings

TORRES, Bruno S.; FERREIRA, Lucas R; AOKI, Alexandre R. A Study of Distribution System Self-Healing Considering Intelligent Approaches. In: 2019 20th International Conference on Intelligent System Application to Power Systems (ISAP). IEEE, 2019. p. 1-8.

RETORTA, Fábio S.; FARINA, Annelise S.; BLOCK, Pedro B.; Ferreira, Lucas R.; et al. Net Metering Analysis of Time-of-Use Tariff in Distributed Generation with Storage System. 2016 CIGRE C4 International Colloquium on EMC, Lightning and Power Quality Considerations for Renewable Energy Systems. CIGRE, 2016.

6.3.3 Master Thesis - Auxiliary Supervisor

TORRES, Bruno S. **Proposta de um Sistema Inteligente Distribuído para Autorrecuperação de Redes Elétricas Inteligentes**. 2016. Dissertação (Mestrado Profissional em Desenvolvimento de Tecnologia). Institutos Lactec.

REFERENCES

AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA. **Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional**: **PRODIST Módulo 8 – Qualidade da Energia Elétrica**. 2012.

AKHTER, M. N.; MEKHILEF, S.; MOKHLIS, H.; SHAH, N. M. Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques. **IET Renewable Power Generation**, v. 13, n. 7, p. 1009–1023, 2019.

ALAM, M. R.; MUTTAQI, K. M.; BOUZERDOUM, A. Evaluating the effectiveness of a machine learning approach based on response time and reliability for islanding detection of distributed generation. **IET renewable power generation**, v. 11, n. 11, p. 1392–1400, 2017.

AMIN, M. Toward a Secure and Smart Self-Healing Grid. 1998. Disponível em: <http://massoud-amin.umn.edu/presentations/CINSI_01-27-1998_RAC.pdf>.

AMIN, S. M.; WOLLENBERG, B. F. Toward a Smart Grid. **IEEE Powe & Energy Magazine**, v. 3, n. 5, p. 34–41, 2005.

AOKI, K.; KUWABARA, H.; SATOH, T.; KANEZASHI, M. Outage State Optimal Load Allocation by Automatic Sectionalizing Switches Operation in Distribution Systems. **IEEE Transactions on Power Delivery**, v. 2, n. 4, p. 1177–1185, 1987.

AOKI, K.; NARA, K.; ITOH, M.; SATOH, T.; KUWABARA, H. A new algorithm for service restoration in distribution systems. **IEEE Transactions on Power Delivery**, v. 4, n. 3, p. 1832–1839, 1989.

APONTE, E. E.; NELSON, J. K. Time Optimal Load Shedding for Distributed Power Systems. **IEEE Transactions on Power Systems**, v. 21, n. 1, p. 269–277, 2006.

AREFIFAR, S. A.; MOHAMED, Y. A. I.; EL-FOULY, T. H. M. Comprehensive Operational Planning Framework for Self-Healing Control Actions in Smart Distribution Grids. **IEEE Transactions on Power Systems**, v. 28, n. 4, p. 4192–4200, 2013.

BELLMAN, R. **Project RAND: An introduction to the Theory of Dynamic Programming**. RAND CORP SANTA MONICA CA, 1953.

BELLMAN, R. **The Theory of Dynamic Programming**. Bulletin of the American Mathematical Society, 1954.

BELLMAN, R. Dynamic Programming and Lagrange Multipliers. **Proceedings of the National Academy of Sciences of the United States of America**, v. 42, n. 10, p. 767–769, 1956.

BELLMAN, R. A Markovian Decision Process. **Journal of Mathematics and Mechanics**, v. 6, 1957.

BERTSEKAS, D. P. Neuro-Dynamic Programming: An Overview. **Proceedings of 1995 34th IEEE Conference on Decision and Control**. p.560–564, 1995.

BOROUJENI, Z.; MOHAMMADI, M.; NEUMANN, D.; GOEHRING, D.; ROJAS, R. Autonomous Car Navigation using Vector Fields. IEEE Intelligent Vehicles Symposium. **Anais...** Changshu, China. p.794–799, 2018.

BOTEA, A.; RINTANEN, J.; BANERJEE, D. Optimal Reconfiguration for Supply Restoration with Informed A* Search. **IEEE Transactions on Smart Grid**, v. 3, n. 2, p. 583–593, 2012.

BROWN, R. E. Impact of Smart Grid on Distribution System Design. **IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century**, p. 1–4, 2008.

BROWNLEE, J. A Tour of Machine Learning Algorithms. Disponível em: <https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>. Acesso em: 12/9/2017.

CARVALLO, A.; COOPER, J. **The Advanced Smart Grid - Edge Power Driving Sustainability**. Artech House, 2011.

CAVALCANTE, P. L.; LÓPEZ, J. C.; FRANCO, J. F.; et al. Centralized Self-Healing Scheme for Electrical Distribution Systems. **IEEE TRANSACTIONS ON SMART GRID**, v. 7, n. 1, p. 145–155, 2016.

CECATI, C.; MOKRYANI, G.; PICCOLO, A.; SIANO, P. An Overview on the Smart Grid Concept. IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society. **Anais...** Glendale, AZ, USA. p.3322–3327, 2010.

CHEN, K.; WU, W.; ZHANG, B.; SUN, H. Robust restoration decision-making model for distribution networks based on information gap decision theory. **IEEE Transactions on Smart Grid**, v. 6, n. 2, p. 587–597, 2015. Disponível em: <http://ieeexplore.ieee.org/abstract/document/6954573/>.

CHENG, L.; YU, T. A new generation of AI: A review and perspective on machine learning technologies applied to smart energy and electric power systems. **International Journal of Energy Research**, v. 43, n. 6, p. 1928–1973, 2019.

CLELLAND, P.; FINK, L.; HAPP, H.; et al. POWER SYSTEM RESTORATION - A TASK FORCE REPORT. **IEEE Transactions on Power Systems**, v. PWRS-2, n. 2, p. 271–277, 1987.

DAS, SANJOY; BOSE, S.; PAL, SIDDHARTHA; et al. Dynamic Reconfiguration of Shipboard Power Using Reinforcement Learning. **IEEE Transactions on Power Systems**, v. 28, n. 2, p. 669–676, 2013. Disponível em: <http://ieeexplore.ieee.org/abstract/document/6305493/>.

DEVANAND, P.; RASTOGI, R.; AHMAD, S.; et al. Advanced Distribution Management System: Improving Distribution Efficiency Through an Integrated Approach. **IEEE Power and Energy Magazine**, v. 18, p. 55–62, 2020. IEEE.

DJAPIC, P.; RAMSAY, C.; PUDJIANTO, D.; et al. Taking an Active Approach. **IEEE Power and Energy Magazine**, v. 5, n. 4, p. 68–77, 2007.

DU, P. Ensemble Machine Learning-Based Wind Forecasting to Combine NWP Output with Data from Weather Station. **IEEE Transactions on Sustainable Energy**, v. 10, n. 4, p. 2133–2141, 2019.

DUGAN, R. C. **Reference Guide: The Open Distribution System Simulator (OpenDSS)**. EPRI, 2013.

OPERADOR NACIONAL DO SISTEMA ELÉTRICO. **Submódulo 11.4 - Sistemas especiais de proteção**, 2009. Disponível em: <http://www.ons.org.br/%2FProcedimentosDeRede%2FMódulo 11%2FSubmódulo 11.4%2FSubmódulo 11.4_Rev_1.0.pdf>.

U.S. Department of Energy. **Voices of Experience: Insights into Advanced Distribution Management Systems**. National Renewable Energy Laboratory, 2015. Disponível em: <http://energy.gov/oe/downloads/voices-experience-insights-advanced-distribution-management-systems-february-2015>.

FARANDA, R.; PIEVATOLO, A.; TIRONI, E. Load Shedding : A New Proposal. **IEEE Transactions on Power Systems**, v. 22, n. 4, p. 2086–2093, 2007.

FERNANDES, T. S. P.; LENZI, J. R.; MIKILITA, M. A. Load Shedding Strategies Using Optimal Load Flow With Relaxation of Restrictions**. IEEE Transactions on Power Systems**, v. 23, n. 2, p. 712–718, 2008. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4494595>.

FERREIRA, L. R. **Algoritmo de Autorrecuperação com Corte Seletivo de Carga para Redes Elétricas Inteligentes**. 2015. 143 f. Dissertação (Mestrado em Engenharia Elétrica), Universidade Federal do Paraná, Curitiba (PR), 2015.

FERREIRA, L. R.; SIEBERT, L. C.; AOKI, A. R. Load Shedding Through Optimal Power Flow to Support Self-Healing Actions in Distribution Feeders. 2014 IEEE PES Transmission & Distribution Conference and Exposition - Latin America (PES T&D-LA). **Anais...** Medellin, Colombia. p.1–6, 2014. IEEE.

FERREIRA, L. R.; SIEBERT, L. C.; AOKI, A. R.; FERNANDES, T. S. P. Load Shedding Through Optimal Power Flow to Support Self-Healing Actions in Distribution Feeders. 2014 IEEE PES Transmission & Distribution Conference & Exposition - Latin America (PES T&D-LA). **Anais...** Medellin, Colombia, 2014.

FLEMING, S. W. A Machine Learning Metasystem for Robust Probabilistic Nonlinear Regression-Based Forecasting of Seasonal Water Availability in the US West. **IEEE Access**, v. 7, p. 119943–119964, 2019.

GHAHRAMANI, Z. **Unsupervised Learning**. Berlin: Springer. p.72–112, 2003.

GHALEH, A. P.; SANAYE-PASAND, M.; SAFFARIAN, A. Power system stability enhancement using a new combinational load-shedding algorithm. **IET Generation, Transmission & Distribution**, v. 5, n. 5, p. 551, 2011. Disponível em: <http://digital-library.theiet.org/content/journals/10.1049/iet-gtd.2010.0626>. Acesso em: 1/12/2014.

GHORBANI, J.; CHOUDHRY, M. A.; FELIACHI, A. A MAS learning framework for power distribution system restoration. 2014 IEEE PES T&D Conference and Exposition. **Anais...** Chicago, IL, USA. p.1–6, 2014. Disponível em: <http://ieeexplore.ieee.org/abstract/document/6863310/>.

GHORBANI, M. J.; CHOUDHRY, M. A. A multiagent design for power distribution systems automation. **IEEE Transactions on Smart Grid**, v. 7, n. 1, p. 329–339, 2016. Disponível em: <http://ieeexplore.ieee.org/abstract/document/7173055/>.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A Fast Learning Algorithm for Deep Belief Nets. **Neural computation**, v. 18, n. 7, p. 1527–1554, 2006.

HODGE, V. J.; AUSTIN, J. A Survey of Outlier Detection Methodologies. **Artificial Intelligence Review**, v. 22, n. 2, p. 85–126, 2004.

HOSSEINNEZHAD, V.; RAFIEE, M.; AHMADIAN, M.; SIANO, P. A comprehensive framework for optimal day-ahead operational planning of self-healing smart distribution systems. **International Journal of Electrical Power and Energy Systems**, v. 99, p. 28–44, 2018.

HOWARD, R. A. **Dynamic programming and markov processes**. Cambridge, MA: MIT press and J. Wiley, 1960.

SIEMENS INDUSTRY INC. **Spectrum Power ADMS**. United States: Alpharetta, 2019. Disponível em: < https://assets.new.siemens.com/siemens/assets/api/uuid:0b3805f8-05d0-4fa1-b753-f7b1113c9350/emdg-b90035-00-4ausbrochure-spectrum-power-adms-72.pdf>.

INTERNATIONAL ENERGY AGENCY. **Technology Roadmap - Smart Grid**, 2011.

KAELBLING, L. P. **Learning in Embedded System**, 1990. Stanford University.

KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement Learning: A Survey. **Journal of Artificial Intelligence**, v. 4, p. 237–285, 1996.

KHAN, S. G.; HERRMANN, G.; LEWIS, F. L.; PIPE, T.; MELHUISH, C. Reinforcement learning and optimal adaptive control: An overview and implementation examples. **Annual Reviews in Control**, v. 36, n. 1, p. 42–59, 2012. Elsevier. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1367578812000053>.

KOCSIS, L.; SZEPESV, C. Bandit Based Monte-Carlo Planning. **European conference on machine learning**. p.282–293, 2006. Berlin.

KOST’ÁLOVÁ, A.; CARVALHO, P. M. S. Towards self-healing in distribution networks operation: Bipartite graph modelling for automated switching. **Electric power systems research**, v. 81, n. 1, p. 51–56, 2011.

KOUPPAS, C.; KING, M.; MENG, Q.; MAJOE, D. Machine Learning Comparison for Step Decision Making of a Bipedal Robot. 3rd International Conference on Control and Robotics Engineering (ICCRE). **Anais...** Nagoya, Japan. p.21–25, 2018. IEEE.

LEITE, J. B.; MANTOVANI, J. R. S. Development of a Self-Healing Strategy with Multiagent Systems for Distribution Networks. **IEEE Transactions on Smart Grid**, v. 8, n. 5, p. 2198–2206, 2017.

LI, J.; LIU, C.; SCHNEIDER, K. P. Controlled Partitioning of a Power Network Considering Real and Reactive Power Balance. **IEEE Transactions on Smart Grid**, v. 1, n. 3, p. 261–269, 2010.

LI, J.; MA, X.; LIU, C.; SCHNEIDER, K. P. Distribution System Restoration with Microgrids Using Spanning Tree Search. **IEEE Transactions on Power Systems**, v. 29, n. 6, p. 3021–3029, 2014.

LI, L.; LIN, Y.; ZHENG, N.; WANG, F. Parallel Learning: A Perspective and a Framework. **IEEE/CAA Journal of Automatica Sinica**, v. 4, n. 3, p. 389–395, 2017.

LIN, C.-H.; CHEN, C.-S.; KU, T.-T.; TSAI, C.-T.; HO, C.-Y. A multiagent-based distribution automation system for service restoration of fault contingencies. **European Transactions on Electrical Power**, p. 239–253, 2011.

LIU, C.; LEE, S. J.; VENKATA, S. S. AN EXPERT SYSTEM OPERATIONAL AID FOR RESTORATION AND LOSS REDUCTION OF DISTRIBUTION SYSTEMS. **IEEE Transactions on Power Systems**, v. 3, n. 2, p. 619–626, 1988.

LIU, F.; GUO, J.; ZHANG, X.; HOU, Y.; MEI, S. Mitigating the Risk of Cascading Blackouts: A Data Inference Based Maintenance Method. **IEEE Access**, v. 6, p. 39197–39207, 2018. IEEE.

LIU, H.; CHEN, X.; YU, K.; HOU, Y. The Control and Analysis of Self-Healing Urban Power Grid. **IEEE Transactions on Smart Grid**, v. 3, n. 3, p. 1119–1129, 2012. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6248197>.

LOPES, J. A. P.; MOREIRA, C. L.; MADUREIRA, A. G. Defining Control Strategies for MicroGrids Islanded Operation. **IEEE Transactions on Power Systems**, v. 21, n. 2, p. 916–924, 2006.

MA, Z.; LIU, F.; SHEN, C.; WANG, Z.; MEI, S. Fast Searching Strategy for Critical Cascading Paths Toward Blackouts. **IEEE Access**, v. 6, p. 36874–36886, 2018. IEEE.

MAINI, V.; SABRI, S. Machine Learning for Humans. Disponível em: <https://medium.com/machine-learning-for-humans/why-machine-learning-matters-6164faf1df12>. Acesso em: 12/9/2017.

METROPOLIS, N.; ULAM, S. The Monte Carlo Method. **Journal of American Statistical Association**, v. 44, n. 247, p. 335–341, 1949. Disponível em: <http://www.jstor.org/stable/2280232%5Cr>.

MISHRA, M.; ROUT, P. K. Detection and classification of micro-grid faults based on HHT and machine learning techniques. **IET Generation**, Transmission & Distribution, v. 12, n. 2, p. 388–397, 2017.

MITCHELL, T. M. **Machine learning**. McGraw-Hill Science/Engineering/Math, 1997.

MOMOH, J. A. Smart Grid Design for Efficient and Flexible Power Networks Operation and Control. IEEE/PES Power Systems Conference and Exposition. **Anais...** Seattle, WA, USA. p.1–8, 2009. IEEE.

MOMOH, J. A. **Smart Grid - Fundamentals of Design and Analysis**. 2012.

MORI, H.; MUROI, T. Application of Probabilistic Tabu Search to Distribution System Service Restoration. 2011 IEEE International Symposium of Circuits and Systems (ISCAS). **Anais...** Rio de Janeiro, Brazil. p.1037–1040, 2011. IEEE.

MORTAJI, H.; OW, S. H.; MOGHAVVEMI, M.; ALMURIB, H. A. F. Load Shedding and Smart-Direct Load Control Using Internet of Things in Smart Grid Demand Response Management. **IEEE Transaction on Industry Applications**, v. 53, n. 6, p. 5155–5163, 2017. Disponível em: <http://ieeexplore.ieee.org/abstract/document/7731836/>.

MUHAMMAD, I.; YAN, Z. Supervised Machine Learning Approaches: A Survey. **ICTACT Journal on Soft Computing**, v. 5, n. 3, p. 946–952, 2015.

NANDI, G. C. Deep Learning Based Intelligent Robot Grasping Strategy. IEEE 14th International Conference on Control and Automation (ICCA). **Anais...** Anchorage, AK, USA. p.1064–1069, 2018. IEEE.

OLGUN, M. C. Autonomous Vehicle Control for Lane and Vehicle Tracking by Using Deep Learning via Vision. 2018 6th International Conference on Control Engineering & Information Technology (CEIT). **Anais...** Istanbul, Turkey. p.1–7, 2018. Instanbul: IEEE.

OUALMAKRAN, Y.; MELENDEZ, J.; HERRAIZ, S. Self-Healing for smart grids: Problem formulation and considerations. 3rd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe). **Anais...** Berlin, Germany. p.1,6,14-17, 2012.

PAL, S.; BOSE, S.; DAS, S.; SCOGLIO, C.; Shipboard power system reconfiguration using reinforcement learning. North American Power Symposium 2010, **Anais…** Arlington, TX, USA. 2010. Disponível em: <http://ieeexplore.ieee.org/abstract/document/5618962/>.

PÉREZ-GUERRERO, R.; HEYDT, G. T.; JACK, N. J.; KEEL, B. K.; CASTELHANO, A. R. C. Optimal Restoration of Distribution Systems Using Dynamic Programming. **IEEE Transactions on Power Delivery**, v. 23, n. 3, p. 1589–1596, 2008.

PILO, F.; PISANO, G.; SOMA, G. G. Advanced DMS to Manage Active Distribution Networks. 2009 IEEE Bucharest PowerTech Conference. **Anais...** Bucharest, Romania. p.1–8, 2009. Bucharest: IEEE.

Projetos Piloto no Brasil. Disponível em: <http://redesinteligentesbrasil.org.br/projetos-piloto-brasil.html>. Acesso em: 12/1/2015.

RESENDE, F. O.; GIL, N. J.; LOPES, J. A. P. Service restoration on distribution systems using Multi-MicroGrids. **EUROPEAN TRANSACTIONS ON ELECTRICAL POWER**, v. 21, n. 2, p. 1327–1342, 2011.

RIAHINIA, S.; ABBASPOUR, A.; MOEINI-AGHTAIE, M.; KHALILI, S. Load service restoration in active distribution network based on stochastic approach. **IET Generation, Transmission & Distribution**, v. 12, n. 12, p. 3028–3036, 2018.

RIBEIRO, R.; ENEMBRECK, F.; GUISI, D. M.; et al. An Advanced Software Tool to Simulate Service Restoration Problems: a case study on Power Distribution System. **Procedia Computer Science**, v. 108, p. 675–684, 2017. Elsevier B.V. Disponível em: <http://dx.doi.org/10.1016/j.procs.2017.05.248>.

RONCANCIO, H.; HERNANDES, C.; BECKER, M. Ceiling Analysis of Pedestrian Recognition Pipeline for an Autonomous Car Application. 2013 IEEE Workshop on Robot Vision (WORV). **Anais...** Clearwater Beach, FL, USA. p.215–220, 2013.

RUMMERY, G. A.; NIRANJAN, M. **On-Line Q-Learning Using Connectionist Systems**, 1994. Cambridge, UK: University of Cambridge, Department of Engineering.

SALLES, D.; PINTO, A. C.; FREITAS, W. Integrated volt/Var control in modern distribution power systems based on support vector machines. **European Transactions on Electrical Power**, n. March 2016, p. 2216–2229, 2016.

SAMPAIO, R. F.; MELO, L. S.; LEÃO, R. P. S.; BARROSO, G. C.; BEZERRA, J. R. Automatic restoration system for power distribution networks based on multi-agent systems. **IET Generation, Transmission & Distribution**, v. 11, n. 2, p. 475–484, 2016.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding Machine Learning**. New York: Cambridge university press, 2014.

SIEBERT, L. C.; FERREIRA, L. R.; EDUARDO, K.; et al. Centralized and Decentralized Approaches to Demand Response Using Smart Plugs. IEEE PES T&D Conference and Exposition. **Anais...** Chicago, IL, USA. p.1–5, 2014.

SILVER, D.; LEVER, G.; HEESS, N.; et al. Deterministic Policy Gradient Algorithms. Proceedings of the 31 st International Conference on Machine Learning. **Anais...** Beijing, 2014.

SOLANKI, J. M.; KHUSHALANI, S.; SCHULZ, N. N. A Multi-Agent Solution to Distribution Systems Restoration. **IEEE Transactions on Power Systems**, v. 22, n. 3, p. 1026–1034, 2007.

SUDHAKAR, T. D.; SRINIVAS, K. N. Restoration of power network – a bibliographic survey. **European Transactions on Electrical Power**, n. 21, p. 1159–1177, 2011.

SULLIVAN, M. J.; VARDELL, T.; JOHNSON, M. Power interruption costs to industrial and commercial consumers of electricity. **IEEE Transactions on Industry Applications**, v. 33, n. 6, p. 1448–1458, 1997.

SUTTON, R. S. Learning to Predict by the Method of Temporal Differences. **Machine Learning**, v. 3, n. 1, p. 9–44, 1988. Disponível em: <citeseer.ist.psu.edu/sutton88learning.html>.

SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: an introduction**. 1998.

TAITLER, A.; SHIMKIN, N. Learning Control for Air Hockey Striking using Deep Reinforcement Learning. International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO). **Anais...** Prague, Czech Republic. p.22–27, 2017.

TANG, J.; LIU, J.; PONCI, F.; MONTI, A. Adaptive Load Shedding Based on Combined Frequency and Voltage Stability Assessment Using Synchrophasor Measurements. **IEEE Transactions on Power Systems**, v. 28, n. 2, p. 2035–2047, 2013.

TANG, L.; YANG, F.; MA, J. A survey on distribution system feeder reconfiguration: Objectives and solutions. 2014 IEEE Innovative Smart Grid Technologies – ASIA (ISGT ASIA). **Anais…** Kuala Lumpur, Malaysia, 2014. Disponível em: <http://ieeexplore.ieee.org/abstract/document/6873765/>.

THAMMACHANTUEK, I.; KOSOLSOMBAT, S.; KETCHAM, M. Support Vector Machines for Road Images Recognition in Autonomous Car. 2018 18th International Symposium on Communications and Information Technologies (ISCIT). **Anais...** Bangkok, Thailand. p.291–293, 2018.

THRUN, S. Monte Carlo POMDPs. **Advances in neural information processing systems**. p.1064–1070, 2000.

THRUN, S. B. **The Role of Exploration in Learning Control**. 1992.

TORRES, B. S.; FERREIRA, L. R.; AOKI, A. R. Distributed Intelligent System for Self-Healing in Smart Grids. **IEEE Transactions on Power Delivery**, v. 33, n. 5, p. 2394–2403, 2018.

TOUNE, S.; FUDO, H.; GENJI, T.; FUKUYAMA, Y. Comparative Study of Modern Heuristic Algorithms to Service Restoration in Distribution Systems. **IEEE Transactions on Power Delivery**, v. 17, n. 1, p. 173–181, 2002.

VENAYAGAMOORTHY, G. K. Potentials and promises of computational intelligence for smart grids. **IEEE Power and Energy Society General Meeting**, 2009.

WACKER, G.; BILLINTON, R. Customer Cost of Electric Service Interruptions. **Proceedings of the IEEE**, v. 77, n. 6, p. 919–930, 1989.

WANG, Z.; CHEN, B.; WANG, J.; Networked microgrids for self-healing power systems. **IEEE Transactions on Smart Grid**, 2016. Disponível em: <http://ieeexplore.ieee.org/abstract/document/7127033/>.

WANG, Z.; WANG, J. Self-healing resilient distribution systems based on sectionalization into microgrids. **IEEE Transactions on Power Systems**, 2015. Disponível em: <http://ieeexplore.ieee.org/abstract/document/7017458/>.

WANG, Z.; WANG, J. Service restoration based on AMI and networked MGs under extreme weather events. **IET Generation, Transmission & Distribution**, v. 11, n. 2, p. 401–408, 2017.

WATKINS, C. J. C. H. **Learning from delayed rewards**, 1989. King's College. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/092188909500026C>.

WATKINS, C. J. C. H.; DAYAN, P. Technical Note: Q-Learning. **Machine Learning**, v. 8, n. 3, p. 279–292, 1992.

WERBOS, P. **Advanced Forecasting Methods for Global Crisis Warning and Models of Intelligence**. General System Yearbook, 1977.

WERBOS, P. J. **Neurocontrol and supervised learning**: An overview and evaluation. Handbook of intelligent control, v. 65, p. 89, 1992.

XU, Q.; DING, Y.; YAN, Q.; ZHENG, A.; DU, P. Day-Ahead Load Peak Shedding/Shifting Scheme Based on Potential Load Values Utilization: Theory and Practice of Policy-Driven Demand Response in China. **IEEE Access**, v. 5, p. 22892–22901, 2017.

YAMASHITA, K.; JOO, S.-K.; LI, J.; ZHANG, P.; LIU, C.-C. Analysis, control, and economic impact assessment of major blackout events. **European Transactions on Electrical Power**, p. 854–871, 2008.

YE, D.; ZHANG, M.; SUTANTO, D. A hybrid multiagent framework with Q-learning for power grid systems restoration. **IEEE Transactions on Power System**, v. 26, n. 4, p. 2434–2441, 2011. Disponível em: <http://ieeexplore.ieee.org/abstract/document/5871714/>.

ZHOU, M.; LI, S.; XU, YING; et al. A Distributed Dynamic Programming-Based Solution for Load Management in Smart Grids. **IEEE Systems Journal**, v. 12, n. 1, p. 402–413, 2016. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7435234/>.

ZHOU, Z.; LI, M. Semi-Supervised Learning by Disagreement. **Knowledge and Information Systems**, v. 24, n. 3, p. 415–439, 2010.

ZHU, X. J. **Semi-supervised learning literature survey**, 2005. University of Wisconsin-Madison Department of Computer Sciences.

ZIDAN, A.; EL-SAADANY, E. F. A Cooperative Multiagent Framework for Self-Healing Mechanisms in Distribution Systems. **IEEE Transactions on Smart Grid**, v. 3, n. 3, p. 1525–1539, 2012.

ZIDAN, A.; KHAIRALLA, M.; ABDRABOU, A. M.; et al. Fault Detection, Isolation, and Service Restoration in Distribution Systems: State-of-the-Art and Future Trends. **IEEE Transactions on Smart Grid**, v. 8, n. 5, p. 2170–2185, 2017.

APPENDIX 1 – THESIS PSEUDOCODE

Pseudocode #5 ADAPTED RL TO SOLVE SH+LS PROBLEM

Import Data

$SWNC(swt)$ normally closed switches
$SWNO(swt)$ normally open switches
$SW(swt) \leftarrow [SWNC \quad SWNO]$
$L(s)$ total load for normal topology (state)
$EPR(eq)$ restriction parameter for each equipment eq

Trigger Selection (Outage/Overload)

IF $outage$ **THEN**
   $swFault$ switch that received the fault
   $swAdjacent \leftarrow find\_adjacent(swFault)$ find adjacent downstream
   $SW \leftarrow remove\_switches(swFault, swAdjacent)$ isolation part
ENDIF

Adapted Reinforcement Learning

Initialize:

s (state) based after the isolation part
$ShedVector \leftarrow [5\ 10\ 15\ 20]$
$\alpha \in [0,1]$
$\gamma \in [0,1]$
$iteration \in Z_+^*$
$A(a) \leftarrow [ShedVector\ SW_{toOpen}\ SW_{toClose}]$ list of actions (a)
$Q(s,a)$ according to the problem, zero for this thesis
$continue \leftarrow true$
$policyStable \leftarrow false$

WHILE $continue$
   Choose a (action) from s (state) using the ε-greed policy
   Take an action a
   Observe s'
   Calculate network constraints (parallelism, overload, over and under voltage) based on OPF function, $EPR(eq)$ and $L(s)$

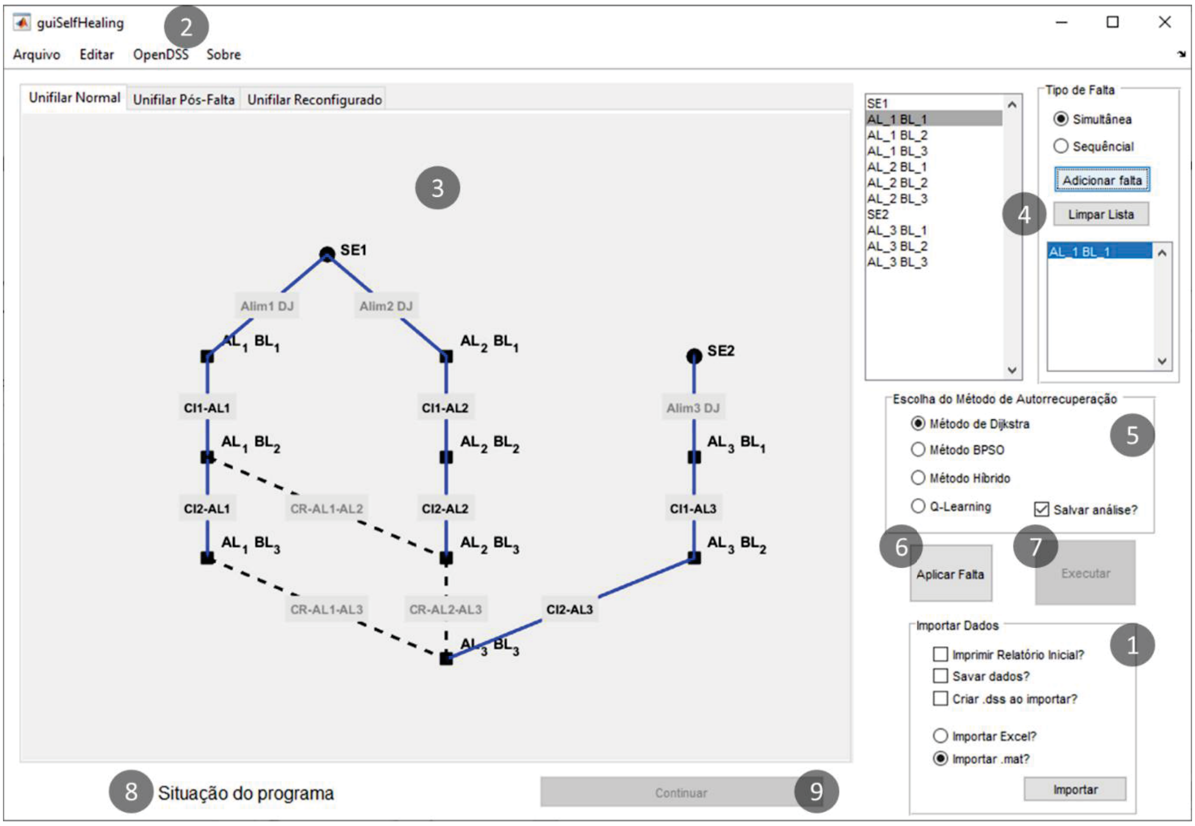
IF constraints exceeded **THEN**
$R(s,a) \leftarrow -1$
ELSE
$R(s,a) \leftarrow \Delta L = L(s') - L(s)$
ENDIF
$Q(s,a) \coloneqq (1-\alpha)Q(s,a) + \alpha(R(s,a) + \gamma max_{a'} Q(s',a'))$
$s \leftarrow s'$
SAVE best topology based on the best system load
IF s selected more than twice **THEN**
$policyStable \leftarrow true$
ENDIF
IF i greater than $iteration$ **OR** $policyStable$ **THEN**
EXIT
ENDIF
ENDWHILE

SOURCE: The author (2020).

APPENDIX 2 – USER GUIDE

The main display is shown in FIGURE 32. Where all control is stablished in this display, to create the database, selected the fault and the type (simultaneous or sequential) and selected the type of technique to resolve the SH problem.

FIGURE 32 – Main Display.



SOURCE: The author (2020).

Legend:

1) Import distribution network data from excel files and create the database for the Matlab program. In case the database was created, it possible just to reimport the database (.mat) without any process.
2) Generate the DSS files for OPENDSS program.
3) Area to plot the graphics for each topology result (normal, pos-isolation and pos-reconfiguration).

4) List of possible faults to simulate. Moreover, can select the fault type, sequential (will process one fault per time) or simultaneous (will process all faults selected at the same time).
5) List of methods to solve the problem.
    a. Dijkstra Method (FERREIRA, 2015);
    b. BPSO Method (FERREIRA, 2015);
    c. Hybrid Method (distributed method) (TORRES et al., 2018);
    d. Q-Learning Method.
6) Button to stat the fault analysis and create the pos-isolation scenario.
7) Button to execute the method to solve the problem.
8) Text to indicate the program process.
9) Button to be used in sequential faults, when the first fault was processed and should continue for the next one.