University of West Bohemia
Faculty of Applied Sciences
Department of Cybernetics

MASTER'S THESIS

Optimization of the power network transfer capability for business purposes

Supervisor: Ing. Martin Střelec, Ph.D. Pilsen, 2020
Bc. Jiří Louda - A17N0016P

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Jiří LOUDA**
Osobní číslo:	**A17N0016P**
Studijní program:	**N3918 Aplikované vědy a informatika**
Studijní obor:	**Kybernetika a řídicí technika**
Téma práce:	**Optimalizace přenosové kapacity elektrické sítě pro obchodní účely**
Zadávající katedra:	**Katedra kybernetiky**

Zásady pro vypracování

1. Seznamte se s úlohou optimalizace přenosových kapacit elektrických sítí.
2. Seznamte se se stávajícími přístupy a nástroji pro optimalizaci přenosových kapacit elektrických sítí.
3. Analyzujte vliv rozdílných citlivostních faktorů využívaných ve standardních řešeních maximalizace přenosových kapacit.
4. Navrhněte vhodnou metodu celočíselné optimalizace lokalizovaného nastavení transformátorů s příčnou regulací fáze.
5. Navrhněte vhodnou metodu pro optimalizaci topologie sítě s ohledem na maximalizaci přenosové kapacity.

Rozsah diplomové práce: **40-50**
Rozsah grafických prací: **dle potřeby**
Forma zpracování diplomové práce: **tištěná**

Seznam doporučené literatury:

1. Jody Verboomen (2008): *Optimisation of Transmission Systems by use of Phase Shifting Transformers.* PhD Thesis, TU Delft.
2. Yung-Chung Chang, Wei-Tzen Yang and Chung-Chang Liu (1993): Improvements on the line outage distribution factor for power system security analysis. *Electric Power Systems Research*, 26.
3. Energinet-DK, Svenska Kraftnät, Fingrid, Statnett (2015): *Methodology and concepts for the Nordic Flow-Based Market Coupling Approach.*
4. https://www.fingrid.fi/globalassets/dokumentit/fi/tiedotteet/sahkomarkkinat/2015/methodology-and-concepts-for-the-nordic-flow-based-market-coupling-approach.pdf
5. Jizhong Zhu (2009): *Optimization of Power System Operation.* J. Wiley & Sons, INC.

Vedoucí diplomové práce: **Ing. Martin Střelec, Ph.D.**
Výzkumný program 1

Datum zadání diplomové práce: **1. října 2019**
Termín odevzdání diplomové práce: **25. května 2020**

Radová

Doc. Dr. Ing. Vlasta Radová
děkanka

Západočeská univerzita v Plzni * Fakulta aplikovaných věd *
L.S.

Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

V Plzni dne 1. října 2019

PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne *22.8.2020* ....

vlastnoruční podpis

Acknowledgment

“I wish to express my sincere extreme appreciation to my supervisor, Ing. Martin Střelec, Ph.D, who convincingly guided and encouraged me to do the right thing even when the road got tough. Without his persistent help and consultations, the goal of this project would not have been realized. I wish to acknowledge the support and great love of my family, my mother, Evženie; my father, Jiří; and my brother Jan. They kept me going on and this work would not have been possible without their input. I would also like to thank my amazing girlfriend Eliška for all her love and support. I would also like to take this opportunity to thank my great friend Ing. Přemysl Voráč, Ph.D for valuable advices.”

Abstrakt

Cílem diplomové práce je optimalizace přenosové kapacity vedení. V teoretické části této diplomové práce je popsáno propojení trhu v Evropě, přístup založený na ATC a Flow-based metodě, optimalizace pomocí změny topologie a phase-shift transformátorů, metody optimálního toku v síti, evoluční algoritmy a analýza citlivosti distribučních faktorů. Praktická část se zabývá implementací konvenčních a evolučních algortimů v programovacím jazyku MATLAB R2018a, kde je využit open-source soubor MATPOWER. Genetické algoritmy za použití roulette wheel selekce a Particle Swarm optimalizace byly zvoleny a následně aplikovány k optimalizaci přenosové kapacity vedení. V závěru kapitol je uvedeno srovnání výsledků poskytnutých těmito algoritmy.

Klíčová slova

elektrické sítě, přenos elektrické energie, optimalizace topologií, optimalizace phase-shift transformátorem, Particle Swarm optimalizace, Genetické algoritmy, MATPOWER, Line outage distribution factors (LODF), Remaining available margin

Abstract

The diploma's thesis focuses on the optimization of the power transfer capacity. The theoretical part of this diploma's thesis describes Market coupling in Europe, ATC and Flow-based approach, Topology and Power Shifting Transformers remedial actions, Optimal Power Flow methods, Evolutionary algorithms and sensitivity analysis of distribution factors. In the practical part, the evolutionary and conventional algorithms were implemented, where MATLAB R2018a was selected as an effective programming language containing the open-source package called MATPOWER. The most promising evolutionary algorithms Particle Swarm Optimization and Genetic Algorithms - roulette wheel were utilized. The analysis and performance assessment was performed and included into the end of every remedial action chapter.

Keywords

power network, power transmission, Topology optimization, Phase Shifting Transformer (PST) optimization, Particle Swarm Optimization, Genetic algorithm, MATPOWER, Line outage distribution factors (LODF), Remaining available margin

Contents

1 Introduction **8**

1.1 Motivation 8

1.2 Market coupling 9

1.2.1 Price determination between two zones 10

1.2.2 Available Transfer Capacity (ATC) 12

1.2.3 Flow-based method (FB) 13

1.2.4 ATC and FB comparison 15

1.3 Problem formulation 16

1.3.1 Overarching goal 16

1.4 Thesis structure 19

2 State of the art in the RAM optimization **20**

2.1 Optimal Power Flow 20

2.1.1 Topology optimization 21

2.1.2 Phase Shifting Transformers optimization 23

2.1.3 Selected artificial intelligence methods 24

3 Topology optimization **26**

3.1 Topology problem statement 27

3.2 Analysis of currently used methods 27

3.2.1 Breadth First Search (BFS) 27

3.2.2 Depth First Search (DFS) 28

3.3 Application of evolutionary algorithms 30

3.3.1 Theoretical framework 30

3.3.2 Genetic algorithms - Roulette wheel selection (GA - RW) . . . 31

3.3.3 Particle swarm optimization . . . 35
3.4 Validation & testing . . . 37
3.4.1 Design of experiments . . . 37
3.4.2 DFS . . . 39
3.4.3 BFS . . . 41
3.4.4 Binary particle swarm optimization . . . 42
3.4.5 Genetic algorithms - Roulette wheel selection (GA - RW) . . . 45
3.5 Results assessment . . . 48

4 Phase Shifting Transformers optimization 51
4.1 PST problem statement . . . 51
4.1.1 Power flow control using PSTs . . . 51
4.2 Validation & testing . . . 53
4.2.1 Design of experiments . . . 53
4.2.2 DFS . . . 54
4.2.3 PSO . . . 55
4.2.4 Genetic algorithms - Roulette wheel selection (GA - RW) . . . 58
4.2.5 Results assessment . . . 61

5 Sensitivity analysis of distribution factors 64
5.1 Overview of power system security assessment . . . 64
5.1.1 Line Outage Distribution Factor . . . 64
5.2 Methodology . . . 65
5.2.1 Inputs . . . 66
5.2.2 Calculation . . . 66
5.2.3 Outputs . . . 66
5.3 Analysis . . . 67
5.3.1 Design of experiments . . . 67
5.3.2 Error analysis . . . 68

6 Conclusion 72

Chapter 1

Introduction

1.1 Motivation

In the beginning, electricity was transferred only over short distances, because power stations were built near electricity consumption areas. During the electrification era, the power system of each state was primarily developed separately and operated independently to surrounding areas. Non-cooperation between individual states was mainly due to the political situation. As cross-border electricity networks were built only for security reasons, most of the money was mainly invested in the construction of a national electricity network. As a result, the power grid was at a good level within one state and often at a lower level among other states. [39]

The liberalization of the energy sector in Europe brought to the traditionally regulated sector competition and pressure to reduce the price of electricity to end-users. For this purpose, member states were forced to separate electricity generation from transmission and distribution. This caused many shutdowns of national energy monopolies and establishment of state-owned companies. Utility companies acting in separated states could now extend their offer to other countries. This led to an increase of competitiveness not only on the utility side but on the customer side as well by significant price differences between states with integrated energy markets. Electricity trading was, therefore, the same as another classic commodity. The customer could choose the best offer in the whole market. Companies able to provide electricity at lower prices than their competitors had a natural tendency to export their electricity to countries where spot market prices were higher. [19]

Nowadays, the European electricity market is significantly integrated. The electricity can be provided by different stakeholders such as large utility companies, small private businesses or even by individual people (i.e. prosumers) for instance. Due to the power system interconnection, this causes unprecedented power flow across traditional levels of power systems. Every day is a large volume of electricity transported over power systems of EU member states, which has strong requirements on the cross-border transfer capacities of individual power systems.

Therefore the cross border exchanges as well as power system operation (i.e. power flows) needs to be managed in a secure and reliable way.

1.2 Market coupling

Market coupling is one of possible methods for integrating electricity markets in different zones. The main goal of market coupling is an effective utilisation of daily cross-border capacities. The market coupling means that the cross-border flows at the day-ahead stage are determined by using the price signals in the day-ahead spot markets in each zone. This enables an efficient European wide price formation mechanism and optimised use of the transfer power grid through a strong interaction between price zones. When two or more zones make an agreement, the market is then coupled. In this case, the electricity from the market with a lower price is sold to the market with a higher one. This leads to price convergence in both markets. In a case when a transfer capacity of a power network between two different countries is sufficient, then the market is completely coupled. It means the prices of electricity are the same in both countries. [19] The most important advantages are:

- Minimising differences in prices between two or more market areas
- Ensuring convergence of power prices across regions
- Reducing price volatility
- Fair and efficient access to cross-border capacities

The main players involved in the setup of the Market Coupling Solution are:

- **Transmission System Operators (TSOs)**

  Independent entities of the other electricity market players, which are responsible for the mass transfer of electric power on the main high voltage electric networks. TSOs provide access to the electric network for electricity market players. The most important market players are companies which generate electrical power, traders suppliers, distributors and directly connected customers. TSOs must follow the non-discriminatory and transparent rules. In order to ensure the security of supply, they also guarantee the safe operation and maintenance of the system. In many countries, TSOs are in charge of the development of the grid infrastructure too. In the Czech Republic, this service is provided by state-owned company ČEPS [11].

- **Power Exchanges (PXs)**
  Entities providing an organized platform for competitive power market. They allow all market players to buy and sell electricity through an auction mechanism. The power exchanges in EU electricity markets include EPEX Spot (France, Germany, Austria, and Switzerland), APX (the United Kingdom, Netherlands, and Belgium), Nord Pool Spot (Nordic and Baltic region), OMIE (Iberian Peninsula), Omel (Spain), IPEX (Italy), and PXE (Central Europe). [4]

1.2.1 Price determination between two zones

Price of electricity is determined on a market. Generally, the market is a collection of buyers and sellers that determine the price of a product. So in this case, the buyers are consumers of electricity or representatives of such consumers, while sellers are electricity-generating resources, and the product in trade is electricity. The trade with electricity proceeds as follows. Electricity traders submit their supply and demand for a certain amount of electricity on the domestic market. All bids are collected by the electricity market operators. After that, they sort them on the basis of the so-called merit-order. Where the supply curve meets the demand curve, the market-clearing price of domestic electricity is determined. This pricing process is done also in neighbouring markets. [19]

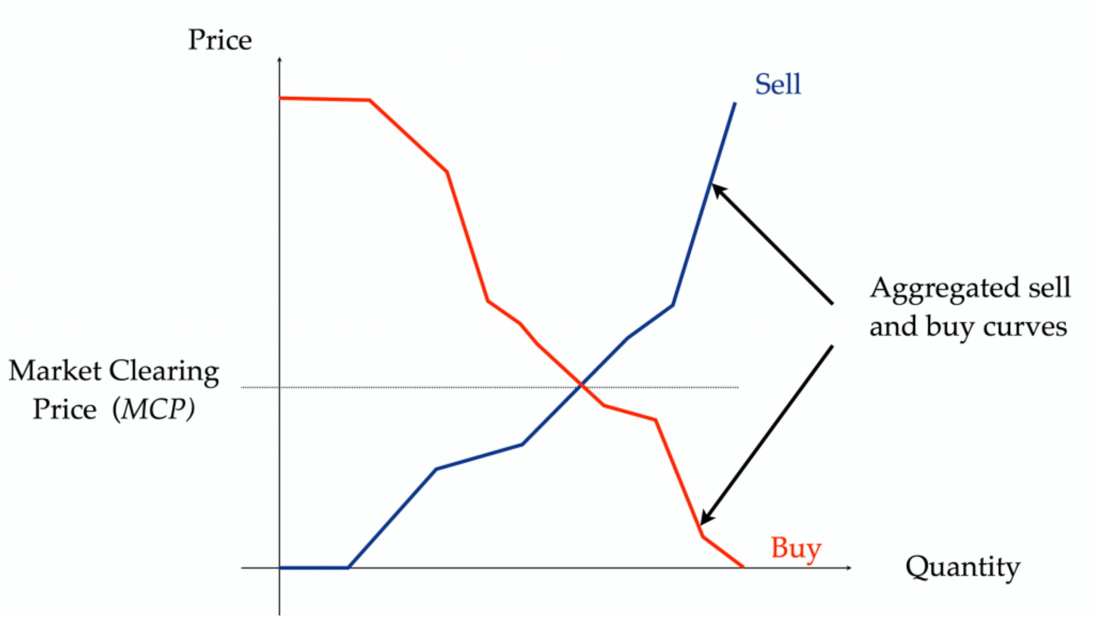


Figure 1.1: Clearing price determination [22]

After these steps the clearing price is determined and compared in 2 different markets. The price will vary depending on the factors. The examples are an energy mix and consumption of a current country. After these steps, the available transfer capacity between states is calculated and cross-border coupling market is triggered. The process of the interconnection of energy markets is shown in the following pictures, where two countries, A and B, have different prices of electricity on the market, which are subsequently interconnected. There are two different cases that may appear during the interconnection [19]:

- **Available transfer capacity(ATC) is large enough**
  The prices on both markets converge. Because the price of market B is higher that the price of market A, the electricity will be exported to market B. The producer, or trader, will export electricity to a neighbouring country as long as the marginal offer price on its market is lower than the marginal demand price on the other market (bid). In this case, where ATC is sufficient, the price is aligned in both markets and the markets are coupled. It is described on the image 1.2.

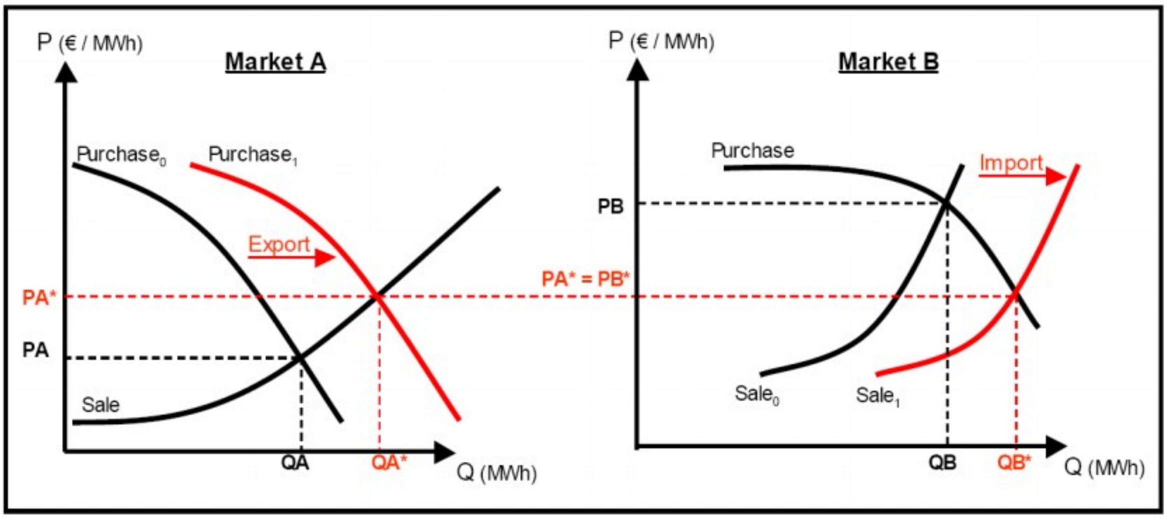


Figure 1.2: Clearing price, ATC is large enough [3]

- **Available transfer capacity(ATC) is not large enough**
  The prices on both markets cannot be fully equalized, so it diverge. It can be seen on image 1.3.

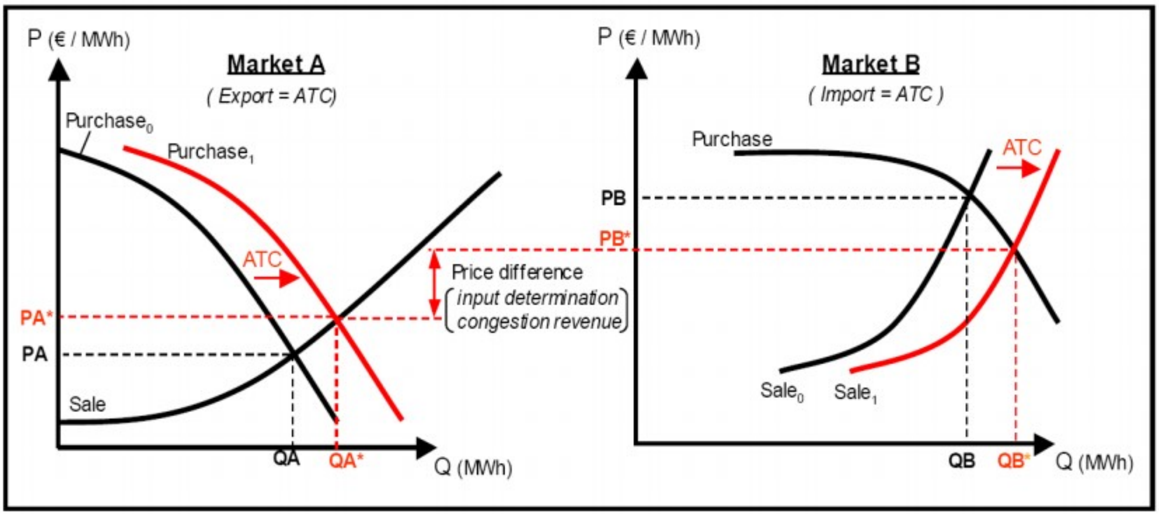


Figure 1.3: Clearing price, ATC is not large enough [3]

These options led to the start of cross-border electricity trade. It has many advantages which include, above all, freedom of trade, so prices are based on supply and demand. On the other hand, this quick start brings a number of problems. Electricity trade faces technical constraints in the form of limited transmission capacities between coupled countries. This problem is caused by the low-level cross-border network that was built in the beginnings and the reason for it is described above. Transmission capacity is significantly different. Among some countries is the transmission capacity almost unlimited - Benelux countries, other countries that do not have sufficient cross-border lines can have a lack of this transmission capacity. Because of economic efficiency and public acceptability of high-voltage power lines, it is not feasible to install sufficient cross-border capacity. Consequently, in many cases the transport capacity demanded by market participants exceeds the available capacity - in this context, it is called cross-border congestion. To prevent these cross-border congestion, the calculation and allocation of available transmission capacity is essential. [3].

- **Explicit vs allocation**
  In explicit allocation, the available cross-border transport capacities and the traded energy are dealt separately from one another. Energy is purchased or sold in a trading venue, for example, an electricity power exchange or OTC (Over the counter market). Cross-border capacity is carried out by auction houses. By trading electricity and capacity separately, one commodity trading may lack information on the price of another commodity, resulting in inefficient use of cross-border capacities. For the explicit allocation are especially typical long-term contracts. They are mostly applied in annual and monthly auctions on borders in Europe. Short-term daily explicit auctions are made as well, although to a lesser extent. Cross-border capacity takes place through the Joint Allocation Office(JAO) that replaced former Central Allocation Office Freising (CAO) and Capacity Allocating Service Company (CASC) [31].

- **Implicit allocation**
  The second type of capacity allocation is implicit allocation. The biggest difference is that the electric power and the cross-border capacity are allocated together. This allocation is mostly used in short-term markets in opposite to explicit allocation.
  There are two different ways used for cross-border capacity calculation in Europe: the Available Transfer Capacity (ATC) and Flow-Based method. These methods are in detail described below.

1.2.2 Available Transfer Capacity (ATC)

The ATC method is widespread in many parts of Europe and it is much simpler than Flow-based method. ATC market coupling takes place on the basis of an implicit auction, where all input data from the daily market are evaluated simultaneously. The ATC of a transfer system is a measure of the transfer capability that remains in the physical transfer network for further commercial activity. This measure depends on a number of factors. The most important is system generation dispatch, system

load level, load distribution in the network, network topology and the limits on the transfer network. These factors create a set of system conditions which specify the power network. The ATC is calculated as a maximum amount of power that a transfer system can transport when power is injected at one place of network and the same amount of power is extracted on another location.
Therefore is the following terminology presented:

TTC	Total Transmission Capacity
TRM	Security margin
NTC	Net Transmission Capacity
ATC$_n$	Available Transmission Capacity in Base Case
PF	Parallel Flow
LF	Loop Flow
ATC	Available Transmission Capacity
AAC	Already Allocated Capacity

The detail calculation is reviewed in the following equations:

$$\begin{aligned} NTC &= TTC - TRM \\ ATC_n &= NTC - PF - LF \\ ATC &= ATC_n - AAC \end{aligned} \tag{1.1}$$

First of all, the NTC is calculated by the difference between TTC and TRM. In the second step, there is made a calculation of ATC_n. This is done by subtraction of PF and LF from NTC. It is because of the N-1 criterion which must be complied in any operational situation. This criterion is very important because it says that any outage of one element of the power grid does not endanger the operation of the whole power grid. Then the ATC value is continuously computed with changes of AAC in yearly, monthly or daily auctions [31].

1.2.3 Flow-based method (FB)

The flow-based (FB) model works essentially on a similar principle as the ATC model, but takes into account more parameters and optimization conditions for a better representation of the actual state of the network. The flow-based method treats power flows as a real flow of energy that respects the Kirchhoff's laws. This approach is, therefore, more accurate then ATC. On the other hand, due to the larger number of parameters describing the power grid, this method is more difficult to calculate the data.

The flow-based model was chosen as the target model across the EU. In contrast to the ATC model, this algorithm should ensure more efficient use of transfer capacities between individual zones. Compared to the ATC model, the flow-based model provides a more realistic view of real flows between neighbouring markets. Unlike the ATC method, which in its calculation takes into account only the restrictions on the transfer line between neighbouring markets.

The Flow-based model is developed from the nodal pricing model, where is are used the power transfer distribution factors (PTDF) to calculate flows. This model imposes an aggregate PTDF matrix on certain areas or lines in order to limit the power exchange between price areas. Therefore, the solutions given by the Flow-based model may still be infeasible in some parts of the network and re-dispatching may be needed. The FBMC model tries to reduce the explicit limitations to cross-border trades, which is an indirect way of dealing with individual line constraints and instead focuses on selected critical branches (CBs) that are the ones most likely to be influenced by cross-border trading.
The most important concepts of Flow-based method are:

- Generation shift key (GSK)
- Critical branch (CB)
- Power transfer distribution factor (PTDF)
- Remaining available margin (RAM)

Generation shift key (GSK)

It is a factor describing the most probable change in net injection at a node, relative to a change in the net position of the zone that it belongs to. The set of GSKs is crucial in the FBMC model. Although the GSKs should be defined before market clearing, in reality, they cannot be known until the FBMC calculation is completed. The TSOs calculate the GSKs using a "base case", anticipating grid topology, net positions, and corresponding power flow for each hour of the day of delivery. In practice, a precise procedure to define the GSKs is missing. Generally, it is an estimate of how a country's total generation is distributed among the generators within the country or area [1].

Critical branch (CB)

Critical branches are transmission lines, cables, or devices that can be significantly affected by cross-border flows. In the beginning, Critical Branches should be chosen from the whole power network. They are expected to be limiting cross-zonal trades. Therefore are these branches monitored during the FB capacity allocation. The determination of CB is based on operational experience of each TSO. The key task is to assess as many CBs as needed, but in ideal case not too much because the more CBs, we have the more complicated it is to be solved. On the other hand, a dismission of CB can lead to invalid results. That is why the significance of CB shall be computed and supervised [1].

Power transfer distribution factor (PTDF)

Power Transfer Distribution Factors (PTDF) is a matrix, where its elements indicate the change of power flows of considered combinations of critical network elements and contingencies. These values provide a linearized approximation of how the flow on the transfer lines is changed. This way, it can be monitored which combinations of cross-zonal exchanges threaten to overload a specific line. There are two types of PTDF. The first one is nodal PTDF and the second one is called zonal PTDF. These matrices are described in the second paragraph [1].

The nodal PTDF matrix is calculated by subsequently varying the injection on each node of the Common Grid Model (CGM). CGM is defined as a Union-wide data set agreed between various TSOs describing the main characteristic of the power system (generation, loads and grid topology) and rules for changing these characteristics during the capacity calculation process. For every single nodal variation, the effect on every critical network element is monitored and calculated as a percentage.

Then the GSK translates the nodal PTDFs into zonal PTDFs as it converts the zonal variation into an increase of generation in specific nodes. The PTDFs characterize the linearization of the model. In the subsequent process steps, every change in net positions is translated into changes of the flows on the critical network element with linear combinations of PTDFs.

Remaining available margin (RAM)

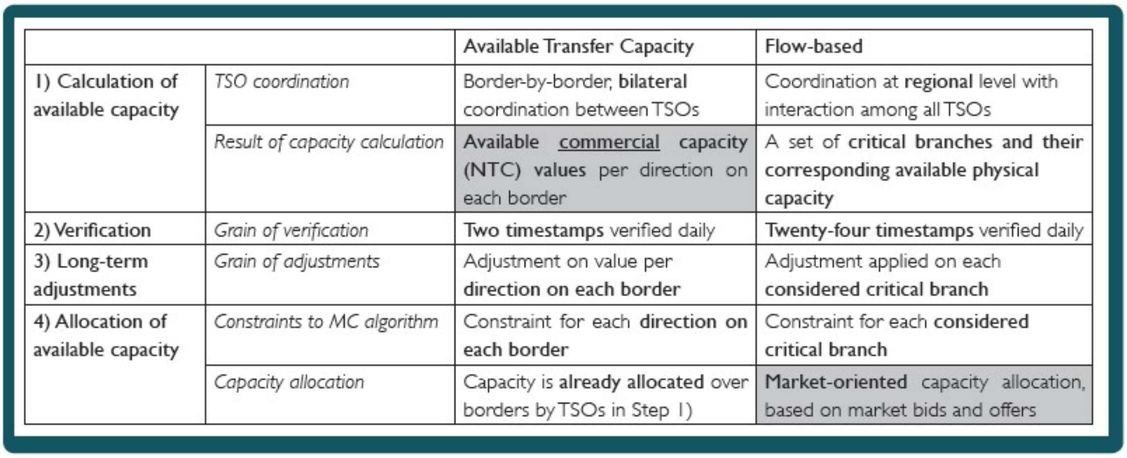
Remaining available margin is the line capacity that can be transferred in a power grid and plays the same role as available transfer capacity in ATC approach. This line capacity is used by the day-ahead market.
There are two main steps of RAM calculation:

- The critical branches and critical outages are determined by TSOs.
- The RAM is calculated for these critical branches that are under critical outages.

The RAM value is crucial for maximization power flow in a power grid. Therefore, the RAM calculation and maximization is described in detail in the following chapter - Problem formulation [1].

1.2.4 ATC and FB comparison

The flow-based method leads to more efficient use of electricity generation and its transfer in a power grid. While under ATC, TSOs themselves determine capacity values based on forecasts and historical data, the FB mechanism allows TSOs only to derive the impact that trade will have in terms of physical flows on the network. More capacity is offered to the market under FB market coupling, resulting in an overall welfare gain and increased price convergence. In a table down below is described the summary of key differences between ATC and FB approach.



		Available Transfer Capacity	Flow-based
1) Calculation of available capacity	*TSO coordination*	Border-by-border, **bilateral** coordination between TSOs	Coordination at **regional** level with interaction among all TSOs
	Result of capacity calculation	**Available** commercial **capacity (NTC) values** per direction on each border	A set of **critical branches and their corresponding available physical capacity**
2) Verification	*Grain of verification*	**Two timestamps** verified daily	**Twenty-four timestamps** verified daily
3) Long-term adjustments	*Grain of adjustments*	Adjustment on value per **direction on each border**	Adjustment applied on each **considered critical branch**
4) Allocation of available capacity	*Constraints to MC algorithm*	Constraint for each **direction** on **each border**	Constraint for each **considered critical branch**
	Capacity allocation	Capacity is **already allocated** over borders by TSOs in Step 1)	**Market-oriented** capacity allocation, based on market bids and offers

Figure 1.4: Summary of key differences between ATC and FB approach [24]

1.3 Problem formulation

1.3.1 Overarching goal

As mentioned in the previous section, two parameters are needed for FBMC: (i) the zonal Power Transfer Distribution Factors (PTDF) and (ii) the Remaining Available Margin (RAM). The FBMC parameter calculation is started two days before the delivery day and finished the morning day-ahead so that they can be used in the day-ahead market clearing.

The RAM express line transfer capacity of power network, which can be used for electricity transfers initiated by realization of market transactions contracted in the day-ahead market for example. The RAM calculation consists of two main steps:

1. The critical branches and critical outages are determined. The result of this process is a set of critical network elements (CNE).
2. The RAM is calculated for all CNE under critical outages.

A transfer line is considered to be significantly impacted if the zonal PTDF for that line is larger than 5%. The critical branches are determined by each TSO for its own power network. For each critical branch, the maximum allowable power flow is determined as the physical (thermal) limit of the transmission element. [37]

The RAM differs from the max capacity of the CNE as in the following equation:

$$RAM(l) = F_{max}(l) - F_{ref}(l) - F_{AV}(l) - F_{RM}(l) \qquad \forall l \tag{1.2}$$

- $F_{max}(l)$ is the maximum allowable power flow on critical branch l in [MW].
- $F_{ref}(l)$ is the reference flow on critical branch l in [MW] caused by commercial transactions outside the day-ahead power exchange. These commercial transactions can be internal (within a market zone) or external (between market zones).
- $F_{AV}(l)$ is the Final Adjustment Value on critical branch l in [MW]. The FAV allows TSOs to take account of knowledge and experience that cannot be introduced in the formal FBMC method, such as an additional margin due to complex remedial actions.
- $F_{RM}(l)$ is the Flow Reliability Margin on critical branch l in [MW]. The FRM is a safety margin that needs to compensate for approximations and simplifications made in the FBMC methodology such as the assumptions inherent to a zonal PTDF, unintentional flow deviations due to load-frequency control, and the use of a linear grid model with a simplified topology.[37]

As mentioned, the RAM is a key parameter to calculate on critical network elements in a power grid. This value describes the line capacity of transferable power flow. For the maximization of transfer capacity of power network and maintaining the power system stability, it is necessary to optimize RAM under security constrains.

Therefore the goal is to create an optimization algorithm that maximizes the RAM on selected critical network elements. Thus, the aim is to maximize minimal RAM of the power network and the objective function is considered as follows:

$$J = \max(\min_{l \in CB}(RAM(l)) \tag{1.3}$$

The RAM is unidirectional variable and if also power flow in opposite direction would be monitored, then another CNE should be added to the optimization model. Thus, RAM for not overloaded CNE takes values from the following interval

$$RAM \in [0; 2(F_{max} - F_{RM} - F_{AV})], \tag{1.4}$$

where the lower limit corresponds to the situation when the CNE is fully loaded, i.e. $F = (F_{max} - RM - AV)$, and the upper limit corresponds to the same loading but with opposite direction of power flow.

Now, let be a RAM of monitored line $l \in CB$ under consideration of outage of any power line $k \in CO$ given as

$$RAM_{ref}^{CBCO}(l,k) \overset{def}{=} F_{max}(l) - F_{RM}(l) - F_{AV}(l) - F_{ref}(l) - LODF(l,k)F_{ref}(k). \tag{1.5}$$

Generally, it is required that the proposed algorithm has the following features:

- None of CBs will be overloaded due to the actions realized by optimization algorithm in nominal state or in case of any contingency CO.

- The solution will not increase loading of already overloaded CBs.
- CO is considered to be an outage of a single CB.

The proposed RAM maximizing algorithm consists of three possible remedial action types adjusting RAM by changing the Phase Shifting Transformers (PST) tap positions, Topological RAs and by Generation Redispatch , respectively, see green blocks of flowchart in figure 1.5. The first two approaches represent non-costly RAs (NC-RAs), which will be preferred over the costly RA. These actions will be used only in a situation if any CB is overloaded and none of non-costly RAs can correct it. On the figure 1.5, simplified workflow for RA optimization algorithm is depicted.

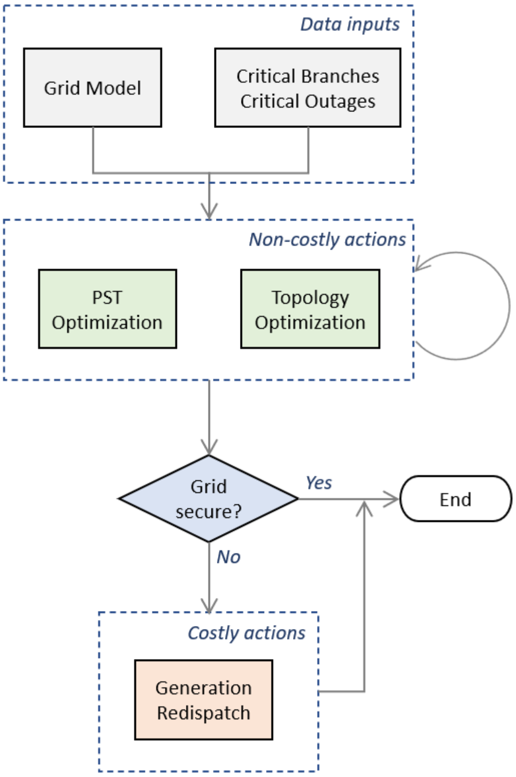


Figure 1.5: Flowchart of Remedial Actions Optimization Algorithm

Phase Shifting Transformers (PST)

Change of tap position of Phase Shifting Transformers (PSTs) is the first non-costly remedial action, which control of power flows in steady state conditions. To use this ability efficiently, it is important to install PST in a suitable place and optimize their operation. The optimal adjustment of PST tap can be considered as an optimal power flow (OPF) problem, which is of a non-linear nature. The solution of this problem means the determination of the optimal settings by adjusting control variables (PSTs' tap position) in a power system.

Topological RAs

Another non-costly remedial actions consist in the change of power network topology in a reasonable way, which change power network transfer capacities and influence power flows. The aim of the topology optimization is to maximize minimal RAM though change of power network topology.

Generation Redispatch

The last action stands for a costly remedial action, which is applied, when non-costly ones are not efficient. This remedial action consists in an adjustment of relevant power generating unit in order to reduce power flows over critical network elements. However, the adjusted generation power causes the power imbalance in the power network, which has to be eliminated through expensive ancillary services.

1.4 Thesis structure

This thesis is divided into 6 chapters including the Introduction and the Conclusion.

In the chapter Introduction, the Market coupling is described. Further, the ATC and Flow-based approaches are considered and compared. This chapter presents the cross - border capacity calculation method as well. At the end, the problem formulation for RAM optimization is introduced.

The chapter 2 focuses on the analysis of the current state of the art in the RAM optimization, where the commonly used methods for Optimal Power Flow were analyzed.

In the 3rd chapter, the Topology optimization was performed. The selected evolutionary algorithms were described at sufficient level of detail. Further the impact analysis of different parameters on the computation performance was presented.

Chapter number 4 is devoted to Power Shifting Transformers optimization, where modified algorithms from Topology were applied and their computation performance were analysed under different algorithm configurations.

In the 5th chapter, the sensitivity analysis of line distribution factors is performed, which are important elements for approximate contingency analysis based on linearized power systems. In this chapter, the accuracy of the approximation were analyzed and the improvement of the accuracy was proposed.

Chapter 2

State of the art in the RAM optimization

As mentioned in the section 1.3, the main goal of the diploma thesis is the RAM optimization using selected remedial actions. The RAM value calculations are performed under Flow based(FB) method representing the currently used methodology in power network systems. The FB is a relatively new approach and therefore there are available only few resources for RAM optimization. On top of that, energy companies (mainly power system operators) dealing with this issue are usually private and their solutions are under commercial secret. For these reasons, it was needed to analyse general methods that are commonly used for the well known energy problems such as the Optimal Power Flow (OPF), which stands for a traditional optimization problem widely solved in the power system community. OPF problem has a long history briefly described below in the following section.

2.1 Optimal Power Flow

Over the past half-century, the Optimal Power Flow (OPF) has gained a great attention due to its importance in power system operation. In general, OPF seeks to optimize the operation of electric power generation, transmission, and distribution networks subject to system constraints and control limits. The most commonly used objective is the minimization of the overall cost function including operation as well as maintenance costs. In addition, several other objectives can be included into optimization such as minimization of the active power loss, nodal voltage security, emission of generating units, number of control actions, and load shedding [45]. In practical power system operation, the OPF problem adjusts the continuous control variables (e.g. power injections and nodal voltages) and discrete control variables (e.g. transformer tap setting, phase shifters, and reactive injections on shunts) to reach the optimal objective function while satisfying a set of physical and operational constraints. [9]

As mentioned and detailed in the section 1.3, three main remedial actions are considered. Each RA works with different state spaces where the required solutions are sought. Based on the state space, only certain methods can be applied effectively. The partitioning of selected approaches linked with variable types is shown in the figure 2.1 further detailed in following sections.

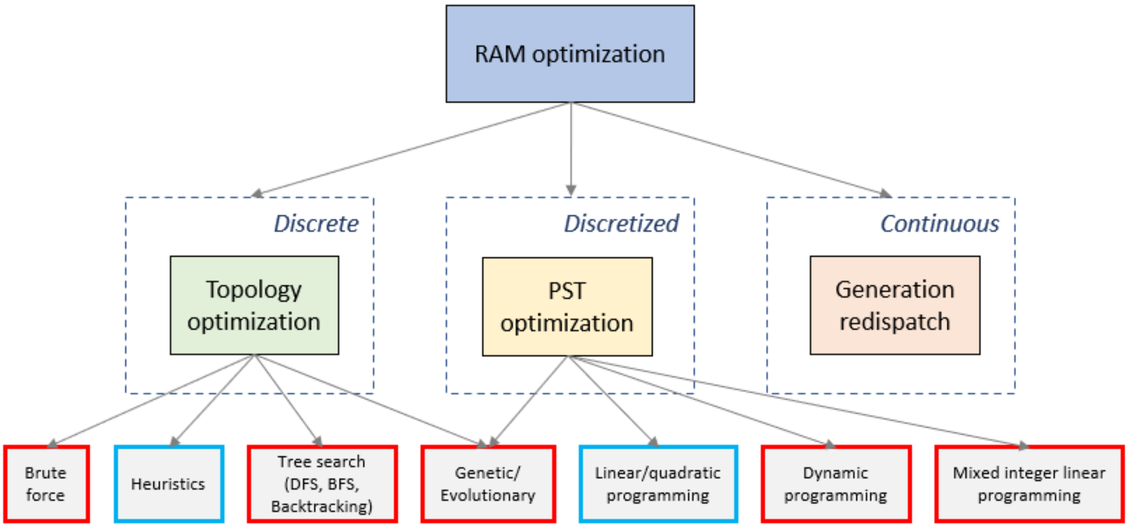


Figure 2.1: The detailed division of RA

2.1.1 Topology optimization

Topology optimization (TO) is a remedial action based on the change of power network topology, where the size of optimized power network is crucial. As shown in the figure 2.1, TO works with the discrete state space. There are different methods than can be used, which especially depends on the size of the state space. In a real case, the number of lines and other elements in a power network is usually huge (more than 15 000 in a merged EU power system). The state space is created by binary vectors representing every single topology change (i.e. power lines operation status). The size of the state space grows exponentially depending on the amount of changeable lines. In this thesis, two main approaches for topology optimization are considered.

- Conventional methods
- Artificial Intelligence methods

Conventional methods

Conventional methods embraces relatively simple methods used for the state space search. They usually achieve very good results, when the state space is small. For these cases, the main advantage is a guarantee of finding the maximum. On the other hand, the major disadvantages are the large demands on memory and computation time. In the case of a large network optimization, these disadvantages make it absolutely impossible to use and it is necessary to find other more suitable algorithms (e.g. Artificial Intelligence methods). The most well-known methods are described in the following itemization.

- Brute force (BF)
  BF is an optimization technique which examines every possible alternative in the solution space to find the best one [44]. Brute force can be used for a small group of issues containing a low number of total possible solutions, while the cases, where the state space is large, this approach is extremely inefficient and often inapplicable due to huge time and memory requirements. Precisely for these reasons, the BF will be used only as a baseline for result validation in our case.

- Heuristic approaches
  Methods using an additional knowledge about the problem being solved, can be, for example, the approximate location of the global maximum. It is significantly better than the brute force technique because the whole state space does not have to be searched.

- Decision tree search
  A data structure representing a whole group of possible solutions. Individual elements (nodes) are arranged so that it is possible to quickly search for a given value in this tree. There are 3 main principles considered:

  – Breadth First Search (BFS)
    BFS algorithm traverses nodes in tree by levels. It starts at the tree root and explores all of the neighbor nodes at the present depth prior move on to the nodes at the next depth level. [43]

  – Depth First Search (DFS)
    An opposite algorithm to BFS prior move on to the nodes at the deeper level.

  – Backtracking
    This is an improvement in the search for brute force solutions in that a large number of potential solutions can be ruled out without direct testing. The algorithm is based on DFS algorithm.

Artificial Intelligence methods

Artificial Intelligence (AI) methods is another type of algorithms primarily applied to solving problems, where conventional optimization methods can not be used. The heuristic search has become a very popular technique in searching the global or near-global optimal solution. One of the subgroups of AI methods are Evolutionary algorithms mostly involving meta-heuristic optimizations [10] such as:

- genetic algorithm (GA)
- particle swarm optimization(PSO)
- evolutionary programming (EP)
- evolution strategy (ES)
- genetic programming (GP)
- learning classifier systems (LCS)

2.1.2 Phase Shifting Transformers optimization

The next step is to analyse methods used for Phase Shifting Transformers (PST) optimization. In the diploma thesis, the PST optimization focuses on the the methods to the problem of optimizing transformer tap ratio settings. This issue contains controllable variables, which create a subset of the state variables. Control variables are of discrete nature, which can be considered as continuous in some optimization approaches (e.g. linear programming). In our case, tap changer positions are considered as discretized, which led to a problem similar to the topological one. The main differences were the size of a state space and optimized variables(Topology - binary, PST - discrete).
Methods for the PST optimization can be divided into two groups as in the case of topology optimization, which area:

- Conventional methods
- Artificial Intelligence(AI) methods

The AI methods used for PST optimization are the same as in topology, where they are described in detail. On the other hand, the Conventional methods for PST are very different caused by continuous variables of a state space, which is often considered as well. The Conventional methods are described in the following section.

Conventional methods

- Linear programming (LP)
  LP allow the use of well-developed LP solution methods, such as the Simplex Method. Such methods are highly desirable for many reasons: efficient handling of inequality constraints, quick recognition of problem infeasibility, speed, reliability, and (especially) excellent convergence properties. In addition, LP formulations are convex, and therefore guarantee a global optimal solution.[9]

- Quadratic programming (QP)
  QP is a special form of nonlinear programming in which the objective function is quadratic of the form $\frac{1}{2}x^TQx + q^T$ and all constraints are linear. QP-based OPF was introduced as alternative to LP for cases where LP formulations perform poorly, such as for loss minimization.[9]

- Dynamic programming
  Dynamic Programming (DP) is an algorithmic technique for solving an optimization problem by breaking it down into simpler subproblems and utilizing the fact that the optimal solution to the overall problem depends upon the optimal solution to its subproblems. DP is often efficiently applied in the problems of energy storage operation.[6]

- Mixed Integer Linear Programming (MILP)
  Continuous LP, QP, and NLP formulations cannot accurately model discrete control elements, such as transformer tap ratios or switched capacitor banks. Discrete variables present one of the most challenging aspects of OPF. MILP is another option for OPF optimization, where the linearization of the system is needed. MILP retains many of the benefits of LP while also accommodating discrete variables. [9]

2.1.3 Selected artificial intelligence methods

For solution of the Topology and Phase Shifting Transformers optimization, two meta-heuristic approaches were selected. Already before the analysis there was a tendency to choose methods using AI especially Evolutionary algorithms (EAs). The analysis confirmed that this approach is very suitable for the RAM optimization, as they can work with a huge state space containing binary(Topology) or discrete(PST) variables and have the ability to find the global maximum. EAs contains many different algorithms where the two most popular ones for optimization in power network systems were selected: Genetic Algorithms(GA) - Roulette Wheel and Particle Swarm Optimization(PSO). For comparison, conventional methods were chosen, namely DFS and BFS. Effective applications of GA and PSO are mentioned from following articles:

Genetic algorithms

Po-Hung Chen [5] presented a large scale economic dispatch problem by GA. He designed new encoding technique where in, the chromosome has only an encoding normalized incremental cost. There is no correlation between total number of bits in the chromosome and number of units. The unique characteristic of Genetic Approach is significant in big and intricate systems which other approaches fails to accomplish.

M. Younes and M. Rahl [28] presented hybrid Genetic Algorithm (combination of GA and Mat power) that was used to solve OPF including active and reactive power dispatches. The method uses GA to get a close to global solution and the package of MATLAB – m files for solving power flow and optimal power flow problem (mat power) to decide the optimal global solution. Mat power is employed to adjust the control variables to attain the global solution. The method was validated on the modified IEEE 57–bus system and the results show that the hybrid approach provides a good solution as compared to GA or Mat power alone.

Walters et al.[41] applied a Genetic Algorithm (GA) to solve an economic dispatch problem for valve point discontinuities.

Particle swarm optimization

T. Saravanan [33] presented the application of PSO technique to solve OPF with inequality constraints on line flow. Algorithm is implemented on a six-bus three-unit system and the results are compared with linear programming method.

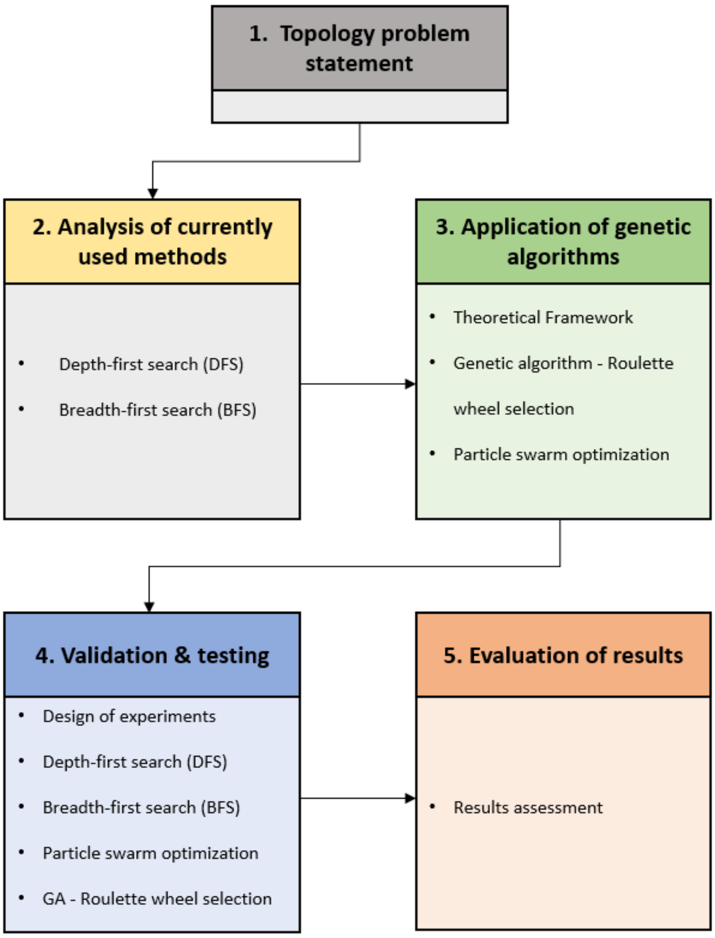
J. Praveen and B. Srinivasa Rao [29] presented optimization problem solved by PSO with power injection model of the FACTS device. The proposed methodology is tested on standard IEEE 30-bus test system and the results are compared for single objective optimization with and without FACTS device.

M.Saravanan et al. [32] proposed the application of Particle Swarm Optimization to find the optimal location, settings, type and number of FACTS devices to minimize their cost of installation and to improve system loadability for single and multi-type FACTS devices. While finding the optimal location, the thermal limit for the lines and voltage limit for the buses are taken as constraints.

Chapter 3

Topology optimization

This chapter focuses on Topological Remedial Actions, where particular steps are detailed. This chapter is divided into 5 sections captured in the figure 3.1.



1. Topology problem statement
2. Analysis of currently used methods
Depth-first search (DFS)
Breadth-first search (BFS)
3. Application of genetic algorithms
Theoretical Framework
Genetic algorithm - Roulette wheel selection
Particle swarm optimization
4. Validation & testing
Design of experiments
Depth-first search (DFS)
Breadth-first search (BFS)
Particle swarm optimization
GA - Roulette wheel selection
5. Evaluation of results
Results assessment

Figure 3.1: Methodology of Topological Remedial Action

3.1 Topology problem statement

This section is devoted to the Topological Remedial Actions (RA). This is one of three RA that were considered to solve the problem detailed in the section 1.3.

Topological RA approach is focused on changes in a power network. The change can be generally following [7]:

- opening/closing of one or more network elements
- node reconfiguration
- cancellation of outages

In case of our topology statement, the option opening/closing of one or more network elements was only considered. For example, network elements can be branches, cables, transformers, bus bar couplers, etc. [7] In our case, only branches in a power network were considered.
Topological RA are usually made by TSOs. TSOs must consider only network elements that may be changed. If topological actions are not correctly taken into account, many simulation results can be affected (e.g. operational security, redispatching costs, market coupling results, etc.) [8]

3.2 Analysis of currently used methods

There are lots of ways for for realization of switching strategies, which can be considered as exploring nodes in a state-space. The most intuitive ones are Depth First Search (DFS) and Breadth First Search (BFS). The reason is primarily simplicity and quality. They are detailed in the next sections 3.2.1, 3.2.2

3.2.1 Breadth First Search (BFS)

BFS principle traverses nodes in a progressive and ordered way, by levels. It starts at the initial node or at random node of a search-space(graph) and explores all of the neighbour nodes at the present depth prior to moving on to the nodes at the next depth level. [12]
There are two main steps required to traverse the graph breadthwise as follows:

1. First move horizontally and visit all the nodes of the current layer
2. Move to the next layer

The figure 3.2 illustrates BFS principle applied to our case. In our Test Case a node represents a certain topology and the whole graph specify a search-space containing all possible topologies. In the figure 3.2, the node marked with a number 1-6 represents a binary search-space of length: $n = 186$. Each parameter S_i can contain value 0 (branch OFF) or 1 (branch ON) representing the branch status.

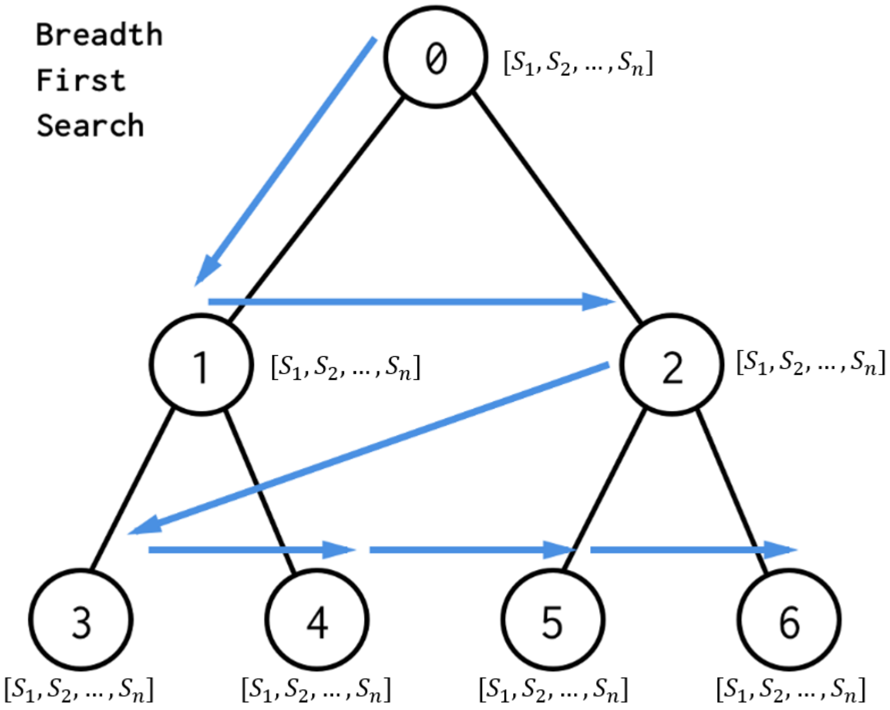


Figure 3.2: BFS principle [27]

The number of searched nodes in the graph at BFS principle is determined by the following equation:

$$t^d \leq 2^t, \tag{3.1}$$

where t is the amount of changeable parameters(i.e. branch status) on nodes and d is the depth level of searched graph. The number can never exceed the value 2^t representing the total possible topology combinations.

This principle was not used due to a huge memory and computational requirements, but the exact calculations are presented in the section 3.4.

3.2.2 Depth First Search (DFS)

The DFS stands for an another principle that traverses nodes in a more recursive way that uses the idea of backtracking. This principle goes to the deepest level from neighbour to neighbour before backtracking. DFS starts the traversal from the root node and visits nodes as far as possible from the root node. It generally requires less memory than BFS. The principle could be used for our Test Case and it is detailed in the section 3.4.2. The figure 3.3 generally shows the DFS principle. [13].

In our case, the DFS principle due to a huge state space was changed. Only the best topologies of each depth of the graph was chosen and further searched at the following depths. A graphical representation is shown in the figure 3.3, where the nodes 0, 1 and 4 marked in green represents the best nodes of every depth. Due to a huge memory and computational requirements, the changed algorithm went directly through the graph with out backtracking.

```
1. set maximum depth
2. set initial search space
3. loop through depth and search space
4.       compute RAM on CB
5.       choose the best result
6.       remember the best result for next loop
7.       stop if required RAM value found or maximum depth reached
```

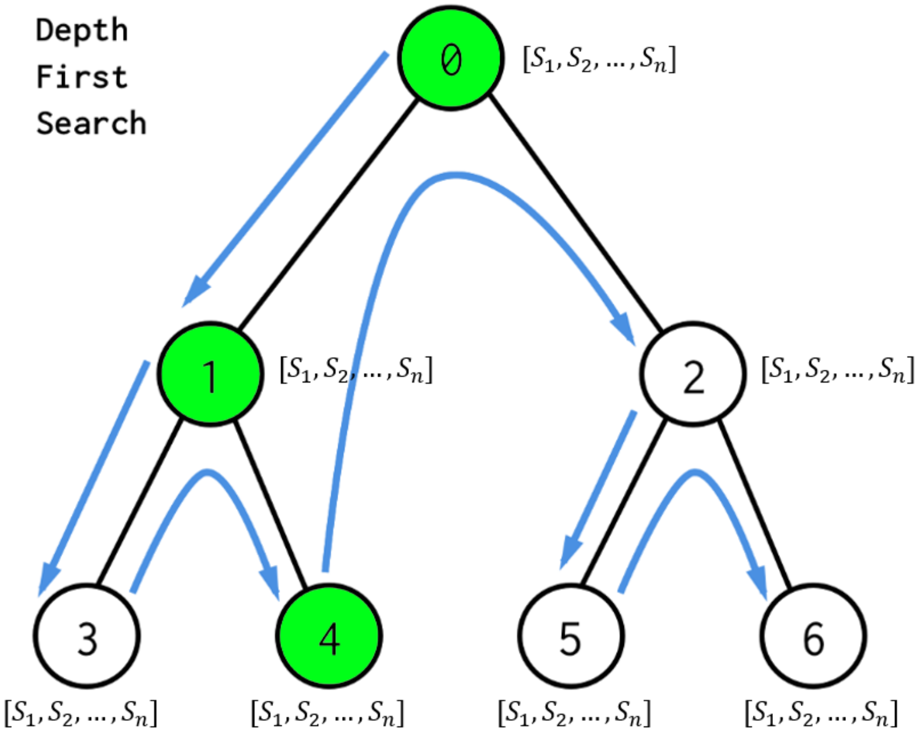


Figure 3.3: DFS principle [27]

The number of searched nodes in the graph at DFS principle is determined by this equation:

$$t \cdot d, \tag{3.2}$$

where t is the amount of changeable parameters on nodes and d is the depth level of searched graph.

3.3 Application of evolutionary algorithms

3.3.1 Theoretical framework

Evolutionary algorithms (EAs) are a family of biologically-inspired algorithms. They are primarily applied to solving problems, where conventional optimization methods can not be used. The main constraints can be a huge number of dimensions or complexity of the problem. do not work well or they can not be used at all due to huge time requirements. *Solving these problems would require exceeding the available resources. In other words, given infinite resources and infinite compute capability it could be possible for a traditional exploitative or stochastic algorithm to reach a conclusion.*[34]

EAs are based on an a population of entities(potential solutions) that evolves. This evolving process taking place on these entities consist of three main actions shows in the figure 3.4:

- Replication: Action, where a completely new generation is created. The generation can be either absolutely random or a some specific (especially good individuals) would be combine.
- Variation: In this part, the created population mutates or combines(crossover). The reason for it is creation a diversity of individuals within a population. Crossover, creates new entities by combining parts of other entities. Mutation may be defined as a small random change, to get a new solution.
- Selection: Selection is based on Darwin's natural selection, or Survival Of The Fittest, where selected entities that show the most promise are carried forward, with variation on the selection being used to create the children of the next generation [34].

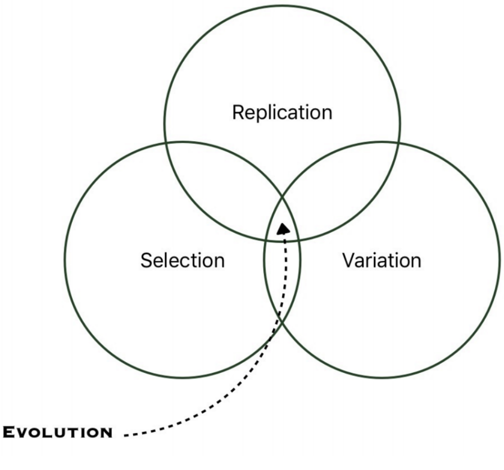


Figure 3.4: Idealized Darwinian Evolution [34]

As a recap, digital evolution is one which a population of entities goes through generational changes. Each change starts with a selection from the previous generation. Each entity is evaluated against a known specific goal i.e. the fitness is established and used as input to the selection algorithm. Once a selection is made, replication occurs with different degrees of variation. The variation is either by some form of recombination from the parent selection and/or some stochastic mutation.

When these actions are transferred into a digital form processed by a computing unit, the actions of EAs shown in the figure 3.5 are sorted as follows. *Digital evolution is one which a population of entities goes through generational changes. Each change starts with a selection from the previous generation. Each entity is evaluated against a known specific goal i.e. the fitness is established and used as input to the selection algorithm. Once a selection is made, replication occurs with different degrees of variation. The variation is either by some form of recombination from the parent selection and/or some stochastic mutation.* [34]

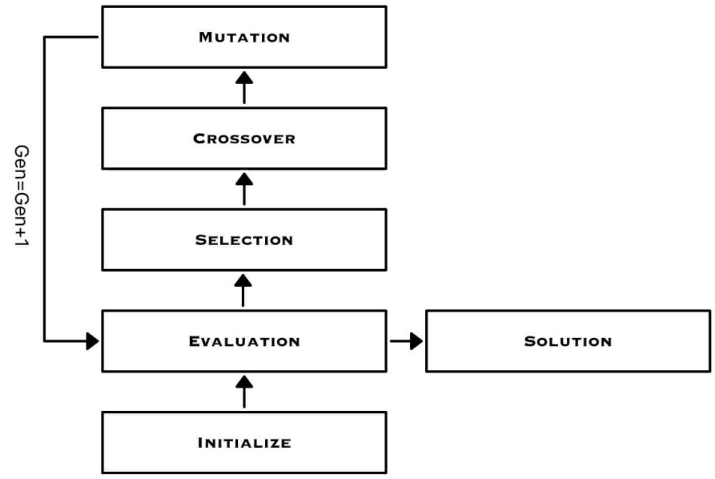


Figure 3.5: Basic Digital Process [34]

3.3.2 Genetic algorithms - Roulette wheel selection (GA - RW)

Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems. GAs handle optimization and configuration problems where there are too many variables or parameters for a traditional method to succeed. [21]

For GA are the most important following elements, which are shown in a figure 3.7

- Population - contains all individuals (chromosomes) which can be either a parent or an offspring
- Chromosome - an individual in a current population consisting of N Genes
- Gene - the smallest element having a single value

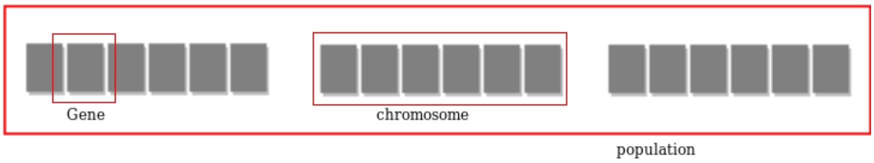


Figure 3.6: Search space [21]

GA algorithms differs from each other according to the applied Genetic operators. There are three considered operators:

1. Selection Operator
   The idea of selection is to give preference to the individuals with good fitness values and allow them to pass there genes to the successive generations. If an individuals are selected, they become parents and they move to the crossover part.
   There exist many different algorithms used for a selection:
   - Roulette wheel selection
   - $(\mu + \lambda)$ selection
   - Tournament selection
   - Truncation selection
   - Elitist selection
   - Ranking and scaling
   - Sharing
2. Crossover Operator
   This represents mating between individuals. During this phase, the individuals selected in the previous phase are crossed or mixed. That is, the genes of the two parents are mixed together to give rise to the different children. There are several methods of crossing, but the most used are the following:

   - One Point Crossover
   - Multi Point Crossover
   - Uniform Crossover

3. Mutation Operator
   The key idea is to insert random genes in offspring to maintain the diversity in population to avoid the premature convergence. The most used operators are:

   - Bit Flip Mutation
   - Swap Mutation
   - Scramble Mutation

From the previous description, there exists a huge amount of possible combinations. For our Test case were used GA with these operators:

- Selection:
  - Roulette wheel selection
  - Improved roulette wheel selection
- Crossover:
  - One Point Crossover
  - Multi Point Crossover
- Mutation:
  - Bit Flip

Roulette wheel

Roulette wheel selection, proposed by Holland [14], is the best known selection type. The basic idea is to determine selection probability or survival probability for each chromosome proportional to the fitness value. Then a model roulette wheel can be made displaying these probabilities. The selection process is based on spinning the wheel the number of times equal to population size, each selecting a single chromosome for the new procedure. The algorithm is graphically shown below:

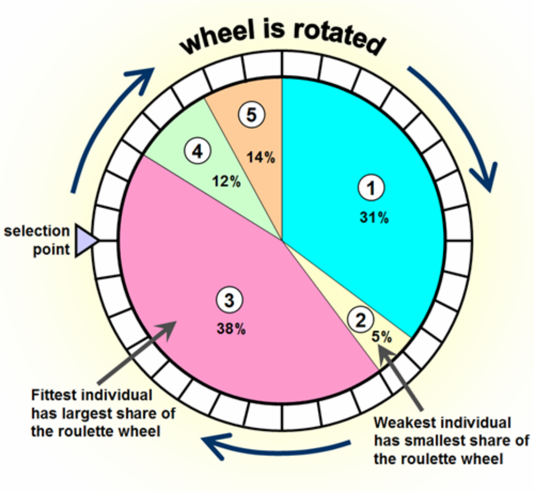

wheel is rotated
selection point
1
31%
2
5%
3
38%
4
12%
5
14%
Fittest individual has largest share of the roulette wheel
Weakest individual has smallest share of the roulette wheel

Figure 3.7: Roulette wheel principle [15]

In the roulette wheel algorithm, following steps were used: These steps were used:

1. Calculate S = the sum of a finesses
2. Generate a random number between 0 and S.
3. Starting from the top of the population, keep adding the finesses to the partial sum P, till $P < S$
4. The individual for which P exceeds S is the chosen individual

During the validation, the original algorithm provided required results, however some problems were occurred.

- the achievement of an acceptable solution took a very long time (more then 1000 generations)
- huge range of provided results
- non-still improving
- the best found parents/offsprings do not continue to the next generation

Due to these disadvantages, the original algorithm was improved, which is described in the next subsection.

Roulette wheel + Best parents

The biggest problem of the previous method was that the best found parents/offsprings do not continue to the next generation. To solve this problem was this algorithm improved by following steps:

- Calculate S = the sum of a finesses
- Generate a random number between 0 and S.
- Starting from the top of the population, keep adding the finesses to the partial sum P, till $P < S$
- The individual for which P exceeds S is the chosen individual
- Save the best N offsprings for the next generation
- Remove the worst N offsprings and replace them by the best offpsrings from previous one

The modification provided a big improvement in computation performance shown in the section Validating & testing.

3.3.3 Particle swarm optimization

The another algorithm of genetic algorithms is a Particle swarm optimization (PSO). It was chosen due to many good references when using this method for any optimization in power networks.
PSO is an optimization method that is population-based and was first developed in 1995 by Dr. James Kennedy and Dr. Russell Eberhart.[16] The main idea of this method is using population of particles creating a swarms. Within them is search he optimal solution of current problem. The search starts by considering each particle as the candidate solution. The PSO algorithm is inspired by the behaviour of humans, fish, insects, or flock of birds, where the individuals look for the best solution in a problem dimensional space. The key terms used in this method are: Particle, Swarm, Fitness,p_{best} and g_{best}. They are similar to the terms from previous genetic algorithm and they are described below.

- Particle: An individual within the swarm. In comparison to previous methods of genetic algorithms, the particle is like an individual/chromosome.
- Swarm: Total population of particles. A swarm is like the whole generation containing N numbers of particles.
- Fitness: A function that gives the interface between the optimization problem and that physical problem, and amplifies the accuracy of the solution within the position in the solution space. In our case, this value represented the RAM value calculated on CB.
- Position: A particle's dimensional coordinates representing the solution to the problem. For our case, the Position was specified by a matrix:
  X_{sp}, where s = size of a swarm, p = length of a particle

- Velocity: This value represents a positional change of a particle. In out case velocity is specified by a matrix V_{sp} with same dimensions as at Position

- P_{best}: The position in the parameter space of the best fitness returned for a specific particle.

- G_{best}: The position in the parameter space of the best fitness returned for the entire swarm.

Because of the nature of our problem, where the Particle stands for a binary vector the PSO algorithm had to be changed to Binary Particle Swarm optimization(BPSO). Therefore we get the particles ordered in Position matrix X_{sp} containing only discrete values$[0, 1]$.
The most important calculations of BPSO are:

1. Inertia weight calculation

$$w = w_{max} - \frac{w_{max} - w_{min}}{it_{max}} \cdot it, \tag{3.3}$$

   where w_{max} and w_{min} are the maximum and minimum inertia weights, it_{max} is the maximum iteration count and it is the current iteration.

2. Velocity calculation

$$V_{ij}^{t+1} = w \cdot V_{ij}^{t} + c_1 m_{1j}^{t} \cdot [P_{best} - X_{ij}^{t}] + c_2 m_{2j}^{t} \cdot [G_{best} - X_{ij}^{t}], \tag{3.4}$$

   where
   X_{ij}^t is the particle's position
   V_{ij}^t is the particle's velocity
   c_1 is the cognitive parameter c_2 us the social parameter m_1, m_2 are the random values from the interval $[0, 1]$;

3. Position calculation

$$X_{ij}^{t+1} = \begin{cases} 1, & \text{if } u_{ij}^t < s_{ij}^t \\ 0, & \text{if } u_{ij}^t \geq s_{ij}^t, \end{cases} \tag{3.5}$$

   where
   u_{ij}^t is the randomly selected value from a uniform distribution in (0,1)
   s_{ij}^t is the sigmoid function calculated:

$$s_{ij}^t = \frac{1}{1 + e^{-V_{ij}^{t+1}}} \tag{3.6}$$

Algorithm

The most important steps of PSO algorithm are shown in the fig 3.8, which has been taken from [23].

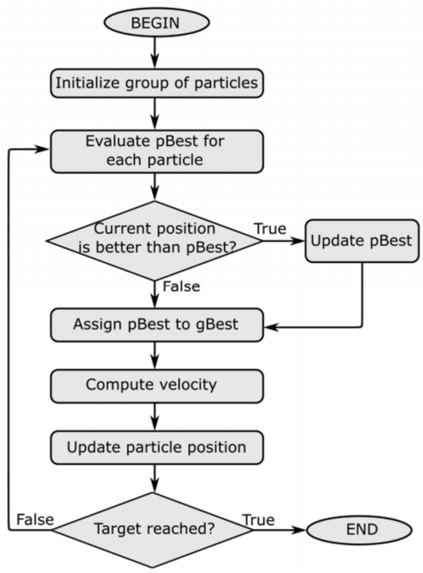


Figure 3.8: Flowchart of the Particle Swarm Optimization algorithm

3.4 Validation & testing

3.4.1 Design of experiments

As mentioned the BPSO and GA methods belong to the group of Evolutionary algorithms, which are characterized by uncertainty of convergence time over multiple executions. The time and a general quality of a single run depends on a deployment of randomly generated initial population in the state space and particular algorithms loops improving an objective function in a nondeterministic way. Therefore these aspects need to be considered during the validation phase. Moreover, heuristic algorithms (i.e. evolutionary) strongly depend on the initial configuration of tuning parameters, which stands for significant degree of freedom for overall computation performance.

Convergence properties of algorithms presented in previous sections will be assessed in this part. The validation will focuses on the assessment of convergence properties and algorithm performance under various initial configuration and under consideration of nondeterministic character of evolutionary algorithms. During experiments, evolutionary algorithms (BPSO and GA) will be run twenty times with different initial population in order to assess the convergence properties robustly. For the acceleration of experiments, the Parallel Computing Toolbox(PCT) detailed in the section 3.4.1 will be used. Both algorithms (BPSO and GA) contains several configuration parameters, where three most significant ones will be analyzed and detailed in relevant sections below.

Software environment

For the implementation of software solution, the MATLAB R2018a programming language was used, where following toolboxes were used:

- MATPOWER
  A free package of M-files initially developed by Ray D. Zimmerman, Carlos E. Murillo Sánchez and Deqiang Gan of PSERC at Cornell University under the direction of Robert J. Thomas. MATPOWER is used for solving power flow, continuation power flow and optimal power flow problems. [30]
- Parallel Computing Toolbox (PCT)
  PCT allows to run many calculations simultaneously and solve computationally and data-intensive problems using multicore processors, GPUs, and computer clusters. [26]

Hardware environment

All methods were calculated on a desktop computer HP Pavilion Gaming 690-0015nc with following parameters:

- Processor: Intel Core i7 8700 Coffee Lake 4.6 GHz, 6 cores, 12 threads
- Installed memory (RAM): 16 GB

During experiments, all six cores were used in parallel calculations.

Test Case

The test case stands for a power network with real electrical properties and relevant size reflecting typical use of the remedial action optimizer. The test case contains these parameters:

- Number of buses: 846
- Number of generators: 216
- Number of all branches: 1634
- Number of selected branches for topology actions: 186

In the case of Topology optimization (TO), only topological actions related to branches in a power network were considered. Every branch contained many parameters, e.g. from bus number, to bus number, resistance, reactance, flow limits and branch status. In TO, the binary branch status (1 - in service, 0 - out of service) is the most important parameter, which change the power network topology. Changing the topology in a desired (i.e. optimal) way stands for a complex problem depending on the size of a simulated power network. Simulation of topological actions is a big challenge specially when large areas are simulated. In these cases is a set of possible solutions huge, which requires large computation effort. In following sections, the computation performance of particular algorithms will be analyzed.

3.4.2 DFS

As mentioned earlier, the DFS principle is more less deterministic and provides always the same values at the exact times due to an algorithm that is not affected by randomness. However, the computation time can be affected bu the operation system processes, therefore DFS was run 20 times in parallel to achieve the same robust result set eliminating the randomness. Results obtained from testing can be summarized as:

- The average time of 1 iteration: 0.5172 s.
- The number of iterations: 15
- The total time: $15 \cdot 0.5172 = 7.7576$

The values of the objective function (RAM) and computation times in particular steps are written in the table 3.1 and graphically represented in the figure 3.9 and 3.10.

92.68 **(0.5172s)**	139.66 **(3.1032s)**	143.92 **(5.6892s)**
123.95 **(1.0344s)**	140.89 **(3.6204s)**	143.95 **(6.2064s)**
133.49 **(1.5516s)**	141.88 **(4.1376s)**	143.97 **(6.7236s)**
136.65 **(2.0688s)**	143.14 **(4.6548s)**	143.98 **(7.2408s)**
138.43 **(2.5860s)**	143.64 **(5.1720s)**	144.13 **(7.7576s)**

Table 3.1: Values in time of DFS algorithm

DFS iterations

In the figure 3.9, the stair graph representing evolution of the RAM value per iteration is shown. The X-axis represents number of iterations and the Y-axis stands for the RAM value. Based on the achieved results, DFS algorithm provides very good results, specially at the initial phase of the algorithm. In the first iteration, 186 topologies were applied and the only 1 branch was switch on, the DFS algorithm improved the result up to 64.36 % of the required value. In the second iteration, where additional branch was turned on, the improvement of RAM value was significant as well (additional 33.74% improvement). The last perceptible improvement was achieved in third iteration, where the final RAM value was equal to 133.49MW representing 92.70 % of desired quality level. From the fifth to the ninth iteration, there is a relatively small improvement in range 1 - 2 MW. The last 5 iterations shows a very small but growing improvement.

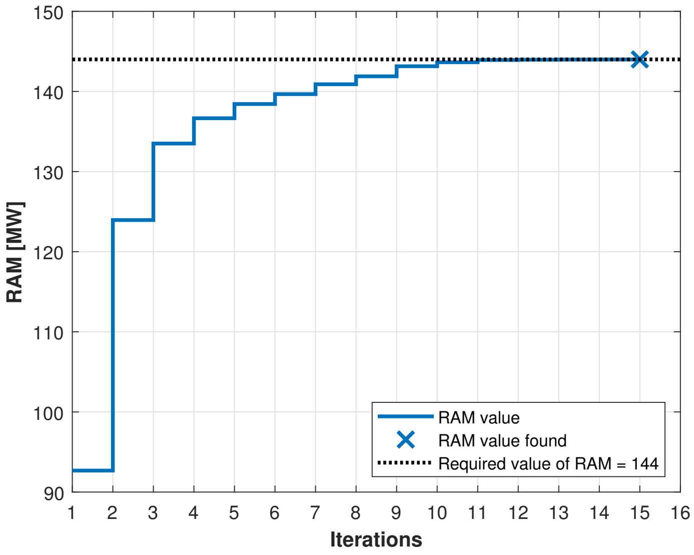


Figure 3.9: The evolution of RAM value over DFS iterations

DFS time

Although, the number of iterations is low, it does not represent the results precisely. To compare the DFS with EAs, compare values over time is needed as shown in the figure 3.10, where time aspects are shown.
The final comparison of DFS algorithm with EAs is described in the section 3.5.

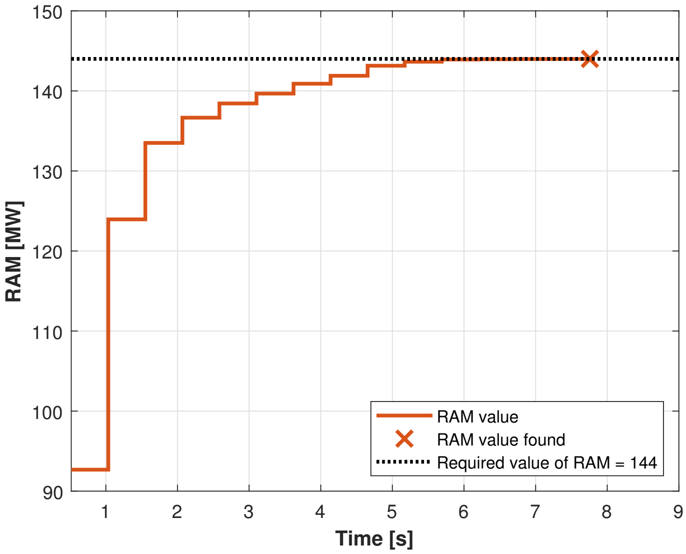


Figure 3.10: The evolution of RAM value over DFS time

It can be seen, from the figure, the computation time is demanding because of high number of evaluation of the objective function per iteration.

3.4.3 BFS

As mentioned in the section 3.2.1, this approach will be not used due to the high time and huge memory requirements. There is an exponential growth of loops needed to search in the binary space, were is amount can be expressed as

$$t^d \leq 2^t, \tag{3.7}$$

where $t = 186$ is the amount of changeable parameters on nodes and d is the depth level of searched graph. The number can never exceed the value 2^t representing the total possible searches. If the depth is set to 3, the total number of objective function evaluation[1] would be equal to

$$186^3 = 6434856. \tag{3.8}$$

The total time to search the entire state space of the test case can be expressed from DFS assessment mentioned in the previous section, where time demanded for one iteration is equal to 0.5172 s on the hardware specified in the section 3.4.1. The total computation time of BFS execution with depth three can be expressed as

$$0.5172 \cdot 6434856 = 3328107.52s = 924.47h \tag{3.9}$$

Therefore this method is practically non-applicable for real world applications and will not be compared with other methods.

[1] Load flow computation and RAM evaluation.

3.4.4 Binary particle swarm optimization

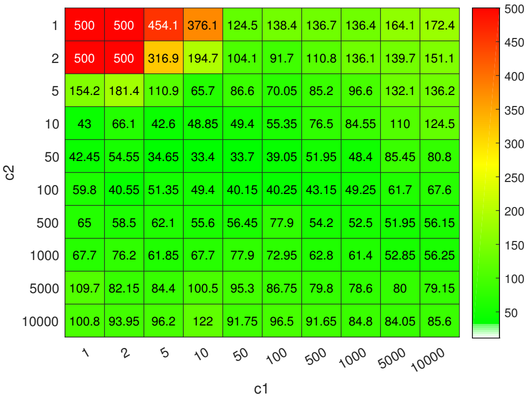
Binary particle swarm optimization(BPSO) method contains several configuration parameters, where three most influencing ones were analyzed in this section. The tested values of BPSO parameters mentioned in the section 3.3.3 were:

- N - size of a swarm: [5 20]
- c_1 - social acceleration coefficient: [1 2 5 10 50 100 500 1000 5000 10000]
- c_2 - cognitive acceleration coefficient: [1 2 5 10 50 100 500 1000 5000 10000]

First, the BPSO algorithm was tested on a smaller number of particles in a swarm (N = 5), where the main advantage compared to size equal to 20 is the higher speed of one iteration. On the other hand, the disadvantage is the lower aggressiveness of the state space search compared to the size N = 20, because the state space is searched in a smaller number of directions. Which of the mentioned advantages is stronger and thus better results are achieved is analyzed.

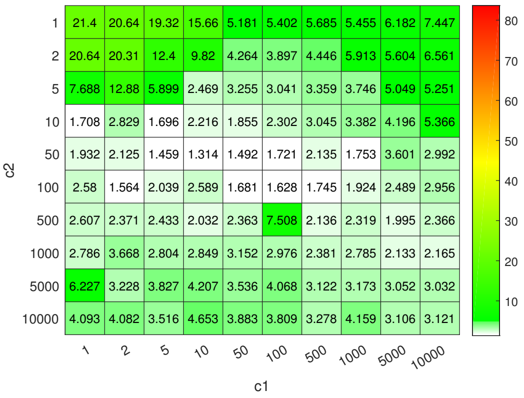
Number of particles N = 5

The figure 3.11 represented by a table shows the comparison of the average number of iterations under consideration of different configuration parameters. For a better measurability, the number of iterations was converted to the average total time displayed in the right figure 3.12. On the x and y axes, the coefficients c_1 and c_2 are shown. It can be seen from both tables that the dominant parameter is c_2 representing the aggressive behaviour throughout the whole state space. The extreme values on both sides of c_2 parameter had bad results, where the absolute worst configurations were found with parameters c_1,c_2 = 1 or 2. In that case, the BPSO algorithm did not ever find the required RAM value in any of the 20 parallel runs, as the maximum number of iterations was set to 500. The results can be seen more clearly from the graph 3.12, where configurations containing the parameter c_2 equal to 50 generally reached the shortest time. Absolutely the best time = 1.314s had the configuration c_1 = 10, c_2 = 50. The average 1 iteration time of all runs containing the number of particles equal to 5 was 0.0422 seconds.



	1	2	5	10	50	100	500	1000	5000	10000
1	500	500	454.1	376.1	124.5	138.4	136.7	136.4	164.1	172.4
2	500	500	316.9	194.7	104.1	91.7	110.8	136.1	139.7	151.1
5	154.2	181.4	110.9	65.7	86.6	70.05	85.2	96.6	132.1	136.2
10	43	66.1	42.6	48.85	49.4	55.35	76.5	84.55	110	124.5
50	42.45	54.55	34.65	33.4	33.7	39.05	51.95	48.4	85.45	80.8
100	59.8	40.55	51.35	49.4	40.15	40.25	43.15	49.25	61.7	67.6
500	65	58.5	62.1	55.6	56.45	77.9	54.2	52.5	51.95	56.15
1000	67.7	76.2	61.85	67.7	77.9	72.95	62.8	61.4	52.85	56.25
5000	109.7	82.15	84.4	100.5	95.3	86.75	79.8	78.6	80	79.15
10000	100.8	93.95	96.2	122	91.75	96.5	91.65	84.8	84.05	85.6

Figure 3.11: Average number of iterations for N = 5



	1	2	5	10	50	100	500	1000	5000	10000
1	21.4	20.64	19.32	15.66	5.181	5.402	5.685	5.455	6.182	7.447
2	20.64	20.31	12.4	9.82	4.264	3.897	4.446	5.913	5.604	6.561
5	7.688	12.88	5.899	2.469	3.255	3.041	3.359	3.746	5.049	5.251
10	1.708	2.829	1.696	2.216	1.855	2.302	3.045	3.382	4.196	5.366
50	1.932	2.125	1.459	1.314	1.492	1.721	2.135	1.753	3.601	2.992
100	2.58	1.564	2.039	2.589	1.681	1.628	1.745	1.924	2.489	2.956
500	2.607	2.371	2.433	2.032	2.363	7.508	2.136	2.319	1.995	2.366
1000	2.786	3.668	2.804	2.849	3.152	2.976	2.381	2.785	2.133	2.165
5000	6.227	3.228	3.827	4.207	3.536	4.068	3.122	3.173	3.052	3.032
10000	4.093	4.082	3.516	4.653	3.883	3.809	3.278	4.159	3.106	3.121

Figure 3.12: Average computation time for N = 5

Number of particles N = 20

In this case, the number of particles in a swarm was increased to 20, which resulted in an increase in the time of one iteration. The average time of 1 iteration was 0.1666 seconds. From the figure 3.13, an obvious presumed decrease of all iterations can be seen. This is caused by exploring larger area of the state space with the higher number of particles. On the other hand, the higher number of particles increases computation time as drafted in the figure 3.14. Configurations with number of particles equal to 20 have similar tendencies as in the previous case for N equal 5. The best solutions were located in the approximately in the middle of the table, where the c_2 parameter again proved to be the best one.

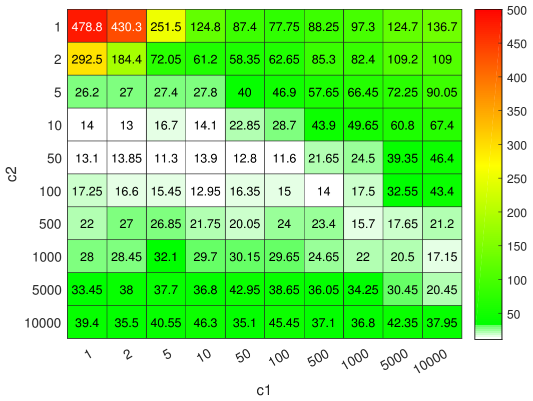


Figure 3.13: Average number of iterations

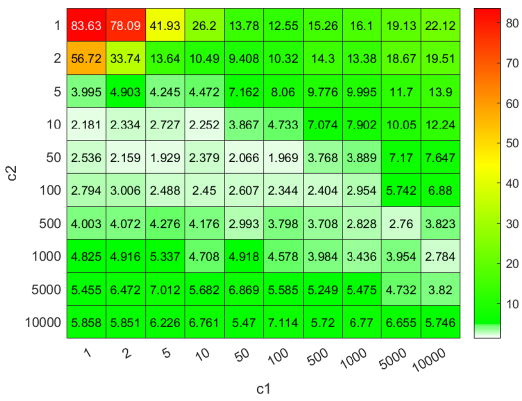


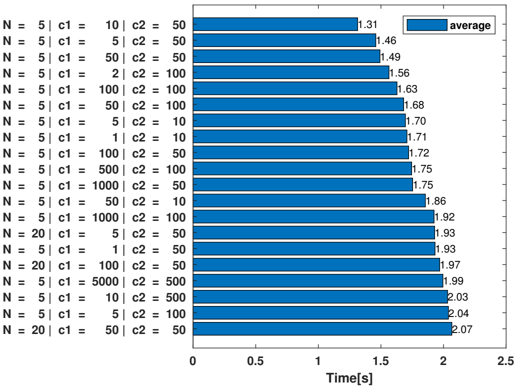
Figure 3.14: Average computation time

The comparison between two executions with different number of particles is shown in the following figures 3.15, 3.16 and 3.17.

Performance comparison of various configurations

Average time comparison

In this section, the computation performance of the BPSO algorithm is assessed for various configurations. The figure 3.15 displays 20 configurations with best computation performance sorted by average time. Based on results, better performance were achieved by the number of particles equal 5. The top 13 solutions were only for N = 5. The biggest time difference was registered between the first and the second coefficients configuration. Coefficients at the position 6 to 11 had almost the same time. For a better overview, in the figure 3.16, the values are displayed in percent.



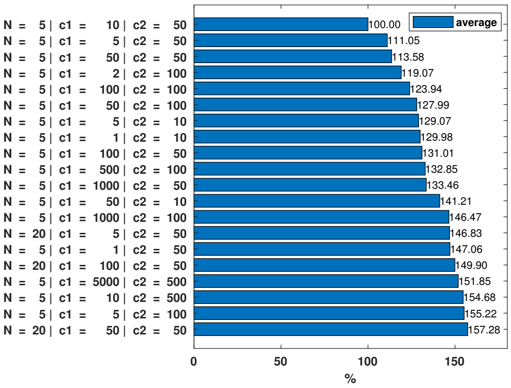
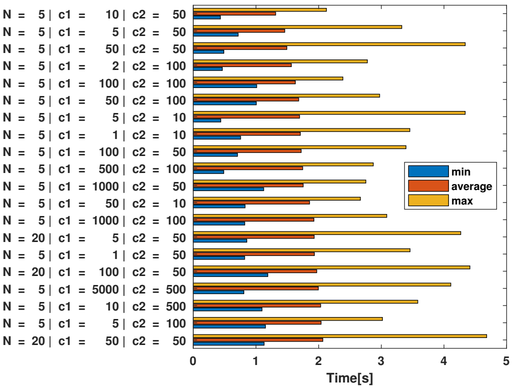


Figure 3.15: The best 20 configurations sorted by time (seconds)

Figure 3.16: The best 20 configurations sorted by time (percent)

Boundary computation time comparison

In this part, boundary computation time are analyzed, whose 20 configuration with best performing computation time are depicted in the next figure 3.17. In this bar chart, the boundary values (minimum and maximum) are displayed. The MIN value represents the fastest solution recorded form the 20 parallel loops. The MAX values stands for the solution with the worst time. The attention should be focused specially on the MAX values. Based on the figure, the worst time (longer than 4s) appears always for the configurations with N equal to 20.



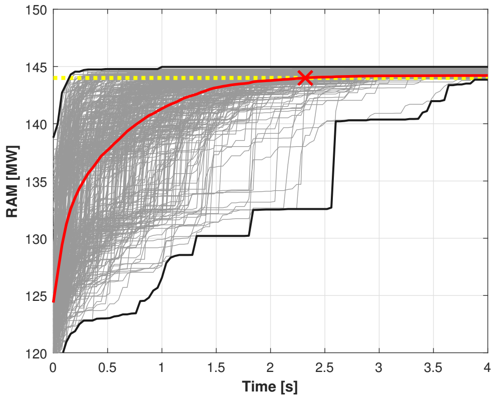


Figure 3.17: The best 20 configurations with boundary values (seconds)

Figure 3.18: Objective function evolution for all realizations

BPSO Realizations

On the last figure 3.18, objective function evolutions are displayed for all 20 realizations under consideration of the best configurations. There are depicted all 20 parallel loops for each configuration (in total 400 graphs). These 400 executions are grey colored. The black colored edges represent the boundary values of all loops, while the red colored line shows the progress over time of the average value. The last yellow dashed line specify the required RAM value equal to 144 MW.

3.4.5 Genetic algorithms - Roulette wheel selection (GA - RW)

In this section, roulette wheel selection algorithm and its modifications are analyzed in the similar was BPSO in the previous section. Here, genetic algorithm were tested for different sizes of populations, which strongly depends on number of best parents proceeding to next generation (n_{BP}) and various crossover methods. Two types of the crossover methods were analyzed: One Point Crossover, Multi Point Crossover. The overall amount of configurations was reduced to 25 due to higher computation requirements and considered configuration can be summarized as

- N - size of a population: 6
  - n_{BP} - number of Best parents: [0 1 2 3 4]
  - n_{CO} - number of Crossover: [1 2 3 5 10]
- N - size of a population: 20
  - n_{BP} - number of Best parents: [0 4 8 12 16]
  - n_{CO} - number of Crossover: [1 2 3 5 10]

Population size N = 6
The chart on the figure 3.19 shows the comparison of the average amount of iterations. On the x and y axis, coefficients n_{BP} and n_{CO} are displayed. On the x axis, results from the original roulette wheel algorithm are located in the first column ($n_{BP} = 0$) and results from modified methods are located in other columns. The lowest number of iterations for the population N = 6 was found for the parameters $n_{BP} = 1$ and $n_{CO} = 3$. The maximum number of iterations was set to 500. As can be seen, the original GA method did not found the required solution in 500 iterations. Because the convergence performance is crucial for practical solutions, the original method was improved by propagation of best parents to next generation, which significantly improved the convergence rate.

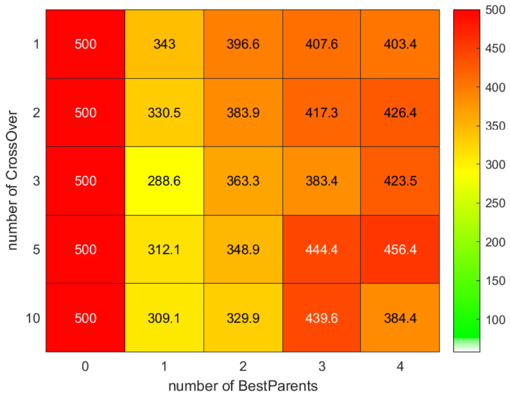


Figure 3.19: Average number of iterations

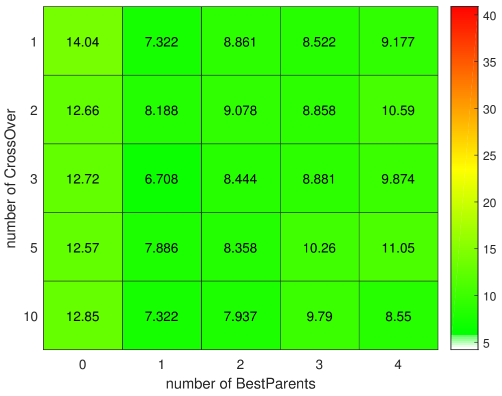


Figure 3.20: Average computation time

The average computation time of GA under consideration of various configuration are displayed in the figure 3.20. The longest computation time values are as expected located in the first column = 0 Best Parents, which relates to the original algorithm. Generally, there are not significant differences between various configurations. The best found configuration - $n_{BP} = 1$ and $n_{CO} = 3$ had the lowest average computation time 6.708s. The average 1 iteration time of all runs containing the number of particles equal to 5 was 0.0237 seconds.

Population size N = 20

The analysis of computation performance with increased population equal to 20 is depicted on the figure 3.21. As can be seen, the lowest value of iteration has the configuration - $nBO = 12$ and $n_{CO} = 5$. As similarly to BPSO method, the increase of population size led to significant reduction of iterations, while increases the computation time on the other hand. In order to find out the best population size , it is necessary to convert numbers of iterations to computation time.

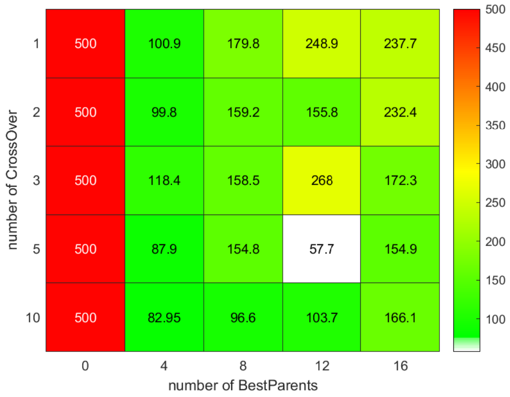


Figure 3.21: Average number of iterations

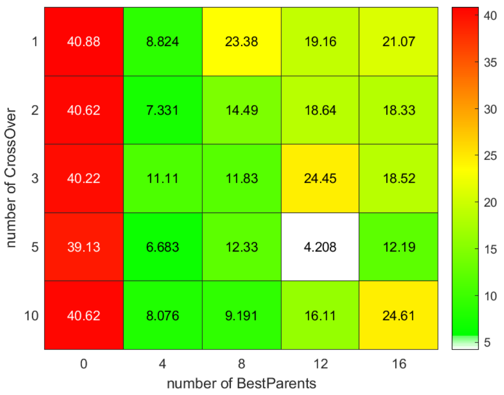


Figure 3.22: Average computation time

The next chart in the figure 3.22 shows time values. There are clear differences from the previous chart in the figure 3.20. The column representing 0 Best Parents is also the worst one, the time of the maximum iteration has supposedly increased due to higher time requirements of 1 loop. Since the number of CrossOver is greatly affected by chance, the values according to the individual columns are especially noteworthy interesting. If the values in individual columns are summed, following results depending on Number of Best Parents are obtained.

- $n_{BP} = 0$ → NOT FOUND
- $n_{BP} = 4$ → total time $\approx$ 42 seconds
- $n_{BP} = 8$ → total time $\approx$ 71 seconds
- $n_{BP} = 12$ → total time $\approx$ 82 seconds
- $n_{BP} = 16$ → total time $\approx$ 95 seconds

As can seen from previous calculations, the increase of computation time depends on raising number of best parents for next generation n_{BP}. The best computation performance is with $n_{BP} = 4$. In comparison to $n_{BP} = 12$ (the best computation time), the configuration with $n_{BP} = 4$ ended in much better average time. Therefore, the GA with configuration $n_{BP} = 12$ and $c_{CO} = 5$ was probably affected by very suitable initial conditions.

Performance comparison of various configurations

Average time comparison

Similar to BPSPO algorithm, the next charts in the figures 3.23 and 3.24 shows the top 20 average time values sorted from best to worst. On the left side, configurations with specific parameters are labeled. The fastest time of finding the required RAM value was reached by the configuration: population N = 20, $nBO = 12$ and $n_{CO} = 5$. The largest difference of 58.83 % was found between the first and second value shown in the figure 3.24.

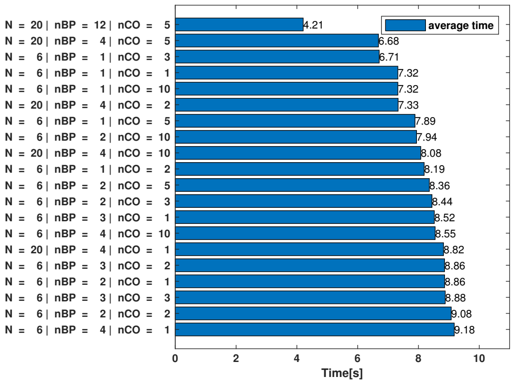


Figure 3.23: The best 20 configurations sorted by time (seconds)

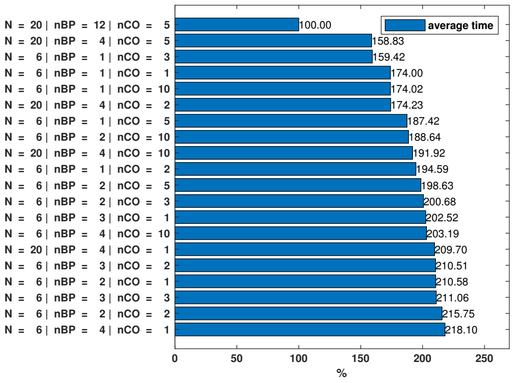


Figure 3.24: The best 20 configurations sorted by time (percent)

Boundary computation time comparison

The figure 3.25 shows the bar chart sorted by the average time as in the previous figures 3.23 and 3.24. For detailed information, there are additionally displayed the MAX and MIN values. As seen, among top 20 configurations are only 5 with the population N = 20. All 5 configurations with population N = 20 except the first one have significantly higher maximum of maximal computation time, which represents the performance inconsistency between MIN and MAX solution. All configurations having the population equal to 6 were much more consistent in computation time. Another interesting sign of the population 20 is that all of them have $n_{BP} = 4$ (equal to 20% of the population). In the case of population size N = 20, better performing configurations are achieved when the n_{BP} is low. Therefore the best configuration for N = 12, where $n_{BP} = 12$, stands for an outlier strongly influenced by initial conditions. Due to mentioned facts, the best configuration in terms of consistency is the 3rd one - N = 6, $n_{BP} = 1$, n_{CO}= 5.

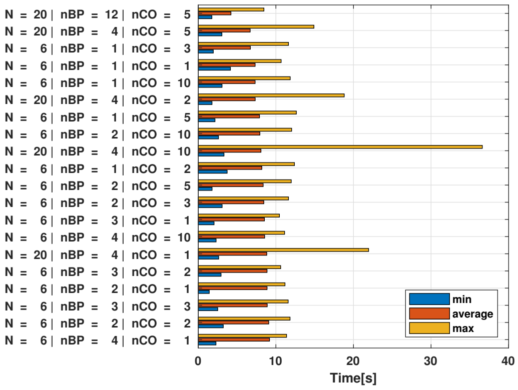


Figure 3.25: The best 20 configurations with boundary values (seconds)

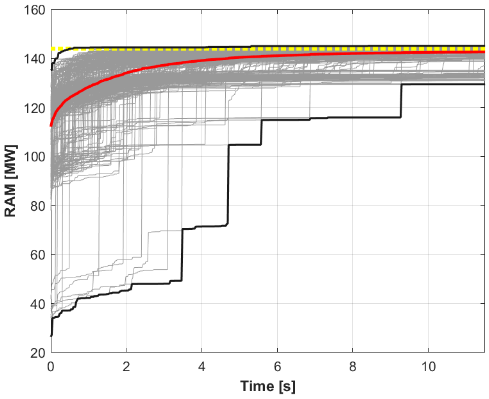


Figure 3.26: Objective function evolution for all realizations

GA Realizations
In the last figure 3.18, the best 20 configuration with 20 realization are shown. All 400 executions are grey colored. The black colored edges represent the extreme values of all loops, while the red colored line shows the progress over time of the average value. The last yellow dashed line specify the required RAM value equal to 144 MW.
In this case, the average value was not found in 11 seconds due to the high number of bad configurations.

3.5 Results assessment

In the previous section, particular aspects of selected methods were analyzed. Here, the results for best configuration of selected methods will be compared. The best found configurations of selected methods are

- DFS: Without parameters
- BPSO: N = 5, c_1 = 10, c_2 = 50
- GA(RW + BP): N = 20, n_{BP} = 12, n_{CO} = 5

For these best configuration (except DFS), twenty realization were executed and the results were stored for further assessment, which is graphically shown on figures 3.27 and 3.28. For the assessment, following metrics were taken into account

- best possible time (thin line ending in a circle)
- average time (thick line ending in a cross)
- worst possible time (thin line ending in a circle)

These values were then compared with the simple DFS method, where only one line is displayed due to zero variability. The overall comparison of all the methods used is shown in the figures 3.27 and 3.28. BPSO and GA (RW +BP) method contains three lines displayed in the figure: Each methods is display in a different color: BPSO (green), GA (RW +BP) (blue) and DFS (red).The more detailed view is shown in the figure 3.28.

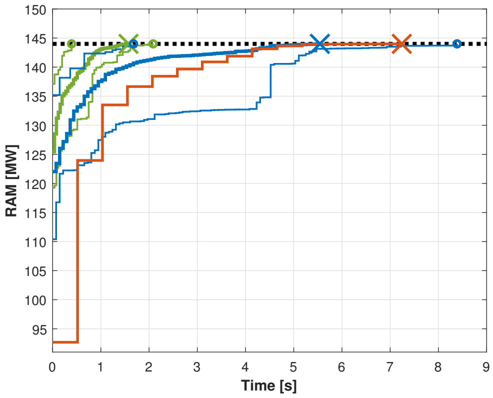


Figure 3.27: Comparison of all methods

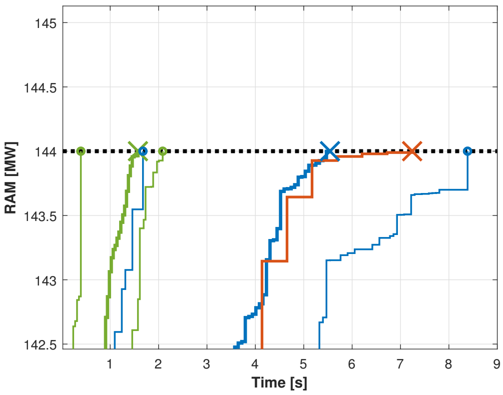


Figure 3.28: Detailed comparison of all methods

Based on the figures 3.27 and 3.28, the BPSO method has the best computation performance even the worst case is considered. The roulette wheel algorithm has significantly lower performance and DFS was the worst one. The exact values of results for all methods are shown in the table 3.2.

	BPSO	**GA**	**DFS**
average	**1.57s (41 iter)**	**5.54s (77 iter)**	**7.24s (15 iter)**
best	0.39s (11 iter)	1.68s (24 iter)	7.24s (15 iter)
worst	2.08s (54 iter)	8.39s (116 iter)	7.24s (15 iter)

Table 3.2: Comparison of all 3 methods by time

The achieved results can be summarized in a following way.

BPSO - time

- the average time is 3.53x faster than GA and 4.61x faster than DFS
- the worst time is 2.66x faster than the GA average time and 3.59x faster than DFS

- the best time is 4.30x faster than the GA best time and 18.56x faster than DFS

GA - time

- the average time is 1.30x faster than DFS
- the best time is just by 11 seconds slower than the BPSO average time
- the worst time is by 75 seconds slower than the DFS

Chapter 4

Phase Shifting Transformers optimization

4.1 PST problem statement

As written in the introduction, one of the problems that can occur in a power network is the uneven loading of parallel transfer lines. The distribution of the power flow between two parallel lines is dictated by their impedances [35]. The line with the smallest reactance carries the largest part of the load causing the situation, when one of the two lines will be operating well below its nominal rating because otherwise the parallel line would be overloaded. Due to that, power flow has to be controlled. One of the possible solution methods is the technology called Phase Shifting Transformers (PSTs), which is relatively old but it is a valuable and popular currently used solution.[40]

4.1.1 Power flow control using PSTs

The active power flow through the transmission line is given by the following equation [25]:

$$P_1 = \frac{V_1 \cdot V_2}{X_L} \cdot \sin\delta, \tag{4.1}$$

where V_1 and V_2 are the voltage modules at the sending and receiving ends of the line, respectively, δ is the power angle (the phase angle difference between V_1 and V_2), and X_L is the line reactance. The equation 4.1 shows that the active power flow through the line can be controlled by changing voltage levels V_1 and V_2, reactance X_L and power angle δ. However, the full extent of active power flow in the line can be changed by adjusting power angle δ. By controlling angle δ, not only the value of the power flow but also the direction of flow can be changed [17]. In practice this can be done by using PSTs.[18]

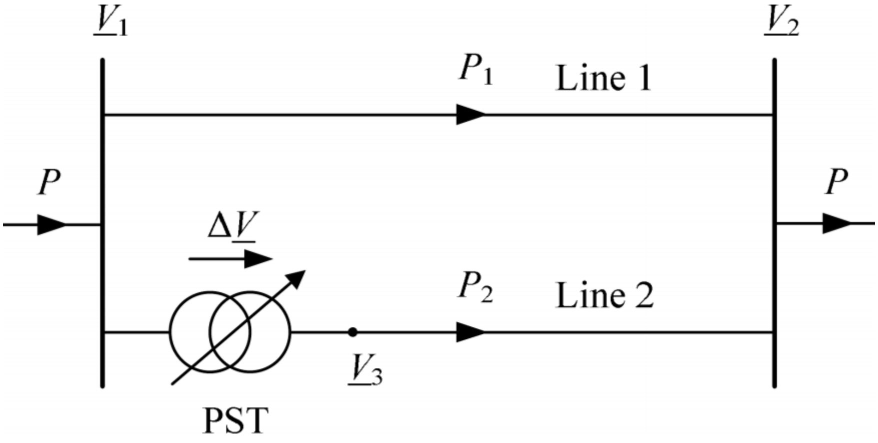


Figure 4.1: Circuit diagram using PST

In the figure 4.1 is shown a PST connected in a serious in a transfer line, where a booster voltage ΔV perpendicular to voltage V_1. The resultant voltage V_3 behinds PST is shifted in phase by angle α and the resultant power angle of the line is equal to $\delta + \alpha$ shown in th figure 4.2. The changed power P_2 using PST that flows on the Line 2 is expressed as the following equation [38]:

$$P_2 = \frac{V_3 \cdot V_2}{X_L + X_{PST}} \cdot \sin(\delta + \alpha), \tag{4.2}$$

where X_{PST} is the PST reactance, α is the PST angle.

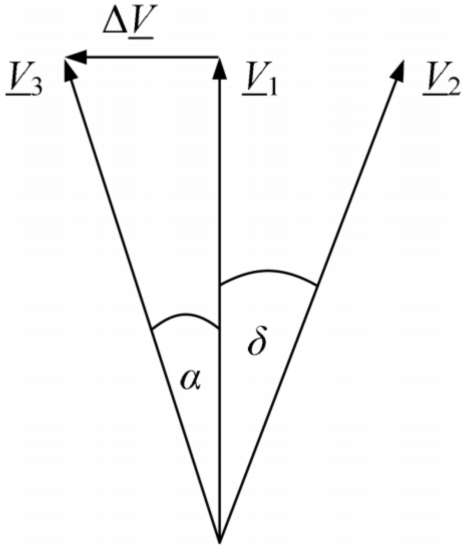


Figure 4.2: Phasor diagram for a line with a PST

The value ΔV changes the angle α causing a change of the resulting power flow P_2.

The voltage ΔV can be controlled within a range from negative values to positive values, resulting in an increase or decrease in the power angle, respectively, and thus the power P_2 [18].

For any desired angle range, the tap positions can be provided and represented in vector containing discrete values. All tap position options create a state space, where the goal is finding the maximum RAM. Due to the similarity of the state space used for Topology optimization, the same optimization methods (GA - Roulette wheel and PSO) were applied. The only change was that the state space was discrete compared to the binary one from the Topology optimization. This resulted in adjustments in both methods. The PSO algorithm was changed to an algorithm working with continuous variables, which was finally rounded. More changes have been made at GA - RW algorithm briefly detailed in the section 4.2.4. The theoretical framework is due to the equality of used methods not described again. The final optimized state space for our problem is detailed in the following section 4.2.

4.2 Validation & testing

4.2.1 Design of experiments

This section focuses on the general information about the PST optimization. The subsections Software environment, Hardware environment are completely same as in the case of Topology optimization (TO). Some changes were considered in the subsection Test Case detailed in the next section.

Test Case

The Test case contains the same properties as in TO but it was extended by 3 Phase Shifting Transformers in 3 different transfer lines in a power network. As briefly mentioned in the section 4.1, the major change from TO was the state space converted from binary to discrete variables. The second big change was the size of the state space, which depends on the number of PSTs in a power network and on the selected range of tap positions. In our case, the range of the tap positions was set from -25 to 25 with the step of 1 (in total 51 tap positions). The total number of all PSTs configuration creating the whole state space is given by the following equation:

$$n_{TP}^{n_{PST}} = S_s, \tag{4.3}$$

where S_s is the state space, n_{TP} is the number of tap position options and n_{PST} is the amount of PSTs. In our case the equation was equal to:

$$51^3 = 132651 \tag{4.4}$$

Due to the smaller state space, every RAM value of PST configuration computed and shown in the figure 4.3. As seen, The RAM values create non linear layers depending on the configuration of PSTs, where the dominant Phase Shifter is the PST_3. The global maximum was found for the configuration $PST_1 = 22$ and PST_2, $PST_3 = -25$ represented in the figure 4.3 by a red cross. The worst values were found in the layer containing the $PST_3 = -7$. The another important layer is the one for $PST_3 = 10$ representing the local maximum causing big issues for DFS algorithm detailed in the following section 4.2.2.

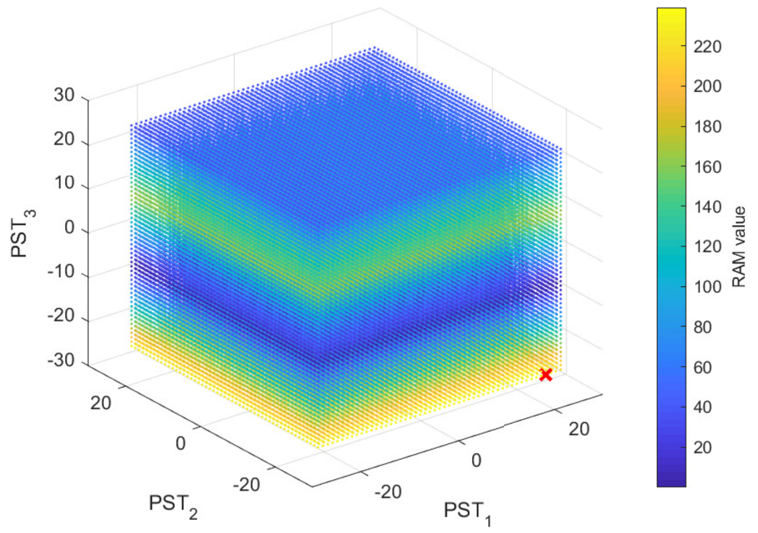


Figure 4.3: All RAM values representing the whole state space

In following sections, the computation performance of particular algorithms is analyzed.

4.2.2 DFS

The DFS algorithm was selected in the case of PST optimization as well. The algorithm consists of same steps as in the Topology and started with the configuration $PST_1 = 0, PST_2 = 0, PST_3 = 0$ and from every depth the configuration with the highest RAM was selected. If the RAM value was the same as in the previous depth, the DFS moved 0 to 3 steps, which was given by random chance.

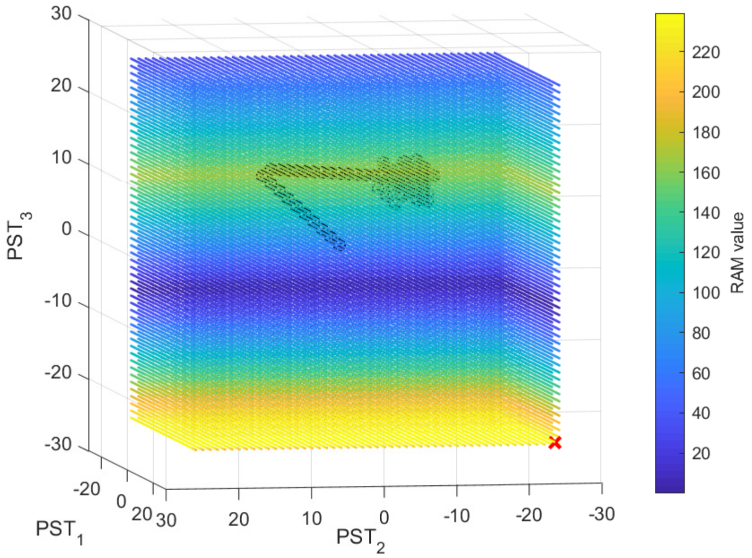


Figure 4.4: DFS algorithm

The figure 4.4 shows 100 iterations of the DFS algorithm, which does not converge to the global maximum. Because of not suitable initial conditions, the algorithm stops in the area of the local maximum. Therefore, the DFS does not meet the requirement of finding a global maximum shown with a red cross. Therefore the DFS is only applicable in the near area of the global optimum, otherwise there is a high probability of reach a local maximum.

4.2.3 PSO

Similar to Topology optimization, three most influencing parameters were analyzed. The range of the analyzed parameters was changed to fit the PST Optimization, where the state space is much smaller and the variables are discrete. The tested values of PSO parameters mentioned in the section 3.3.3 were:

- N - size of a swarm: [10 20]
- c_1 - social acceleration coefficient: [0.1 0.4 0.7 1 1.5 2 5 10 50 100]
- c_2 - cognitive acceleration coefficient: [0.1 0.4 0.7 1 1.5 2 5 10 50 100]

The figures 4.5, 4.6, 4.7 and 4.8 represents tables each containing 100 values that specify the influence of the selected parameters. In the left tables, the average number of iterations are displayed, where the better results are expected for the bigger swarm N = 20. The very important sign is shown by all tables, which is the clear dominance of the parameter c_2 representing the aggressive behaviour throughout the whole state space. The extreme values on both sides of c_2 represented a bad configuration represented, where the absolute worst configuration, where the algorithm with parameters $c_1 = 0.1$, $c_2 = 100$ and $N = 10$ did not ever converge to the global maximum by any of the 20 parallel calculations.

Number of particles N = 10

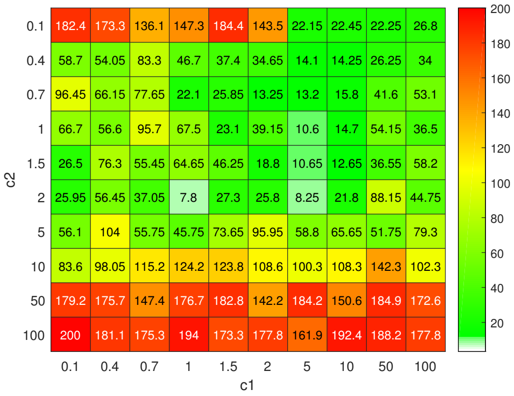


Figure 4.5: Average number of iterations

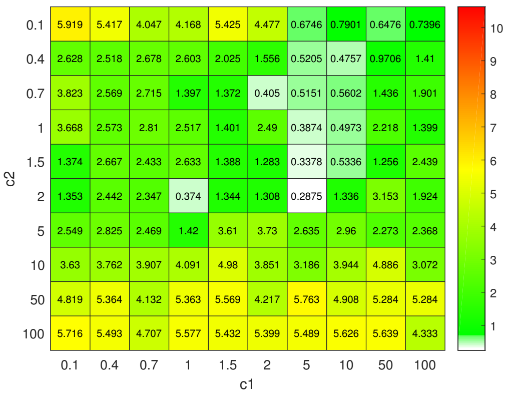


Figure 4.6: Average computation time

In the case of the dominant parameter $c_2 = 50$, the behaviour of swarm particles was too aggressive, which resulted in an immediate move to the extreme limits of the state space, where the swarm stayed throughout the all iterations. The opposite extreme values of $c_2 = 0.1$ are very ineffective as well, which is caused by a very calm behaviour meaning a very small improve. The number of iterations represented in the tables on the left side is not an accurate indicator due to a different size population and therefore the particular configurations were converted to values over time.

Number of particles N = 20

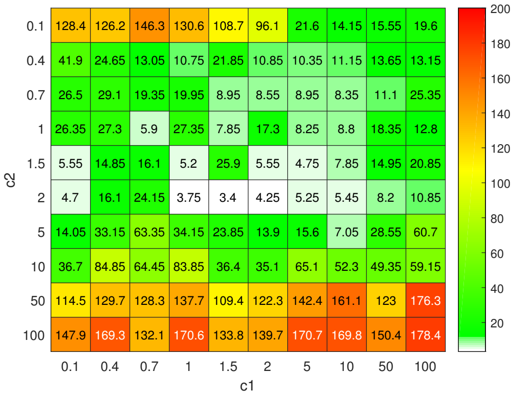


Figure 4.7: Average number of iterations

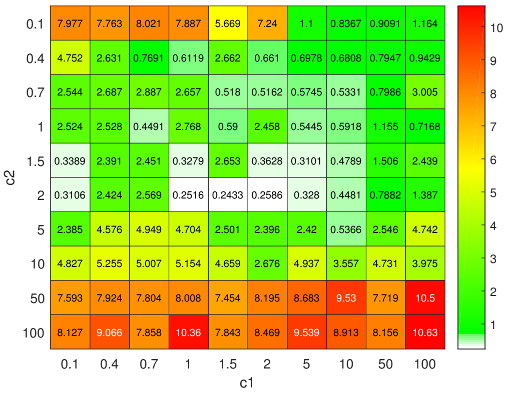


Figure 4.8: Average computation time

Performance comparison of various configurations

Average time comparison

In this section, the computation performance of the PSO algorithm is assessed for various configurations. The figure 4.9 displays the best 20 configurations sorted by average time. The results show that the best RAM values were achieved with the parameters $c_2 = 2$, which provides the ideal global aggressiveness of the swarm. The second best parameter c_2 equal to 1.5 forming together with the best $c_2 = 2$ dominant values guaranteeing the best configurations, which is recorded at the best 12 ones in a row. Another very key parameter is the size of a swarm N, where 9 of the 11 best solutions were for size N = 20. The least essential parameter is the c_1, where the range of values was the largest. In general, it can be said that the best setting of c_1 is in the range of 0.1 to 5. The absolute best solution was with the configuration $N = 20$, $c_1 = 1.5$ and $c_2 = 2$. For a better overview, in the figure 4.10, the values are displayed in percent.

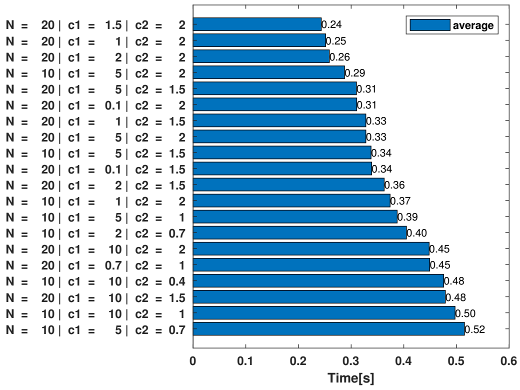


Figure 4.9: The best 20 configurations sorted by time (seconds)

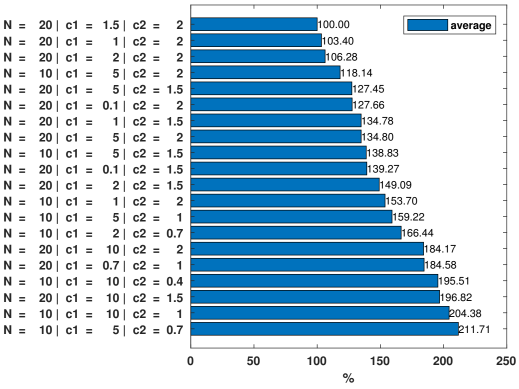


Figure 4.10: The best 20 configurations sorted by time (percent)

Boundary computation time comparison

In this part, boundary computation time are analyzed, whose 20 configuration with best performing computation time are depicted in the next figure 4.11. In this bar chart, the boundary values (minimum and maximum) are displayed. The MIN value represents the fastest solution recorded form the 20 parallel loops. The MAX values stands for the solution with the worst time. The attention should be focused specially on the MAX values. Based on the figure, the worst time (longer than 1.5s) appears once for the configurations with N equal to 20 and once for the configurations N = 10. Therefore, it cannot be said, as at a Topology optimization, that the size of the swarm N affects the maximum time.

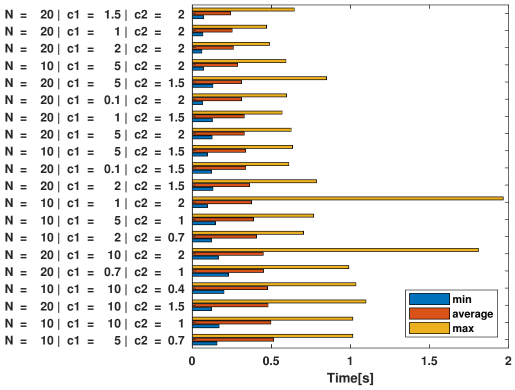


Figure 4.11: The best 20 configurations with boundary values (seconds)

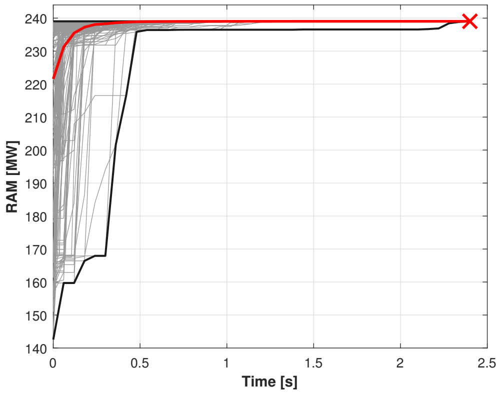


Figure 4.12: Objective function evolution for all realizations

PSO Realizations

On the last figure 4.12, objective function evolutions are displayed for all 20 realizations under consideration of the best 20 configurations. There are depicted all 20 parallel loops for each configuration (in total 400 graphs). These 400 executions are grey colored. The black colored edges represent the boundary values of all loops, while the red colored line shows the progress over time of the average value. The global maximum (max RAM) is represented by the red cross at the end. According to the average value and individual executions, it is obvious that except for some extremes, most of the runs were very fast ending approximately in 0.7s. The zone of the most runs is very narrow as well.

4.2.4 Genetic algorithms - Roulette wheel selection (GA - RW)

As briefly mentioned in the section 4.1.1, the Genetic algorithm using the Roulette wheel selection had to be changed due to the 3 main reasons:

1. Different variables in a state space
   The PST optimization works with in a state space containing discrete variables, while the topology uses the binary ones.

2. Smaller state space (population) size
   The PST optimization searches a state space containing 132651 states, which is a huge difference from the topological one, where the number of states was equal to 2^{186}.

3. Smaller size of individuals with in a population
   The PST optimization uses an individual of length equal to number of Phase Shifting Transformers(3 in our case), which is very different from the length 186 used for Topology.

The major changes are:

1. Crossover: All selected best parents were combined with each other(individual Phase Shifters separately). The conventional Crossovers (i.e One Point Crossover, Multi Point Crossover) provide very bad results due to the small size of an individual in a population. The Crossover was the same for all tested runs.

2. Mutation: The aggressivity was increased by higher random change(positive or negative direction) and range mutation

Number of best parents n_{BP} and the size of a population N were considered for the testing. The overall amount of configurations was reduced to 25 due to higher computation requirements and considered configuration can be summarized as:

- N - size of a population: [30 40 50 60 100]
- n_{BP} - number of Best parents: [1 2 3 5 10]

The computation performance of the GA - RW algorithm is assessed for various

configurations. The range of the parameters was set to best describe the quality range of this algorithm. The figures 4.13 and 4.14 display the best 20 configurations sorted by average time represented in two tables side by side. The left one shows the average number of iterations, while in the right one, the average time is shown. As seen from the figure 4.13, the higher number of Population, the lower number of iterations. Dependence of the n_{BP} is seen clearer from the right table. The worst values were represented by the extreme values of n_{BP}. In the case of $n_{BP} = 1$ focuses the algorithm only on 1 parent, which prevents diversity of solutions and only the way of the best parent is searched. On the other hand, the computed RAM values for configurations, where the n_{BP} includes more that 20% of the number of Population, were very week as well. The absolute worst average time equal to 6.366s had the configuration $N = 30$, $n_{BP} = 10$, where every third individual was the Best Parent. The ideal number of best parents was the number 3, where even the absolute best average time having the population equal to 50 was found.

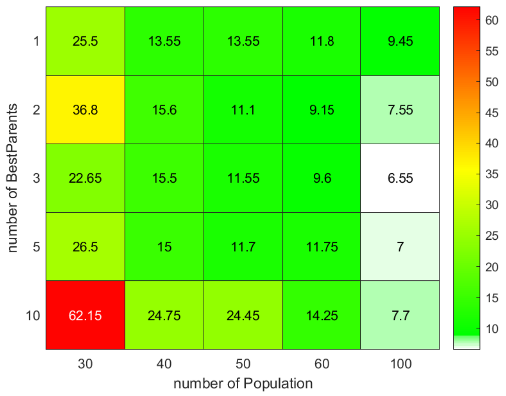


Figure 4.13: Average number of iterations

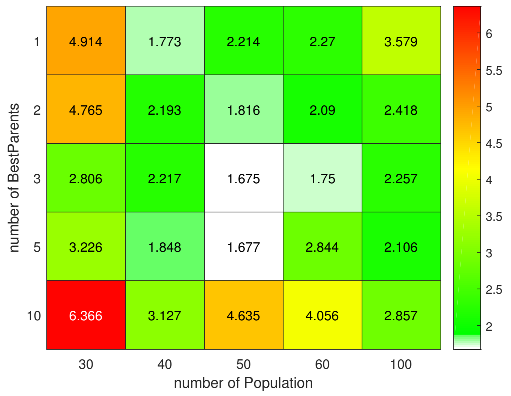


Figure 4.14: Average computation time

Performance comparison of various configurations

Average time comparison

Similar to PSO algorithm, the next charts in the

figures 4.15 and 4.16 shows all 25 average time values sorted from best to worst. It can be seen from the graphs that the top 6 configurations have reached a very similar time, where the exact percentage difference is displayed in the right chart. The size of a population ranged between 40-60 and the best 2 configurations had the size equal to 50. The quality of n_{BP} varied depending on the number of population.

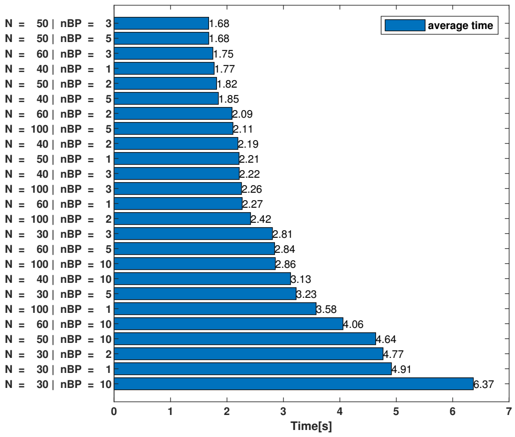


Figure 4.15: All 25 configurations sorted by time (seconds)

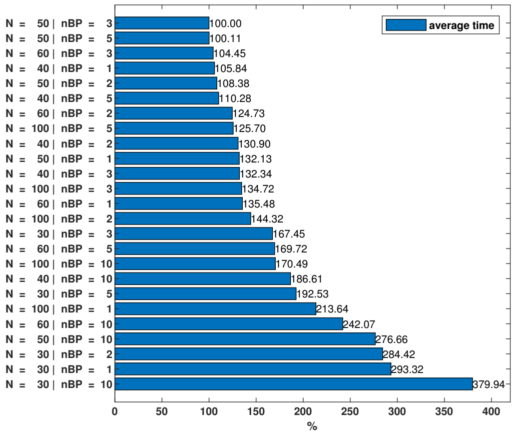


Figure 4.16: All 25 configurations sorted by time (percent)

Boundary computation time comparison

The figure 4.17 shows the bar chart sorted by the average time as in the previous figures 4.15 and 4.16. For detailed information, there are additionally displayed the MAX and MIN values. The MIN value represents the fastest solution recorded form the 20 parallel loops. The MAX values stands for the solution with the worst time. The attention should be focused specially on the MAX values. Based on the figure, the worst time (longer than 10s) appears always for the configurations either containing the lowest size of a population equal to 30 and/or the $n_{BP} = 10$ (highest value). For all other cases, the MAX time was in the narrow range approximately between 4 and 8 seconds.

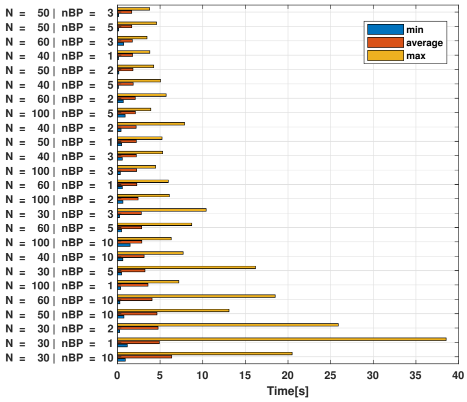


Figure 4.17: All 25 configurations with boundary values (seconds)

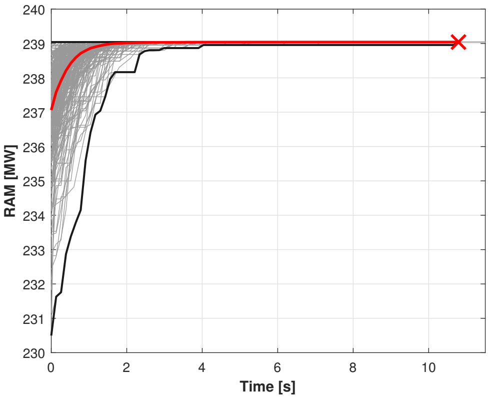


Figure 4.18: Objective function evolution for all realizations

GA - RW Realizations
On the last figure 4.18, objective function evolutions are displayed for all 25 realizations. There are depicted all 20 parallel loops for each configuration (in total 500 graphs). These 500 executions are grey colored. The black colored edges represent the boundary values of all loops, while the red colored line shows the progress over time of the average value. The global maximum (max RAM) is represented by the red cross at the end. According to the average value and individual executions, it is obvious that except for some extremes, most of the runs were very fast ending approximately in 2.2s. The zone of the most runs is very narrow as well.

4.2.5 Results assessment

In this section, the results for best configurations of selected methods will be compared. The best found configurations are:

- DFS: Not finding a global maximum
- PSO: N = 20, c1 = 1.5, c2 = 2
- GA(RW + BP): N = 50, nBP = 3

For these best configuration of EAs, twenty realization were executed and the results were stored for further assessment, which is graphically shown on figures 4.19 and 4.20. For the assessment, following metrics were taken into account:

- best possible time (dashed thin line ending in a circle)
- average time (thick line ending in a cross)
- worst possible time (dashed thin line ending in a cross)

The comparison of GA + RW and PSO algorithm is shown in the figure 4.19 and 4.20, where the same colors as in Topology optimization were used (PSO - green, GA RW - blue). The DFS could not be compared due to failure to meet the required RAM value. From the figure 4.19 it can be seen that the average value absolutely depends on the worst result computed from 20 parallel runs, which is shown on the values in the table 4.1. The detailed run with the worst time of GA + RW can be seen in the figure 4.20, where the value stops at the 12th iteration with the RAM of 238.9561 representing 99.96% of the required value 239.0419 (global maximum). The PSO algorithm has significantly better performance. The PSO average time of 1 iteration was only 0.0716s, while for the GA (RW) + BP algorithm was 0.1451s, which is more than 2x faster. In both cases, the global maximum was found after the first iteration., where the resulting time is equal to the average value of 1 iteration. Even the average number of iterations was almost 3x lower at the PSO algorithm (9 iter) compared to GA (RW) + BP (26 iter). The average amount of iteration. Another big difference was the range of all parallel computations. All PSO runs were in range from 0.0716 to 0.5720 seconds, while the GA (RW) + BP values were in range from 0.1459 to 3.6265 seconds, which almost 7x bigger. The PSO algorithm achieves better results in all aspects and therefore the PSO is a better optimization algorithm for Phase Shifting Transformers as well.

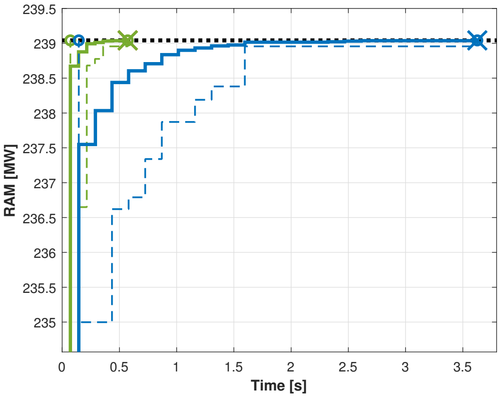


Figure 4.19: Comparison of PSO and GA (RW) + BP

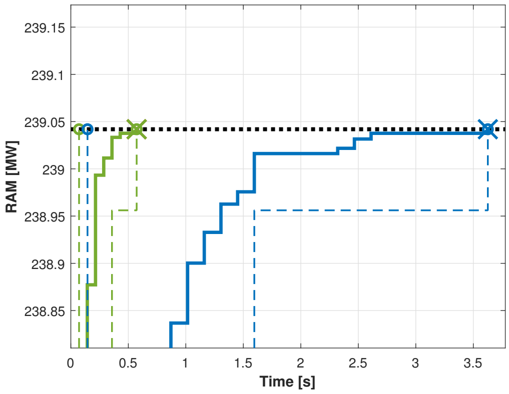


Figure 4.20: Detailed comparison of PSO and GA (RW) + BP

The exact results are presented in table 4.1 for a better general overview.

	PSO	**GA(RW) + BP**
average	**0.5725s (9 iter)**	**3.7726s (26 iter)**
best	0.0716s (1 iter)	0.1451s (1 iter)
worst	0.5725s (9 iter)	3.7726s (26 iter)

Table 4.1: Comparison of PSO and GA(RW) + BP by time

Chapter 5

Sensitivity analysis of distribution factors

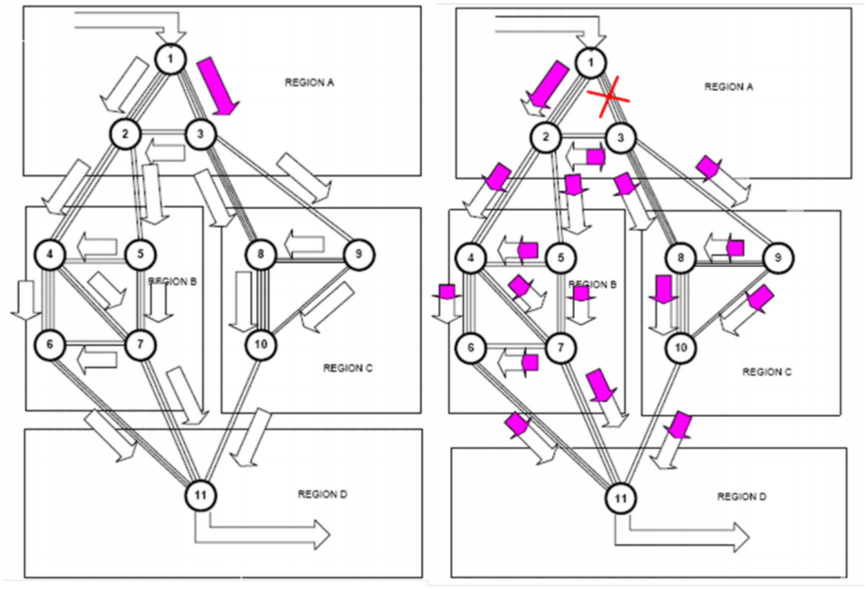
5.1 Overview of power system security assessment

Power lines are crucial elements in power network, where some unpredictable failures (e.g. lightening or system faults) can occur. The consequence may be tripped a power line, which may cause a huge interruption to the whole power system and causes damage. In this case, the power flow from the tripped line is redistributed to the other lines in a power network, which may results in overloading of other power lines. In the worst cases, a cascading failure and system collapse can happen. A single line outage do not endanger all other lines in the system but only some specific ones. Certain lines can be very affected, while some other lines can not. Therefore the TSOs have to know the line impact distribution. Since the power network usually consists of a huge number of lines, the problem of studying thousands of possible outages becomes very difficult to solve if it is desired to present the results quickly [2]. For the security assessment of power lines outages , so called contingency analysis is calculated, which consists of security assessment of the post-contingency state of the power network. The assessment relies on the multiple calculation of load flow problem, which can be time demanding for large power networks. In some application, therefore a simplified contingency analysis based on an approximate solution is performed. In this case, a post contingency state is evaluated as a linear projection of the state of linearized power network model (DC power network model), line outage distribution factors (LODF) and nominal power flows [42].

5.1.1 Line Outage Distribution Factor

LODF is proposed to estimate the steady-state active power flow redistribution after a line outage. The major advantage is the high calculation speed. On the other hand, the post contingency state includes an approximation error, because of linearized estimation. LODF is a percentage value which indicates the proportion between the change of active power flow on one line and the active power flow on the outage line before it tripped [18].

The power flow redistribution after a line outage is shown in the figure 5.1, which has been taken from [36].



9 (a) (b)

Figure 5.1: (a) Power flow before line 1-3 outage (b) Power flow after line 1-3 outage

White arrows indicate the flow of power before the outage, while purple arrows in (b) show the variation of power flow on the rest of system due to the outage of line 1-3. If line k is tripped, the active power flow on l is

$$f_l = f_l^0 + LODF_{l,k} \cdot f_k^0, \tag{5.1}$$

where f_k^0 is an active power flow in line k before line l outage, f_l^0 is an active power flow on line l before l outage and $LODF_{l,k}$ is a line outage distribution factor monitoring line l after an outage on line k. [42]

5.2 Methodology

As mentioned earlier, the LODF provides only estimated values describing the power network state prediction. The major goal of this chapter is to analyze the quality of LODF results compared to real Alternating Current (AC) flow, which calculates accurate power network state based on full nonlinear power network model. The methodology consists of 3 main parts (Inputs, Calculation, Outputs) detailed in the relevant subsections. For a better overview, these parts performed in the methodology are shown in the figure 5.2 as well.

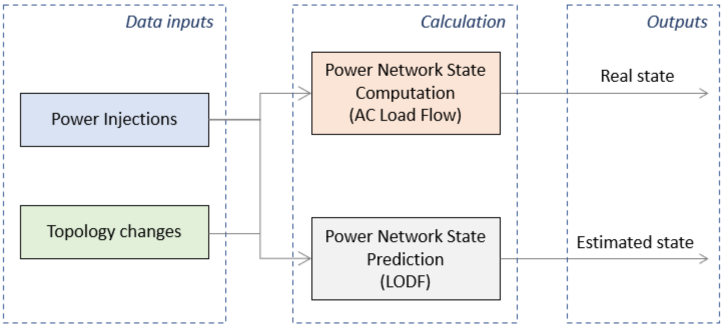


Figure 5.2: Sensitivity analysis of LODF

5.2.1 Inputs

In this part, the Topology and Power injection were considered as inputs.

- Topology - includes the basic power network topology and contingency cases
- Power injections - represents active and reactive power inputs producing by power injection units (i.e. generation, consumption)
- Voltage magnitude setpoints - nodal voltage magnitude setpoints on selected buses (PV buses)

5.2.2 Calculation

It is a main part, where the power network with current parameters changed by inputs were computed.

- Power network state computation - calculates actual power network state based on AC load flow
- Power network state prediction - calculates prediction of power network state based on current state and LODF application

5.2.3 Outputs

The final part, where the data describing the real and estimated state of power network were obtained.

- Real state - actual power network state based on AC load flow
- Estimated state - prediction of power network state based on LODF

The real and estimated state data were then analysed and compared detailed in the section 5.3.

5.3 Analysis

5.3.1 Design of experiments

This section describes the application of previously mentioned parts of methodology. First, the software and hardware environment and the test case are specified. In the second part, the results of LODF method are evaluated. At the end, the method for improving the results is proposed.

Software environment

Similarly to evolutionary algorithms, the MATLAB R2018a programming language was used. The main reason was the open source programming package called MATPOWER detailed in the section 3.4.1. Due to the smaller power network represented by a test case detailed below and therefore the smaller computational requirements was the Parallel Computing Toolbox not run.

Hardware environment

All computations ran on the same computer detailed in the section 3.4.1.

Test Case

For the sake of clarity, the small power network represented by a 30-bus case, based on IEEE 30-bus case [30] was selected. It is very suitable for the simulations due to the size. The test case contains these parameters:

- Number of buses: 30
- Number of generators: 6
- Number of all branches: 41

Application

- Topology input
  This part was similar to the Topology optimization in the section 3. The binary parameter - branch status (1 - in service, 0 - out of service) was gradually changed (only one branch was switched off to simulate a network outage at a time).
- Power injection input
  It simulates the actual loading of the power network, which affects volume of power injections in two parts:

- Consumption - real and reactive power demand is multiplied by the selected loading factor
- Generation - real and reactive power output is multiplied by the selected loading factor

The following set of loading factor was considered: : [0.1 0.4 0.7 1 1.3 1.6]

- Power network real state computation
  The calculations were performed with with specific input parameters by calling a $MATPOWER$ $runpf$ function, which returns the exact power flow on branches (i.e. power lines, transformers).
- Power network state prediction
  The LODF matrix was calculated, where two MATPOWER functions were applied - $makePTDF$ and $makeLODF$. The $makePTDF$ function returns the PTDF matrix for a given choice of slack. Based on the calculated PTDF matrix the $makeLODF$ function forms the LODF matrix. LODF matrix is $n_{br} \times n_{br}$, where n_{br} is the number of branches [30]. In our case the LODF matrix is 41×41.
- Output results
  The comparison of real and predicted power network state is done by subtraction the power flows on specific branches. The error of the predicted state compared to the real one is further analyzed.

5.3.2 Error analysis

In this section, the precision of LODF estimation in comparison to actual power system state was analyze. The analysis is based on the estimaiton error analysis, which consists in the difference between the current and estimated power flow on selected power lines. In the analysis, mainly a visual and statistic analyses were conducted.

For the calculation, the following notation can be introduced:

- Absolute error:
  $$\Delta_{pl} = P_{real,pl} - P_{est,pl} \tag{5.2}$$
- Relative error:
  $$\delta_{pl} = \frac{P_{real} - P_{est}}{P_{real}}, \tag{5.3}$$

where P_{real} is the real active power and P_{est} is the estimated active power for power line pl.

Let us assume that the absolute and relative errors are depended on the power system loading, then the error equations can be rewritten as

- Absolute error:
$$\Delta_{pl}(l) = P_{real,pl}(l) - P_{est,pl}(l) \tag{5.4}$$

- Relative error:
$$\delta_{pl}(l) = \frac{P_{real}(l) - P_{est}(l)}{P_{real}(l)}, \tag{5.5}$$

where l stands for a power system loading in terms of multiplication of the values of nominal state power injections.

At fist , dependencies of absolute and relative error depending on the given system loading were analyzed. This is depicted on following histograms, where the error is marked on the x-axis and the frequency on the y-axis. In the figure 5.3, the absolute error was considered, while the figure 5.4 represents a relative error.

As seen from the figure 5.3, the error distribution depends on the power network load. As the load increases, the frequency of values near zero decreases and the variance of error increases. The total number of values for each load was 1681. The relative error shown in the figure 5.4 is very similar for each load.

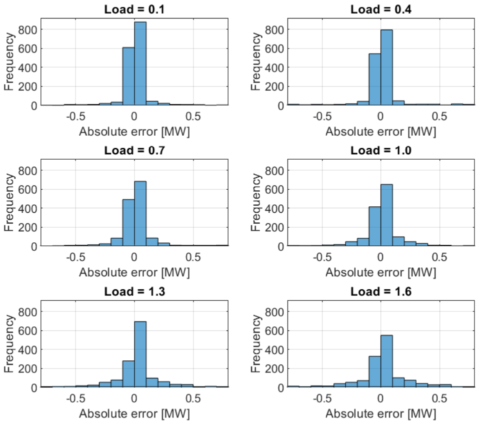


Figure 5.3: Absolute error distribution

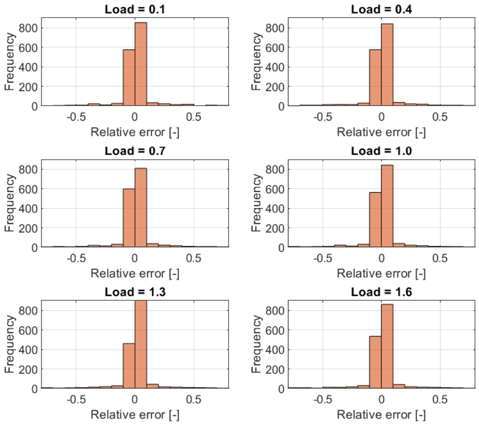


Figure 5.4: Relative error distribution

For the histograms, however, the correlations between absolute and relative errors are not fully clear. Therefore, special scatter plot were designed in order to link several aspects of errors. In the following figure 5.5, the individual percentage errors[1] (y-axis) are displayed for whole set of power system loading (x-axis). The size of circles as well as their color represents absolute errors. The percentage error of most network configurations is between 0 and -200% (0 and -2 in relative error confirmed by the histogram in the figure 5.4). The maximum absolute error equal to 81.9 MW was measured on line 40 at outage of line 10, which is highlighted by the red circles. From the figure can be seen, the absolute error is rising with increasing power

[1] Relative errors multiplied by 100.

system loading. Further, the high relative error are not always necessary caused by relative small actual power flows. Further, the errors are mostly of monotonic nature and can be approximed by a suitable polynom as discussed in the following text.

The next figure 5.6 shows the dependency of the line outages to absolute errors and links they with the power lines. On the figure, detailed analysis of the absolute error for only one load. As nominal loading, the largest loading values 1.6 was selected, where the error is most significant. As seen, the the most frequently line with high absolute error over most of contingencies is the line number 15, which can be considered as a bottleneck.

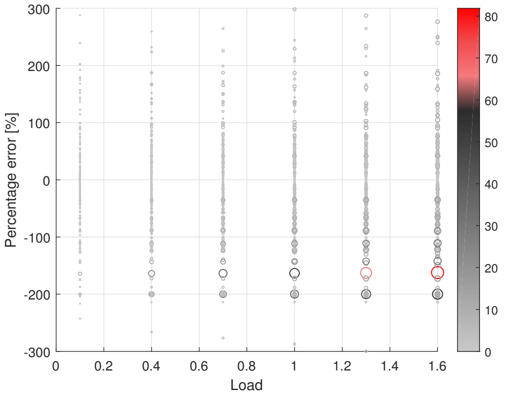


Figure 5.5: Dependence of absolute and percentage error on load

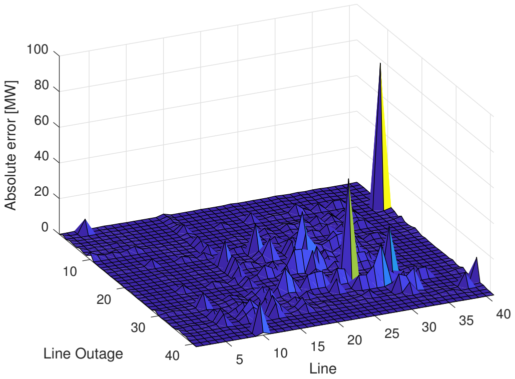


Figure 5.6: Load of individual lines after an outage (load 1.6)

The bottleneck power line is further visually detailed in the figure 5.7, where the absolute error depending on the load is displayed. There are 41 charts, each representing the absolute error for each line outage. As expected, individual values increase based on a larger load.

The polynomial behaviour of the absolute error can be utilized for the reduction of the estimation error for selected branches[2]. This can be achieved by a polynomial approximation of the error shown in the example of bottleneck in the figure 5.8. For the approximation, the *polyfit* MATLAB function of degree $n = 2$ was applied, from which the correction factor (CF) was calculated. Finally, the CF represented by 41 × 41 matrix was added to the LODF estimation. For every power line, the equation 5.1 of active power flow f_{pl} can be defined as follows:

$$f_{pl} = f_{pl}^0 + LODF_{pl,k} \cdot f_k^0 + CF_{pl,k}(l), \tag{5.6}$$

where f_{pl}^0 is the nominal active power flow over powerline pl, k represents $k-$th contingency (outage) and $CF_{pl,k}(l)$ is the polynomial correction factor for pl-th power line, k-th outage and system loading l.

[2] The bottlenect power line in our case

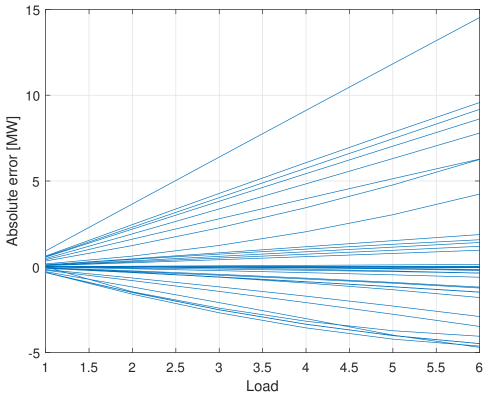


Figure 5.7: Absolute error on a bottleneck

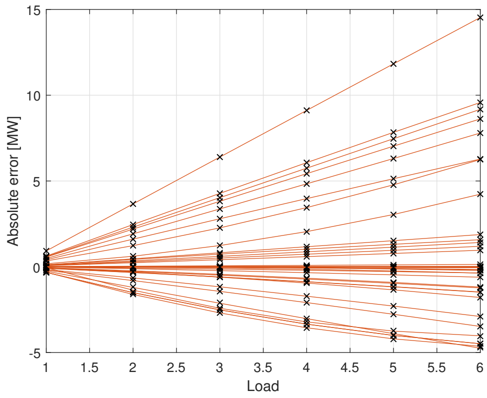


Figure 5.8: Polynomial error approximation

Histograms in the figure 5.9 and 5.10 were again used to compare the final absolute and relative error before and after the application of the correction factor. After application, there has been a significant improvement, which can be seen on the x-axis, where the absolute or relative error is displayed. The absolute error shown in the figure 5.9 is now mostly between -5 and 5 kW, while before the application, the error was in range of -500 and 500 kW. The relative error compared to the absolute error is even smaller. Of interest is the relative error at the lowest network load of 0.1, where the variance is the largest. This is probably due to a bad approximation of degree 2 for the very low irregular values.

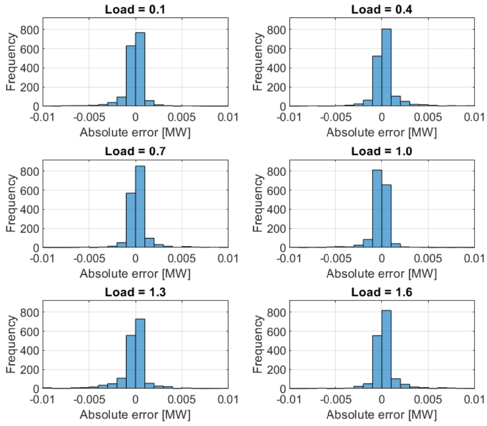


Figure 5.9: Distribution of reduced absolute error

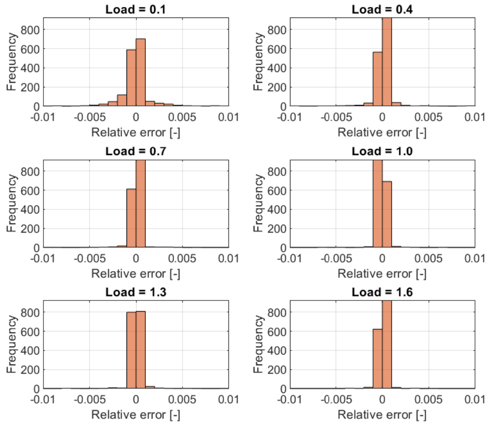


Figure 5.10: Distribution of reduced relative error

Chapter 6

Conclusion

Interconnection of individual European markets called a Market Coupling is a topical issue. Its importance grows along with the growing demand for efficient use of the power transfer network system between European countries. Market Coupling has a long history, as the beginnings are determined by the liberalization of the energy sector. Nowadays, transmission system operators (TSOs) work with complex algorithms calculating the transmission capacity between two neighbouring markets. Most European countries, including the Czech Republic, use a traditional approach called ATC for the capacity calculation. On the other hand, in the Benelux countries, a new approach called Flow-based (FB) is applied. The FB method works essentially on a similar principle as the ATC, but it contains a higher amount of parameters representing the actual state of the power network [20]. One of the main advantages over the ATC approach are that the FB uses critical network elements, especially critical branches (CBs). CBs are transfer lines, cables, or devices that can be significantly affected by cross-border flows. Therefore, they have to be monitored, and the most important value called Remaining Available Margin (RAM) must be calculated and controlled on these selected CBs. RAM represents the remaining capacity that can be given to the market taking into account the already allocated capacity [20]. RAM value needs to be optimized for the most efficient use of the transfer network, which is the main goal of this diploma thesis. The first chapter is devoted to these mentioned issues and explanation of the most important ATC and Flow-based method parameters. At the end of this chapter, the remedial actions (RAs) for RAM optimization are presented, where the Topological RAs, Phase Shifting Transformers RAs and Generation Redispatch RAs were considered.

In the second chapter, the state of the art in RAM optimization was analyzed. Because FB is a relatively new approach, the commonly used methods for Optimal Power Flow (OPF) issues were researched. Many different algorithms for particular RAs were analyzed, and then the two best rated evolutionary algorithms were selected: Particles Swarm Optimization (PSO) and Genetic Algorithms (GA) - roulette wheel selection. Both algorithms are suitable for Topological and Phase Shifting Transformers RAs. To evaluate the quality of these algorithms, conventional methods of decision tree search - Breadth-First Search (BFS) and Depth First Search

(DFS) were considered.

In the third chapter, Topology optimization is detailed. The Topological RA consisted of opening or closing of one or more lines in a power network. All possible topology combinations formed a binary state-space of the size equal to 2^{186}, where 186 is the number of changeable lines from a simulated Test Case and 2 represents two states of the branch status (1 - in service, 0 - out of service). First, an analysis of conventional methods was performed. Subsequently selected evolutionary algorithms are described. In the next part, the previously mentioned algorithms were implemented in the MATLAB programming language. In conclusion, the individual algorithms were compared with different parameters. For optimizing RAM on CB using Topological RA, PSO algorithm with the average time equal to 1.57 seconds achieved the best results. The average time 5.54 seconds of the algorithm GA - the roulette wheel finished in the second place. The DFS algorithm was with the average time of 7.24 seconds the worst one. The BFS algorithm could not be used due to the high time and huge memory requirements.

The fourth chapter describes the optimization using PSTs, which can redistribute the power flow between two parallel lines. For PSTs, the tap position representing the given angle was changed. The ranged was between -25 and 25 and the number of PSTs in the power network was 3. The RAM calculations on CB were simulated on the same Test Case as the topology optimization. All combinations of these parameters formed a discrete state space of the size 51^3, where the 51 is the number of tap positions and 3 is the amount of PSTs. Slightly modified algorithms used for the topology were applied for PST RAM optimization as well. The comparison of particular methods turned out very similarly to topology optimization. The PSO algorithm with the average time of 0.5725 seconds was the best. The GA - the roulette wheel finished in second place with the average time of 3.7726 seconds. The DFS algorithm did not find the required RAM value due to a local maximum in state space.

In the last chapter, the sensitivity analysis of distribution factors was performed due to the security assessment of power line outages. The assessment relies on the multiple calculations of load flow problem, which can be time demanding for large power networks. In some application, therefore a simplified contingency analysis based on an approximate solution is performed. In this case, a post contingency state is evaluated as a linear projection of the state of linearized power network model (DC power network model), line outage distribution factors (LODF) and nominal power flow [42]. LODF is proposed to estimate the steady-state active power flow redistribution after a line outage. The major advantage is the high calculation speed. On the other hand, the post contingency state includes an approximation error, because of linearized estimation. The individual calculations were simulated on a smaller 30-bus case, based on the IEEE 30-bus case. For the implementation, the Matlab programming language was used, where the open-source programming package called MATPOWER was used. On the test case, the network topology was changed (only one branch was switched off to simulate a network outage at a time). The second parameter was the power injection input simulating the increased real and reactive power flow in a network. At the end of this chapter, the absolute, relative and percentage error obtained after subtracting estimated power from the real power was analyzed. First, the absolute active error was evaluated using a histogram. Subsequently, the most frequently loaded line called a bottleneck was selected from the network, where the absolute active power was polynomial approximated. Finally, this value was added to the estimated value, which significantly reduced the error.

The diploma thesis showed that evolutionary algorithms are very suitable for solving selected aspects of the RAM optimization. Especially, the PSO algorithm used at Topology and PST optimization achieved excellent results. The LODF provides very good estimates in a short time as well. All set goals of this diploma thesis were achieved.

Bibliography

[1] Explanatory note on the day-ahead common capacity calculation methodology for core ccr, 2017. https://consultations.entsoe.eu.

[2] Rashid H AL-Rubayi, Afaneen A Abood, and Mohammed R Saeed Alhendawi. Simulation of line outage distribution factors (lodf) calculation for n-buses system. *International Journal of Computer Applications*, 156(3), 2016.

[3] Belgium Belpex, SA POWERNEXT, et al. Trilateral market coupling. algorithm appendix. 2006.

[4] Sergio Ulgiati B.Sudhakara Reddy. *Energy Security and Development*. Springer, 2015.

[5] Po-Hung Chen and Hong-Chan Chang. Large-scale economic dispatch by genetic algorithm. *IEEE transactions on power systems*, 10(4):1919–1926, 1995.

[6] Educative. What is dynamic programming? https://www.educative.io/courses/grokking-dynamic-programming-patterns-for-coding-interviews/m2G1pAq0OO0.

[7] ENTSO-E. Explanatory document to the proposal for the coordinated redispatching and countertrading methodology for capacity calculation region core in accordance with article 35 of the commission regulation (eu) 2015/1222 of 24 july 2015 establishing a guideline on capacity allocation and congestion management, sep 2018. https://eepublicdownloads.blob.core.windows.net/public-cdn-container/clean-documents/Network

[8] ENTSO-E. Explanatory document to all tsos' proposal for the methodology and assumptions that are to be used in the bidding zone review process and for the alternative bidding zone configurations to be considered in accordance with article 14(5) of regulation (eu) 2019/943 of the european parliament and of the council of 5th june 2019 on the internal market for electricity, oct 2019. www.entsoe.eu/news/2019/10/07/bidding-zone-review-methodology-assumptions-and-configurations-submitted-to-nras.

[9] Stephen Frank, Ingrida Steponavice, and Steffen Rebennack. Optimal power flow: a bibliographic survey i. *Energy Systems*, 3(3):221–258, 2012.

[10] Mitsuo Gen, Runwei Cheng, and Lin Lin. *Network models and optimization: Multiobjective genetic algorithm approach*. Springer Science & Business Media, 2008.

[11] Helena Gerard, Enrique Israel Rivero Puente, and Daan Six. Coordination between transmission and distribution system operators in the electricity sector: A conceptual framework. *Utilities Policy*, 50:40–48, 2018.

[12] HackerEarth. Breadth first search tutorials and notes, sep 2018. https://www.hackerearth.com/de/practice/algorithms/graphs/breadth-first-search/tutorial/.

[13] HackerEarth. Depth first search tutorials and notes, sep 2018. https://www.hackerearth.com/de/practice/algorithms/graphs/depth-first-search/tutorial/.

[14] JohnH Holland. adaptation in natural and artificial systems, university of michigan press, ann arbor,". *Cité page*, 100, 1975.

[15] Dalton John. Genetic algorithms. http://www.edc.ncl.ac.uk/highlight/rhjanuary2007.php.

[16] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.

[17] H Kocot, R Korab, W Lubicki, M Przygrodzki, G Tomasik, and K Żmuda. Improving the cross-border transmission capacity of polish power system by using phase shifting transformers. *Proc. 45th Cigre Session*, 1, 2013.

[18] R Korab and R Owczarek. Impact of phase shifting transformers on cross-border power flows in the central and eastern europe region. *Bulletin of the Polish Academy of Sciences. Technical Sciences*, 64(1), 2016.

[19] Tomáš Kubišta. Market coupling-propojování evropských energetických trhů, oct 2015. https://oenergetice.cz/trh-s-elektrinou/market-coupling-propojovani-evropskych-energetickych-trhu.

[20] Tomáš Kubišta. Flow-based market coupling na evropských trzích, july 2016. https://oenergetice.cz/evropska-unie/flow-based-market-coupling-na-evropskych-trzich/.

[21] Atul Kumar. Genetic algorithms. https://www.geeksforgeeks.org/genetic-algorithms/.

[22] J. Lago. Clearing price determination, 2017. https://jesuslago.com/introduction-to-the-electrical-markets.

[23] Lu Minh Le, Hai-Bang Ly, Binh Thai Pham, Vuong Minh Le, Tuan Anh Pham, Duy-Hung Nguyen, Xuan-Tuan Tran, and Tien-Thinh Le. Hybrid artificial intelligence approaches for predicting buckling damage of steel columns under axial compression. *Materials*, 12(10):1670, 2019.

[24] KU Leuven. Cross-border electricity trading: towards flow-based market coupling. *EIFACT SHEET*, 2, 2015.

[25] J Machowski, JW Bialek, and JR Bumby. Power system dynamics stability and control, john wiley&sons, 2008.

[26] MathWorks. What is parallel computing? https://www.mathworks.com/help/parallel-computing/what-is-parallel-computing.html.

[27] Mishadoff.com. Dfs on binary tree array, sep 2018. http://mishadoff.com/blog/dfs-on-binary-tree-array/.

[28] M. Rahli M.Younes and L. Abdelhakeem-Koridak. Optimal power based on hybrid genetic algorithm. *Journal of Information Science and Engineering*, 23(4):1801–1816, 2007.

[29] J. Praveen and B. Srinivasa Rao. Single objective optimization using pso with interline power flow controller. *International Electrical Engineering Journal*, 5(11):1659–1664, 2014.

[30] Carlos E. Murillo-Sánchez Ray D. Zimmerman. User's manual, 2019. https://matpower.org/docs/manual.pdf.

[31] Bc Jiří Salavec. Congestion income distribution under flow-based method, 2016.

[32] M Saravanan, S Mary Raja Slochanal, P Venkatesh, and J Prince Stephen Abraham. Application of particle swarm optimization technique for optimal location of facts devices considering cost of installation and system loadability. *Electric power systems research*, 77(3-4):276–283, 2007.

[33] T Saravanan, G Saritha, and V Srinivasan. Optimal power flow using particle swarm optimization. *Middle-East Journal of Scientific Research*, 20(11):1554–1560, 2014.

[34] Andrew Sloss and Steven Gustafson. 2019 evolutionary algorithms review, 06 2019.

[35] IEEE Power Engineering Society. Ieee guide for the application, specification, and testing of phase-shifting transformers. *C57. 135*, 2002.

[36] Mutmainna Tania. *Wide Area Measurement Applications for Improvement of Power System Protection*. PhD thesis, Virginia Tech, 2013.

[37] Kenneth Van den Bergh, Jonas Boury, and Erik Delarue. The flow-based market coupling in central western europe: Concepts and definitions. *The Electricity Journal*, 29(1):24–29, 2016.

[38] Dirk Van Hertem, Jody Verboomen, Ronnie Belmans, and Wil L Kling. Power flow controlling devices: An overview of their working principles and their application range. In *2005 International Conference on Future Power Systems*, pages 6–pp. IEEE, 2005.

[39] Dr. Ulrich Heimeshoff Veit Böckers, Professor Justus Haucap. Benefits of an integrated european electricity market: the role of competition. 2013.

[40] Jody Verboomen, Dirk Van Hertem, Pieter H Schavemaker, Wil L Kling, and Ronnie Belmans. Phase shifting transformers: principles and applications. In *2005 International Conference on Future Power Systems*, pages 6–pp. IEEE, 2005.

[41] David C Walters and Gerald B Sheble. Genetic algorithm solution of economic dispatch with valve point loading. *IEEE transactions on Power Systems*, 8(3):1325–1332, 1993.

[42] Nan Wei. *A Line Outage Study for Prediction of Static Power Flow Redistribution.* PhD thesis, Virginia Tech, 2016.

[43] Wikipedia contributors. Breadth-first search — Wikipedia, the free encyclopedia, 2020. [Online; accessed 13-August-2020].

[44] H Lee Willis. *Power distribution planning reference book.* CRC press, 2004.

[45] Allen J Wood, Bruce F Wollenberg, and Gerald B Sheblé. *Power generation, operation, and control.* John Wiley & Sons, 2013.