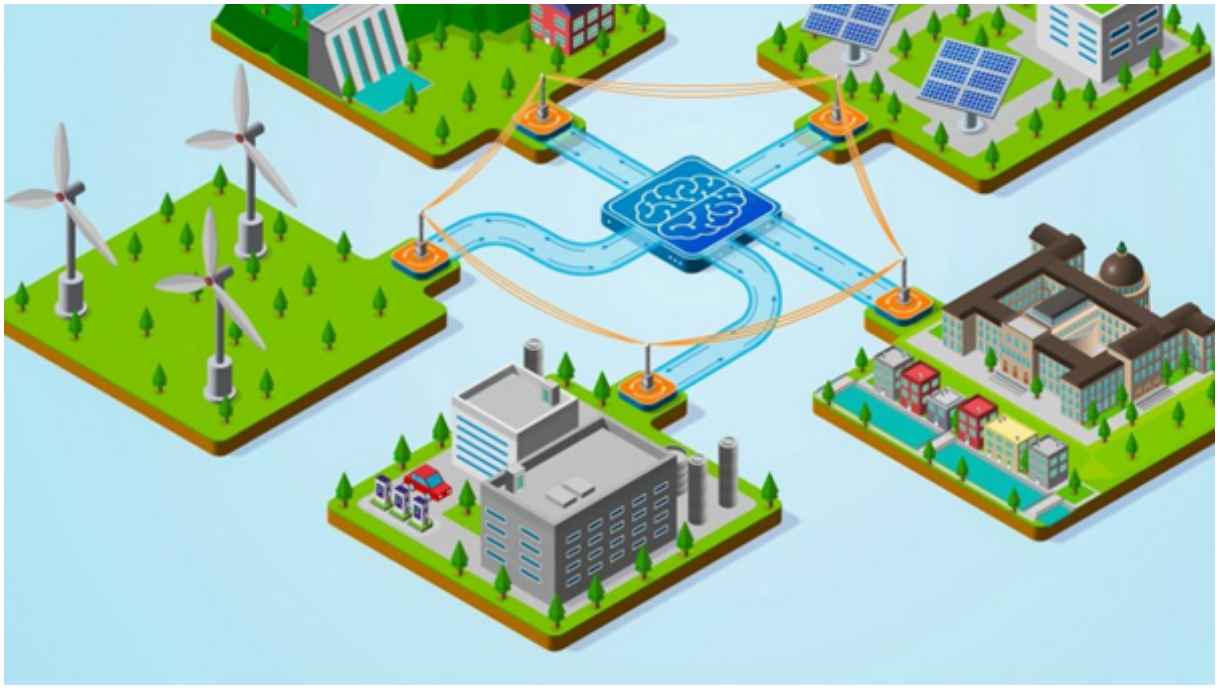
Multi-Objective Optimization of Toplogical Remedial Actions in Power Grids using Evolutionary Algorithms



Author: Job Groeneveld (2608853)

1st supervisor:	Alessandro Zocca
2nd supervisor:	Ger Koole
Supervisor TenneT:	Jan Viebahn

A thesis submitted in fulfillment of the requirements for the VU Master of Science degree in Business Analytics - Specialization in Optimization of Business Process

August 7, 2024

Contents

Chapter 1

Abstract

This thesis presents an innovative approach to optimize congestion management in electrical power systems through the application of Evolutionary Algorithms (EAs). This research addresses the growing complexity and congestion challenges faced by transmission system operators due to the increasing demand for electricity and the integration of renewable energy sources. These challenges require efficient management strategies to maintain grid stability and operational efficiency.

This study focused on dynamic grid topology reconfiguration, utilizing EAs to determine the optimal grid configuration at predetermined time intervals. The algorithms were designed to optimize multiple conflicting objectives, including minimizing N-0 and N-1 load flows, reducing the frequency of switching actions, minimizing topological depth, and maximizing the utilization of a reference topology using real-world data from TenneT's grid operations.

Six different EAs were developed to investigate the most effective methods for applying EAs to congestion management. Each developed EA targets a distinct set of objectives. Performance metrics such as hypervolume, inverted generational distance plus (IGDplus), and coverage were employed to assess the quality and effectiveness of these EAs. The findings confirm that EAs can be effectively applied to congestion management, with one algorithm scoring high in all performance metrics.

The top-performing EA in this study delivers actionable insights for TenneT and potentially other transmission system operators (TSOs) by offering a practical optimization model that enhances grid stability and efficiency. The research also identifies key factors influencing algorithm performance and suggests future research directions to improve the model's robustness and further applicability.

Chapter 2

Preface

This thesis has provided me with everything I had hoped for. My goal was to work on a complex optimization problem at a company with a sustainability purpose. I was immediately enthusiastic when this opportunity with TenneT came by. I enjoyed being part of the data science team, where I had the opportunity to grow into a more mature data scientist.

From a technical perspective, I have learned a lot about evolutionary algorithms, as well as managing a project in a company. Beyond that, I found it interesting to learn about the way TenneT is shaped and the challenges they currently work on. I am passionate about developing optimization models for the energy sector and I am looking forward to the path ahead.

This thesis not only taught me how to theoretically apply my knowledge but it also triggered me to think about the practical opportunity of an algorithm. Data science helps to overcome the challenges of the energy sector in such a way that sustainable solutions can develop further.

Throughout this journey, I faced challenges, but in general I feel that I learned from them. I could not have achieved this alone and I want to express my gratitude to those who supported me. First, I would like to thank Jan Viebahn for providing me with the opportunity to do this thesis at TenneT. Thank you for getting me started, for the enjoyable Monday morning meetings, and for bringing up all those innovative ideas. Thanks to Alessandro Zocca for supervising this assignment, sharing your knowledge during our meetings, and being critical when needed. You helped me to shape my thesis by providing structure and technical feedback. For both of you, the attentiveness, quick responses to my questions, and the time and effort you dedicated to my thesis were valuable to me. And, thanks to my direct colleagues at TenneT for making me feel welcome. I enjoyed both the technical and informal conversations with you all. Also, special thanks to Ger Koole for making the effort to be part of my committee.

Finally, and foremost, I want to thank the people closest to me. My friends and family for their support throughout this process. Your support made the thesis process much more enjoyable.

Job Groeneveld Amsterdam, August 2024

Chapter 3

Acronyms

- **AC** - Alternating Current
- **ASF** - Achievement Scalarization Function
- **CROF** - Control Room of the Future
- **CSSC** - Combined Solution Set Coverage analysis
- **DC** - Direct Current
- **DTR** - Dynamic Thermal Rating
- **EA** - Evolutionary Algorithm
- **ESS** - Energy Storage Systems
- **GA** - Genetic Algorithm
- **IGDplus** - Inverted Generational Distance Plus
- **LOLP** - Loss of Load Probability
- **MILP** - Mixed Integer Linear Programming
- **MOEA** - Multi-Objective Evolutionary Algorithm
- **N-0** - Normal Load Flow
- **N-1** - Single Contingency Load Flow
- **NSGA-II** - Non-dominated Sorting Genetic Algorithm II
- **NSGA-III** - Non-dominated Sorting Genetic Algorithm III
- **OTS** - Optimal Topology Switching
- **RES** - Renewable Energy Sources
- **RL** - Reinforcement Learning

- **RTE** - Réseau de Transport d'Électricité (French TSO)
- **TCGEP** - Transmission-Constrained Generation Expansion Planning
- **TSO** - Transmission System Operator

Chapter 4

Introduction

TenneT is responsible for managing congestion on 110 kV and higher voltage transmission grids. This involves performing a network security analysis across multiple time frames, including intra-day assessments. Starting in the evening before the business day, TenneT evaluates whether the grid will remain safe in terms of currents, voltages, and other criteria, taking into account power forecasts and the security limits of the elements of the grid. However, system operators are confronted with new (more generation-dominated) flow patterns. These new flow patterns are caused by, for example, the increase in renewable energy sources such as solar and wind, which are less controllable, geographically distributed, and variable in output. These new flow patterns have resulted in the electricity network becoming more congested, requiring system operators to operate the grid with an increasing number of interventions and/or closer to its limits.

This congestion is expected to increase in the future, putting greater pressure on maintaining high-level security of electricity supply. Congestion management is a real-world decision problem characterized by large action spaces, sequentially (including different time horizons), uncertainty, and multiple objectives. Due to the latter, decisions often need to be taken when trade-offs between conflicting objectives occur. For example, system operators simultaneously need to consider security constraints, hard time constraints, and financial costs.

This research is focused on dynamic grid topology reconfiguration. In dynamic grid topology reconfiguration, the grid is reconfigured at pre-set time intervals using an optimization algorithm. This method is a cost-effective and flexible solution for congestion management that leverages existing infrastructure. In addition, it can be implemented in the short term, whereas new electrical lines require several years to build. A promising strategy is the use of Evolutionary Algorithms (EAs). EAs are designed for solving complex optimization problems, particularly those that are too difficult to solve using exact methods due to their size or complexity.

4.1 Problem Background

TenneT is making significant efforts towards the energy transition. The energy transition is the global shift from traditional energy sources, such as fossil fuels (coal, oil, and natural gas), to renewable and sustainable energy sources, such as wind and solar. The biggest obstacle to achieving this is the current energy congestion problem. TenneT is heavily investing in projects to solve this issue and has identified the importance of dynamic grid topology reconfiguration to address it. Therefore, TenneT has launched the Control Room of the Future (CROF) program to support the development of new optimization tools. The program is confined to the 110 kV and 150 kv lines of Groningen and Drenthe. This area contains nine different substations that are used for grid operations. The analysis uses intra-day load flow calculations of N-0, N-1, and topological depth for one day, with data collected for each hour.

The goal of this project is to develop tools that provide decision support on topological measures to solve congestion more effectively and efficiently. This will contribute to the following aspects:

- By increasingly exploring the topological solution space, the tool can find, assess, and propose new solutions that were formerly unexploited, hence contributing to the security of supply.
- By leveraging non-costly topological remedial actions, the costs for congestion management can be reduced.
- By finding more capacity in the transmission grid, especially the meshed parts, maintenance can be facilitated, and new customer connections or expansions can be enabled.
- By providing timely recommendations, response time can be decreased, which is increasingly important given the growing volatility in the grid.

To achieve these goals, a decision support tool called GridOptions is being developed. The GridOptions tool will offer operators a set of optimal strategies (or schedules) that exhibit different trade-offs between the different four objectives mentioned above.

A method that handles both the large action space and multi-objective problem well are EAs. EAs are optimization techniques inspired by natural selection and genetics. They can explore large solution spaces effectively by evolving a population of potential solutions through selection, crossover, and mutation. Moreover, multi-objective EAs can provide an accurate estimation of the Pareto front, which is the set of optimal strategies representing the trade-offs between different objectives. This allows operators to choose the most suitable strategies based on specific needs and priorities.

During my internship at TenneT as part of the CROF team, I investigated how EAs can be applied to the congestion management problem and applied them to a real-world dataset. This thesis describes how a Multi-Objective Evolutionary Optimization Algorithm (MOEA) can be applied to the congestion management problem and includes a detailed methodology to evaluate its effectiveness.

4.2 Problem Statement

The electricity network consists of substations, transmission lines, energy generators (e.g. power plants and renewable energy sources), and energy consumers (e.g. residential, commercial, and industrial loads). Energy generators and consumers input and take energy out of the system, respectively. These are called load injections, which cause energy to flow through the network. Operators use load flow analysis to calculate load flows on transmission lines. This analysis includes both normal operating conditions (N-0) and contingency scenarios (N-1) where an element in the network fails, ensuring the system's robustness. Throughout the day, load injections vary due to changing demand from energy consumers and the fluctuating supply from energy generators. These variations are influenced by factors such as the weather conditions for renewable sources and the operating schedules of the power plants. To manage these fluctuations and prevent transmission lines from becoming overloaded, operators can change the grid configuration, allowing for a more optimal distribution of load flows and ensuring a balanced and efficient energy distribution across the network. However, operators have to take multiple objectives into considerations. These include:

- The $N-0$ and $N-1$ loadflows: Ensuring acceptable load levels in both normal and contingency scenarios is crucial to maintain system stability and reliability.
- The complexity of the network: Some topologies (definition in section 6.1) have more complex configurations than others, making them harder for operators to understand and increasing the risk of operational mistakes.
- The number of topology switches: A topology that efficiently distributes energy at one time can cause congestion later in the day. Frequent switching degrades assets and increases the likelihood of failure.

4.3 Research Objective

The goal of this research is to provide system operators with a set of optimal strategies that exhibit different trade-offs between multiple objectives. Here, a strategy is defined as a sequence of topologies. The choice of the final strategy is left to the preferences of the human operator. A mathematical representation of the topological properties and objectives is given in Chapter 6.

To achieve the goal of this research, EAs are applied to a real-world congestion management problem in electrical power systems. The behavior of EAs are studied by experimenting with different components, such as testing different selection algorithms, objectives, and evaluation functions.

Therefore, this research seeks to answer the following question:

> **"How can Multi-Objective Evolutionary Algorithms (MOEAs) be applied to effectively optimize the congestion management problem?"**

This question involves both a "how" and a "what" aspect, which results in the following subquestions.

1. How to apply an EA to the congestion management problem? There is investigated how an EA should be specifically designed and adapted to address our congestion management problem, 6.4.

2. What EA method is most effective? Different configurations of the designed EA are tested and compared. The effectiveness of each method is evaluated based on several criteria, such as solution quality.

4.4 Contributions

By developing MOEAs optimized for the management of electrical power systems, this research aims to provide actionable insights for TenneT to enhance the stability and efficiency of the grid. The algorithm balances multiple objectives, which minimizes load flows N-0 and N-1, reduces switching timestamps, minimizes topological depth, and maximizes reference topology. This comprehensive approach enables TenneT to optimize their operations, improve system reliability, and manage grid transitions more effectively. Additionally, the research results in a practical tool that can be integrated into TenneT's existing workflows to support strategic decision-making and operational optimization. These contributions can inspire future research and practical applications in optimizing multi-objective problems in other domains, such as transportation, logistics, and telecommunications.

4.5 Structure of Thesis

The rest of this thesis is structured with the following chapters to answer the research question in 4.3. Chapter 5 explains the literature on optimal topology switching (OTS). Chapter 6 outlines the specific congestion management problem faced by TenneT, explaining the electrical grid's topology, key objectives, and the formulation of these objectives as a multi-objective optimization problem. Chapter 7 provides an overview of the data used in this research, detailing the structure and contents of the data files and pre-processing steps. Chapter 8 describes the EAs developed for this research, including chromosome representation, population initialization, fitness functions, selection methods, and genetic operators. Chapter 9 details the methodology employed, including the objectives used for optimization, the different EAs compared, and the evaluation measures used to assess their performance. Chapter 10 explains the evaluation measures to assess the performance of each EA. Chapter 11 presents the results of the EAs, analyzing their performance in optimizing the specified objectives, the diversity of solutions, and computation time. Chapter 12 interprets the results, addresses limitations, and integrates insights from the data analysis to identify key factors influencing the performance of the algorithms. And finally, chapter 13 summarizes the key findings, discusses practical applications for TenneT, and offers recommendations for future research directions.

Chapter 5

Literature Review

This chapter explains the literature on optimal topology switching (OTS) which is identical to the concept of "dynamic grid topology reconfiguration" introduced in Chapter 4. There is literature available on the OTS problem, however, there are many variants of the problem. This includes the difference in load flow computations used, e.g., Alternating Current (AC) or Direct Current (DC) approximations explained in subsection 6.2. Secondly, there are flexibility options such as dynamic thermal rating (DTR), energy storage systems (ESS), and renewable energy sources (RES) that are modeled in the OTS problem. Lastly, data can be discrete or stochastic [29]. Our problem is characterized by using DC load flow calculations. Our model does not consider any flexibility options; it only allows for changes to transmission lines. Furthermore, the data used in our model are discrete in nature.

5.1 Optimization Algorithms

5.1.1 Classical Optimization Problems

In the literature, approximated DC OTS problems are typically solved using mixed integer linear programming (MILP), [29]; [28], [30], [22], [1]. The objective in these models is to minimize the total generation, which focuses on efficiency by redistributing power flows more effectively. Transmission line capacities are modeled as a constraint [29].

5.1.2 Topology Optimization using Reinforcement Learning

There are also several papers on grid topology optimization using reinforcement learning. In 2019, the French TSO RTE launched the Learning to Run a Power Network (L2RPN) challenge, encouraging diverse researchers to use reinforcement learning (RL) for power network maintenance [25]. Several RL solutions have been created, including a paper by van der Sar et al. (2023) presents a multi-agent reinforcement learning framework, making use of the problem's inherent hierarchical structure [32]. Soft actor-critic discrete and proximal policy optimization agents have been used in order to find the best actions and policies. Another paper utilizes the popular AI algorithm AlphaZero, which has achieved great success in games such as chess and Go [9]. This study explores the application of AlphaZero to optimizing the topology of the power grid, demonstrating its effectiveness in handling large-scale combinatorial optimization tasks.

5.1.3 GridOptions Algorithm

TenneT has also developed its own heuristics for its tool called GridOptions [39]. The approach uses a dynamic programming algorithm combined with heuristics. Initially, load-flows are computed for various network topologies using power load and generation forecasts. This list is narrowed down with a heuristic favoring topologies with the longest congestion-free periods. Finally, a sequential decision graph of network states is built and strategies are derived using random weight sampling together with Dijkstra's algorithm.

5.1.4 Evolutionairy Algorithms

An OTS problem has also been solved using a Genetic Algorithm (GA) [13]. A GA is an EA that is typically used for single-objective optimization. Another study effectively used an MOEA to solve OTS problems with two objectives. The additional objective in this case was to minimize the total generation cost and the Loss of Load Probability (LOLP), which indicates the probability that a power system will fail to meet the prescribed load during a specified period. Solutions derived from this approach can help system operators better balance two key objectives: cost and reliability. The paper approximated the Pareto front of these two objectives [41].

5.1.5 Similar Applications in Field of Electrical Power

Next to OTS problems, MOEA has been applied to other problems in the field of electrical power. The comprehensive survey on NSGA-II for multi-objective optimization and applications provides several examples of how NSGA-II has been applied to combinatorial problems in the field of electrical power [24]. NSGA-II is a selection algorithm used in MOEAs. The selection algorithm is also used in this research is be explained in Section 8.5.1. In the field of electrical power, NSGA-II has been used in the IEEE 30-bus test system. The IEEE 30-bus test system is a widely used standard in power system analysis and research, helping researchers and engineers evaluate the performance and effectiveness of different optimization algorithms in a controlled environment [24]. In a study by Murugan et al. (2019) NSGA-II was applied to the transmission-constrained generation expansion planning problem (TCGEP) [27]. The TCGEP problem involves determining the optimal locations, capacities, and types of new power generation facilities to be added to an existing power grid. The experimental results on the IEEE 30-bus test system by Murugan et al. (2019) validated the effectiveness of NSGA-II. This aligns with other studies on NSGA-II that solve the TCGEP problem, [27] , [23], [15], [34].

5.1.6 Evolutionary Algorithms for the Shortest Path

In section 6.4 it is explained that the congestion problem can be composed as a multi-objective shortest-path problem. To conclude the literature review, relevant papers covering MOEAs solving a shortest path problem were found. Gen M. and Lin L. (2006) proposed a new chromosome formulation for random key-based GA, showing significant improvements over standard chromosome formulations [12]. Beke et al. (2021) compared different chromosome representations for a shortest path problem to determine which is the most effective. Other papers demonstrated the effective use of EAs on the shortest path problem [16], [35], [17] .

Chapter 6

Mathematical Problem Background

This chapter provides a mathematical representation of the congestion management problem. It begins with key definitions of the electricity grid. Then, it introduces the DC approximation method for load flow analysis, explaining its assumptions and mathematical formulations. Next, it covers load flow computations, focusing on capacity utilization and robustness metrics (N-0 and N-1 load flows). Finally, the chapter models the congestion management problem as a shortest path problem and outlines the sets, parameters, and objectives for optimizing network performance.

6.1 Key Definitions in Electricity Network

In this section, definitions related to electricity networks are provided. These definitions are used throughout the following sections to explain the technical details of the network's operation.

- **Topology**: The configuration of the interconnection of components such as generators, transformers, and transmission lines in the power grid.
- **Substation**: A substation is a location in the electrical grid where the electricity flow is managed. It includes the following components:
  - **Busbars**: Busbars are conductive bars or strips within a substation that serve as a common connection point for multiple circuits. They distribute electrical power from incoming electricity lines to outgoing electricity lines of a substation
  - **Couplers**: Couplers, or bus couplers, are switches or circuit breakers that connect two sections of busbars within a substation. They allow for the reconfiguration of the network (i.e., switching topology). Busbar couplers can be open (disconnected) or closed (connected).

  An operator can control the load flows on the electricity grid by changing the topology, which involves opening or closing specific busbar couplers in the substations.
- **Susceptance:** Susceptance is a measure of how easily electricity flows through a transmission line.
- **Injection:** These are a combination of generator and load injections.

  - **Load injections:** These are points where power is generated and injected into the network. These injections can come from power plants and renewable energy sources.
  - **Load Injections:** These are points where power is withdrawn from the network. Examples of this this include residential, commercial and industrial loads.

- **Voltage angles:** Represent the phase differences between buses. These are used to determine the direction and magnitude of power flows and the resulting currents in the network.

6.2 DC Approximation

Electric power can be transmitted and utilized in two main forms: alternating current (AC) and direct current (DC). Each type of current has unique characteristics and applications. In AC, the voltage and current change magnitude and direction in a sinusoidal pattern. This current is used in the electricity grid because of its ability to easily be converted to high and low voltage. Transmitting electricity at high voltage results in lower current for the same power level, which reduces energy losses. Because of this, AC can efficiently transmit electricity over long distances.

In contrast, DC is an electric current that flows in a constant direction without changing its magnitude. The voltage in a DC circuit is also constant. DC is commonly used in batteries, electronics, and devices like laptops and mobile phones. To determine the load flow, or the amount of current on a transmission line AC or DC loadflow computations are used. AC computations are difficult to perform because they involve solving non-linear equations due to the sinusoidal pattern.

In addition, they require precise data on system parameters, which can be challenging to obtain and forecast accurately. Therefore, load flow forecasts are calculated using the DC approximation. This is because the equations that need to be solved are linear, which makes them much easier to solve. The data used in this research is also calculated using DC load flow computations. In the rest of this section, a mathematical explanation is given on how to derive the capacity utilization and calculate it for all transmission lines using DC load flow computations.

- $G = (V, E)$: Graph representing the buses and transmission lines in the network
- P_i : Net real power injection at bus $i \in V$, which is the generation minus load
- θ_i : Voltage angle at bus $i \in V$
- B_{ij} : Susceptance between transmission line $(i, j) \in L$
- f_{ij} : Power flow between transmission line $(i, j) \in E$

In addition to this sets,the following assumptions need to be made:

- **Constant Voltage Magnitudes**: All voltages are assumed to be constant instead of sinusoidal.
- **Ignore Reactive Power**: Only real power is considered, and reactive power is ignored.

- **Small Angle Differences**: Voltage angle differences between buses are small, so the following approximations are valid $\sin(\theta_i - \theta_j) \approx \theta_i - \theta_j$ and $\cos(\theta_i - \theta_j) \approx 1$.

The elements of the substance matrix $\mathbf{B_{ij}}$ can be calculated using the following formulas.
For the diagonal elements:

$$B_{ii} = \sum_{j \in B | j \neq i} b_{ij} \tag{6.1}$$

For the off-diagonal elements:

$$B_{ij} = \begin{cases} -bij & \text{if there is a direct line between bus i and j} \\ 0 & \text{if there is no connection between bus i and j} \end{cases} \tag{6.2}$$

In order to calculate the power injections the following formula is used:

$$\mathbf{P} = \mathbf{B} \cdot \mathbf{\Theta} \tag{6.3}$$

The voltage angles $\mathbf{\Theta}$, can be calculated by rearranging Equation (6.3) as follows:

$$\mathbf{\Theta} = \mathbf{B}^{-1} \cdot \mathbf{P} \tag{6.4}$$

Once the voltage angles θ_i are known, the power flow f_{ij} on each line can be calculated using the following equation:

$$f_{ij} = B_{ij}(\theta_i - \theta_j) \tag{6.5}$$

6.3 Load Flow Computations

In an electricity network, each transmission line has a set capacity, which is fixed and predetermined by TenneT. Based on this capacity, the capacity utilization, which measures how much capacity is used, of all transmission lines is calculated. The capacity utilization of a transmission line is determined by dividing the load flow by the predetermined capacity and then multiplying by 100 to get a percentage. This percentage indicates how much of the line's capacity is being utilized.

The load flows $N-0$ and $N-1$ are metrics used to evaluate the capacity utilization of the network under different conditions. The $N-0$ load flow represents the maximum capacity utilization of a topology under normal operating conditions. Operators use this metric to evaluate how strained the network is under normal circumstances. The $N-1$ load flow represents the maximum capacity utilization a topology in case a transmission line fails. This is also known as a contingency analysis. This metric tests the robustness of the network against line outages, ensuring that the network can handle the loss of a single line without becoming overly strained.

To calculate the $N-1$ load flow, a contingency list is used. This list contains critical elements of the grid, such as transmission lines and transformers. Each element in this list is taken out one by one, and the $N-0$ flows are recalculated for the remaining topologies. The $N-1$ load flow is the maximum load flow observed in all scenarios. In the next subsection, a mathematical representation of these load flow values is provided.

6.3.1 Calculation of $N-0$ Load Flows

- K_{ij} : Capacity for transmission line i, j, defined by TenneT
- l_{ij} : Capacity utilization at tranmission line (i, j)
- n : Number of transmission lines

For transmission line (i, j) the capacity utilization is calculated as follows:

$$l_{ij} = \frac{P_{ij}}{K_{ij}} \tag{6.6}$$

The $N-0$ load flow for a certain topology can now be calculated as follows:

$$N-0 = \max\{l_{ij} \mid 1 \leq i, j \leq n\} \tag{6.7}$$

6.3.2 Calculation of $N-1$ Load Flows

In this subsection a mathematical representation of these loadflow values are given:

- C : Set of contingencies
- P_{ijc} : Net real power injection at transmission line (i, j) under contingency $c \in C$
- l_{ijc} : Capacity utilization at transmission line (i, j) under contingency $c \in C$

For each contingency c, the way it impacts the power injections and the susceptance of the network is re-evaluated. Then, the net power injections at each bus can be recalculated as described in Section 6.2. Having recalculated P_{ijc}, the capacity utilization for each transmission line (i, j) is calculated as follows:

$$l_{ijc} = \frac{P_{ijc}}{K_{ij}} \tag{6.8}$$

The maximum capacity utilization over all transmission lines for a given contingency c is calculated as follows:

$$L_c = \max\{l_{ijc} \mid 1 \leq i, j \leq n\} \tag{6.9}$$

The load flow $N-1$ is the maximum capacity utilization across all contingencies:

$$N-1 = \max\{L_c \mid \forall c \in C\} \tag{6.10}$$

This calculation identifies the worst-case scenario for the network in any single contingency.

6.4 Problem Objectives

As explained in the introduction, the goal of this research is to provide system operators with a set of optimal strategies that exhibit different trade-offs between different objectives. The final choice of a strategy is determined by an operator. The list below provides a description of the objectives identified by TenneT. These objectives are mathematically represented in Section 6.4.2.

- **Minimize the maximum N-0 load flow**: This refers to the highest N-0 load flow value across all time steps.
- **Minimize the maximum N-1 load flow**: This refers to the highest N-0 load flow value across all time steps.
- **Minimize the number of open busbar couplers**: i.e., minimize the topological depth
- **Minimize the amount of switching timestamps**: i.e., maximize the duration of the topologies are used
- **Maximise the utilization of reference topology**: Maximise the number of timestamps the reference topology is used. e.g. all busbars closed.

The problem is treated as a multi-objective optimization problem. Multi-objective optimization involves finding a set of non-dominated solutions, also known as Pareto-optimal solutions. These solutions represent the best trade-offs between the conflicting objectives, where no single solution is superior in all objectives simultaneously.

6.4.1 Shortest Path

The values $N-0$, $N-1$, and the topological depth depend on the topology and the timestep in which it is used. The problem, which is defined in 6.4, can be modeled as a graph $G = (V, E)$, where V represents the topology and timestep combinations as nodes. The connecting edges E represent the possible topology changes from one timestep to the next. The goal is to find a set of topologies called a strategy that minimized the objectives defined in Section 6.4. In this shortest-path problem, the starting node is a random node in the first timestep, and the destination is one of the nodes in the last timestep. To handle this scenario with multiple potential starting and ending points, a super-source (s) and super-destination (d) node is added to the start and end of the graph, respectively. The super-source node connects to all nodes in the first timestep with zero-weight edges, ensuring the shortest path can start from any node in the first timestep. Similarly, the superdestination node connects to all nodes in the last timestep with zero-weight edges, allowing the shortest path to end at any node in the last timestep. Figure 6.1 provides an illustration of this problem. Framing the congestion management problem as a shortest path problem will help model it mathematically. This is done in the next subsection.

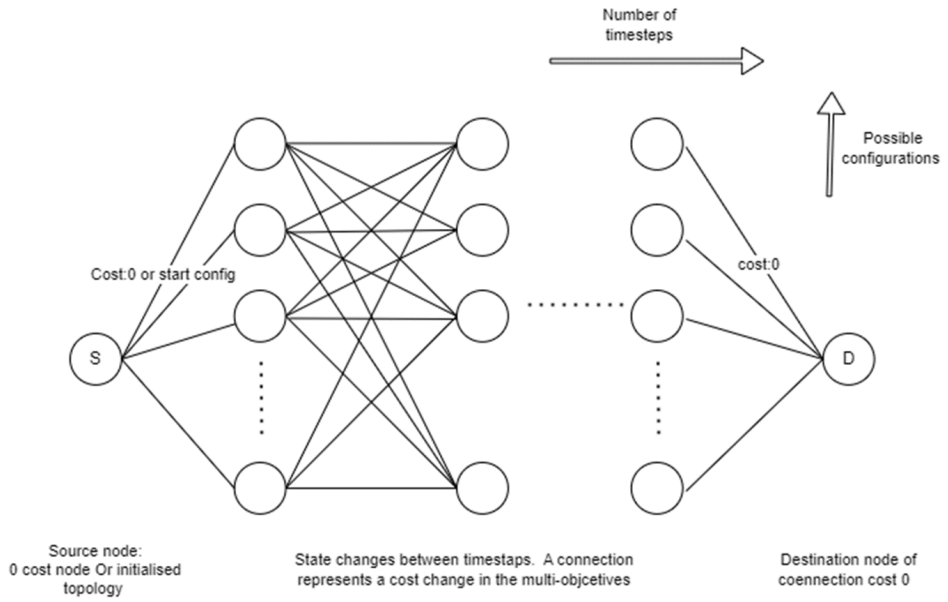


Figure 6.1: Illustration of shortest path problem. S and D are the source and destination nodes respectively having zero-weight edges. The other nodes represent a transition from one topology to another topology in the next timestep.

6.4.2 Sets and Parameters

In this subsection, a mathematical representation of the congestion management problem is given. First the sets and parameters are introduced below:

- The set of timesteps, T, ranges from 0 to 26, excluding the source and destination nodes.
- The set of topology IDs at timestep t is represented by J_t.
- The set of nodes, V, consists of combinations of timesteps and topology IDs, denoted as (t, j) where $t \in T$ and $j \in J_t$.
- The set of edges, E, includes edges from node (t, j) to node $(t + 1, j')$ where $j' \in J_{t+1}$.

Additional variables include:

- $n0_{tj}$ representing the $N - 0$ loadflow at node (t, j).
- $n1_{tj}$ representing the $N - 1$ loadflow at node (t, j).
- d_{tj} representing the topological depth loadflow at node (t, j).

The variable r_{tj} indicates if the reference topology is not used, defined as:

$$r_{tj} = \begin{cases} 1 & \text{if } i \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

The variable $s_{tjj'}$ indicates if there is a switch between topologies, defined as:

$$s_{tjj'} = \begin{cases} 1 & \text{if } j \neq j' \\ 0 & \text{otherwise} \end{cases}$$

6.4.3 Objective Functions

Using the defined sets and variables, the objectives in Section 6.4 are mathematically represented below.

1. **N − 0 loadflow:** $\max\{n0_{tj} \mid \forall (t,j) \in V\}$
2. **N − 1 loadflow:** $\max\{n1_{tj} \mid \forall (t,j) \in V\}$
3. **Topological depth:** $\max\{d_{tj} \mid \forall (t,j) \in V\}$
4. **Mean depth:** $\{\sum_{(t)\in T} d_{tj}/|T| \mid \forall j \in J_t\}$
5. **Not reference topology used:** $\sum_{(t,j)\in V} r_{tj}$
6. **Number of switches:** $\sum_{(t,j,j')\in E} s_{tjj'}$

Chapter 7

Dataset

This chapter introduces the dataset used for analyzing congestion management problem explained in 6.4. It outlines the information contained in the dataset, details the pre-filtering steps applied, and describes the size of the solution space.

7.1 Attributes of Dataset

The dataset contains data of 27 timesteps containing of the following features; timestep topology id, $N-0$, $N-1$ load flows and topological depth. The dataset has 161014 unique topologies and 366611 rows, which represent the number of nodes in out problem ($|V| = 366611$). The $N-0$ and $N-1$ loadflows are calculated using the method explained in section 6.3. The dataset has been pre-filtered to include only data where the topological depth is less than 3. Depth values higher than 3 are considered too complex by operators. Furthermore, the dataset does not contain load flows $N-0$ and $N-1$ above 110. This pre-filtering step simplifies our objective space, making it less complex and easier to work with. In Table 7.1 below, the number of topologies and the mean of $N-1$ per timestep are given. In timesteps 15, 16 17, and 18 the number of topologies is lower and the mean topological depth is higher. This indicates that it is a time frame of high congestion, the lower number of topologies is the result of the pre-filtering step.

Timestep	0	1	2	3	4	5	6	7
#Topologies	17570	17600	17658	17615	17606	17656	17664	17664
mean($N-1$)	78.6	78.5	78.1	77.1	77.0	76.8	78.0	77.1
Timestep	8	9	10	11	12	13	14	15
#Topologies	17676	17676	17602	15976	15781	14907	5675	530
mean($N-1$)	77.1	77.2	77.9	77.2	76.8	83.8	96.0	97.3
Timestep	16	17	18	19	20	21	22	23
#Topologies	577	476	472	4283	12460	15465	15976	17565
mean($N-1$)	93.2	96.4	93.6	95.1	92.9	76.3	76.4	78.6
Timestep	24	25	26					
#Topologies	17451	17459	17571					
mean($N-0$)	78.6	78.6	77.9					

Table 7.1: #Topologies and mean(N-0) per timestep.

7.1.1 Topological Depth Count

In Table 7.2, the count of topological depths is presented, showing the frequency of the number of topologies for each depth value. The dataset can be seen as unbalanced, having only one topology with a depth value of 0 and significantly higher counts for larger depth values.

Topological depth	**Count**
3	139640
2	20938
1	435
0	1

Table 7.2: Topological IDs and corresponding depth count.

7.1.2 Analysis of Objective Space

The objective is equal to all possible topologies combinations for each timestep, which is $\prod_{t \in T} n_t$. Where n_t is the number of nodes in timestep t of set T. This is equal to:

$$S = \prod_{i \in I} n_i = 451 \cdot 10^{16} \tag{7.1}$$

Chapter 8

Algorithm Description

This chapter focuses on how a multi-objective EA can be applied to the congestion management problem defined in Section 6.4.

8.1 Description of the Evolutionary Algorithm

An EA is a computational method inspired by the principles of natural selection and genetics. It is designed to find optimal or near-optimal solutions to complex problem iteratively improving its solutions. The solutions are represented as a population, evolving them over successive generations to improve the overal quality. according to a predefined fitness function. The core process of an EA can be summarized as follows and is illustrated in Figure 8.1:

- **Initialization**: Start with a randomly generated population of candidate solutions.
- **Evaluation**: Assess the fitness of each candidate using a fitness function that quantifies the quality of the solution.
- **Selection**: Select a subset of the current population based on fitness, giving preference to better-performing candidates to serve as parents for the next generation.
- **Variation**: Generate new candidates (offspring) through recombination and mutation:
  - **Crossover**: Combine parts of two or more parent solutions to create one or more offspring.
  - **Mutation**: Introduce random modifications to individual parent solutions to create new offspring.
- **Replacement**: Form the next generation by selecting among the combined pool of parents and offspring, often based on fitness.
- **Iteration**: Repeat the evaluation, selection, variation, and replacement steps until a termination condition is met, such as reaching a satisfactory solution or a predefined number of generations.

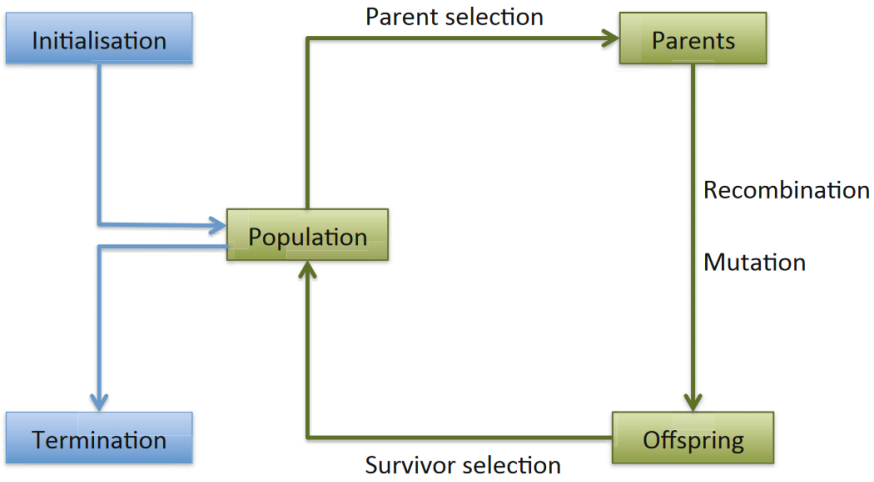


Figure 8.1: The general scheme of an evolutionary algorithm as a flowchart. The blocks marked in blue and green are part of the initialization and evolution process, respectively [10].

The process in an EA is inherently stochastic, involving randomness in selection, recombination, and mutation. This stochasticity ensures that the population remains diverse and that a wide range of solutions is explored, which prevents premature convergence to local optima [10]. In the following subsections, each component is explained, including what it is and how it is specifically defined and adapted to address the congestion management problem (Section 6.4). The list below specifies what components are needed to run an EA [10]:

- Chromosome representation
- Initial Population
- Fitness Function
- Selection Method
  - Multi-Objective Evolutionary Algorithm (MOEA)
- Genetic Operators
  - Crossover ('exploitation')
  - Mutation ('exploration')
- Stopping Criterion

8.2 Chromosome Formulation

The first step in defining an EA is to decide how possible solutions should be specified and stored in a way such that they can be manipulated by a computer. That is, to set up a link between the

original problem context and the problem-solving space where evolution takes place . Objects forming possible solutions within the original problem context are called **phenotypes**, while their encoding, that is, individuals within the EA, are called **genes** [10]. The chromosome representation encodes potential solutions (phenotypes) into a format that the EA can manipulate (genotypes) . This encoding allows the algorithm to apply various evolutionary operators such as selection, crossover, and mutation to evolve solutions over time. For example, consider the value 18. This value is a phenotype that represents a possible solution in its original context. Within an EA, this value needs to be encoded in a format that the algorithm can process. The encoded version of this value, using a binary representation, would be 10010. This binary string is the genotype that corresponds to phenotype 18 [10]. An effective chromosome formulation ensures that all feasible possible solutions can be represented within the genotype space. It also guarantees a clear and invertible mapping between genotypes and phenotypes, meaning that each genotype corresponds to at most one phenotype. This invertibility is crucial for decoding the optimized solution (a phenotype) from the best-performing genotype after the evolutionary process ends [10].

The Figure 8.2 below illustrates the components of chromosome formulation using biology-oriented terminology. Each chromosome consists of several genes. The value of a gene is called an allele, and the index or position of a gene within the chromosome is called the locus. Chromosomes are part of a population.

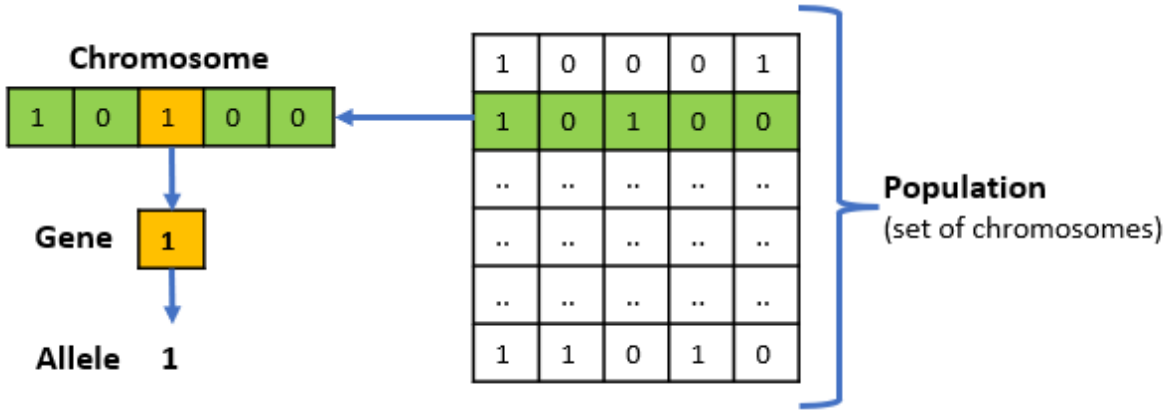


Figure 8.2: A population consists of chromosomes that represent solutions in the genotype space. Each chromosome comprises genes, where each gene contains a specific value known as an allele [37].

There are several ways in which the congestion management problem can be encoded (Section 6.4), which is based on the shortest path problem. Beke et al. (2021) compared three different methods, which are further analyzed in this section [2]. For each encoding, there is explained how it works and what the advantages and disadvantages are. These encodings are:

- Direct Variable Length Encoding
- Integer-Valued Priority-Based Encoding
- Random Key-Based Encoding

The representations are explained using the example Figure 8.3 below.

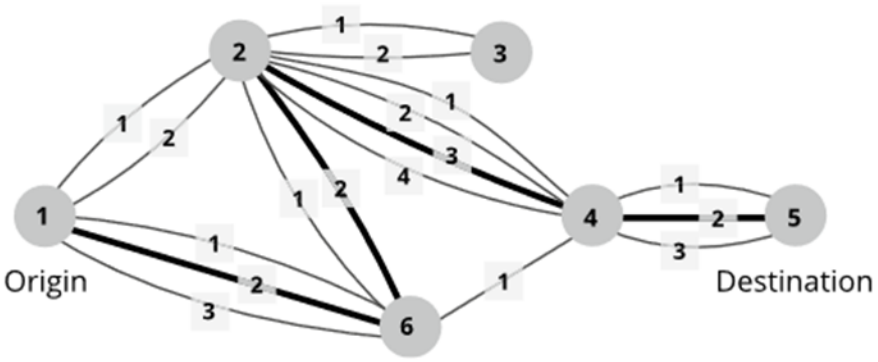


Figure 8.3: Example graph of the shortest path problem. The shortest path is shown in bold [2].

8.2.1 Direct Variable Length Encoding

With direct variable length encoding the number of genes is equal to the length of the path. The path is denoted by a sequence of IDs , which represent the nodes that are travelled to. The solution to Figure 8.3 is [1, 6, 2, 4]. Beke et al. (2021) argued that the main advantage of this representation is that it gives a one-to-one mapping, which is usually preferable over one-to-many mapping, since it avoids introducing plateaus in the search space [2]. In one-to-n mapping, several different chromosomes might encode the same solution path, and thus have the same associated fitnesses, forming a plateau [2]. Secondly, the genotype representation is equal to the phenotype representation. This direct mapping ensures that genetic operators can be applied directly to the phenotype, which simplifies the EA process. Finally, the chromosome length is equal to the number of timestamps, which is the smallest and, therefore, the most efficient memory usage method out of the three representations. The paper argues that a disadvantage of using this method is that genetic operators could lead to loop formation, and therefore the offspring need to be checked and repaired after mutation and crossover [2]. However, in our stated shortest-path problem, it is not possible to form a loop since at every timestamp it is only possible to move to a node in the next timestamp and not to a node in the previous or current timestep.

8.2.2 Integer-Valued Priority-Based Encoding

With integer-valued priority-based encoding the number of genes is equal to the number of nodes. The locus(index) of the chromosome represents the node id. The value represents a priority, which indicates what the next node should be given the connections of a node. For example, the path [1, 6, 2, 4] in Figure 8.3 is formulated as [6, 3, 1, 2, 5, 4] using integer-valued priority-based encoding. Below we can see how these values were formulated.

1. Start origin node 1 available connections are $[6, \mathbf{3}, 1, 2, 5, \mathbf{4}]$. $\max\{3, 4\} = 4$, node 6 is chosen.

2. Start node 6 available connections are $[6, \mathbf{3}, 1, \mathbf{2}, 5, 4]$. $\max\{3, 2\} = 3$, node 2 is chosen.

3. Start node 2 available connections are $[6, 3, \mathbf{1}, \mathbf{2}, 5, 4]$. $\max\{2, 1\} = 2$, node 4 is chosen.

Beke et al. (2021) argues that the main advantage of this representation is that a random permutation of the priorities will always be decoded to some valid path starting from the origin node [2]. This means that more traditional crossover operators can be used and expected to produce feasible paths, unlike in the case of direct representations.

8.2.3 Random Key-Based Encoding

Random key-based encoding is an improved version of integer-valued priority-based encoding. It works in the same way as integer-valued priority-based encoding, where a path is constructed based on the available connections and the corresponding key values. The formulation is better since there exist better crossover and mutation operators for float-represented genotypes.
In the integer-valued priority-based encoding and random-key-based encoding, genotype and phenotype space are represented differently. According to Beke et al. (2021) Random key-based encoding seems to be the superior representation of the three [2]. The genotype space with this representation is well defined, allowing for good use of the operators which leads to better convergence. This method works well for general shortest path problems with a reasonably sized node count. However, in our specific problem (Section 6.4), the length of a chromosome becomes excessively large, as it equals the number of nodes. With 366,611 nodes (7) in this problem, maintaining a reasonably sized population, such as one thousand, will require substantial memory space (1000 * 366,611) and also require a lot of computational power to evolve the population.

The main advantages of this method are that it consistently forms a valid path and ensures that no loops are created after crossover and mutation. However, because of the structure of our problem, it is not possible to form loops. The only connection available from one node to another is to move to a node on the next timestamp. The advantages of random key-based encoding which form a problem in direct encoding will therefore be negligible.

As a result, direct variable-length encoding has been opted for as the chromosome representation. This approach minimizes memory usage, and the phenotype is the same as the genotype. To use direct representation for our problem, the chromosome values are represented by the parameter V_{tj} defined in Section 6.4. The values (t, j) correspond to the locus and allele value of a gene, respectively. A chromosome C is represented as follows: $\{V_{0j}, V_{1j}, \ldots, V_{nj}\}$.

8.3 Initial Population

A population is a collection of individual solutions, each represented as a chromosome. Before the evolutionary process begins, the initial population must be generated. The prerequisites for the initial population are that it must contain valid solutions and be diverse. This diversity ensures that the algorithm has a large search space to explore, which increases the chances of finding an optimal solution.

There are several ways to generate the first population. A common method is to randomly generate solutions. The problem with this approach is that the population will not be diverse in the number of switches. This is because the likelihood of having the same topology two or

more times in a row is very low due to the large number of possible topology choices at each timestep. The same issue arises for maximum depth. As shown in Table 7.2, the distribution is skewed. There are more topologies with a depth value of 3 than there are with a depth value of 0. Random sampling over this set will also result in many solutions having an max depth of 3, which again does not make our initial population very diverse.

8.3.1 Custom Initialization Method

Instead of random sampling, a new custom sampling method is defined. The initial population is sampled using a new approach. This new approach is formulated below. the sets from subsection 6.4.2, have been used.

Let S be the total solution space described in section 7.1.2

- Let $W_t \subseteq S$ be the set of solutions where t switches occur, for $t \in T$.
- Let $R_t \subseteq W_t$ be a random sample from W_t, for $t \in T$.
- Then, the initial population P_S is the union of all random samples R_t.

$$P_S = \bigcup_{t \in T} R_t \tag{8.1}$$

- Let $D_t \subseteq S$ be the set of solutions with topological depth t, where $t \in \{0, 1, 2, 3\}$.
- Let $F_t \subseteq D_t$ be a random sample from D_t, for $t \in \{0, 1, 2, 3\}$.
- Then, the initial population P_D is the union of all random samples F_t.

$$P_D = \bigcup_{t \in \{0,1,2,3\}} F_t \tag{8.2}$$

The final initial population is:

$$P = P_S \cup P_D$$

8.3.2 Custom Initialization Method Switches

In this subsection a method for attaining any sample R_t is given. Let K_i be the set of topology ids in $i \in T$. We define a function $f(i, j)$, where $(i, j) \in T$ and $i < j$:

$$(i, j) = \bigcap_{p=i}^{j} K_p \tag{8.3}$$

This formula is then used to randomly create a chromosome for any number of switches s explained in Algorithm 1 below.

Algorithm 1 Randomly Create Chromosome with s Switches

Input: Number of switches s
Output: Solution array C
Initialize set $I = \{0, 26\}$
Randomly select s elements from $T \setminus \{0, 26\}$ without replacement
Add the selected elements to set I
Sort set I in increasing order
Initialize array C with length 27
for each pair (i, j) in I **do**
    Compute $f(i, j)$
    for $k = i$ to j **do**
        $C[k] = f(i, j)$
    end for
end for
return C

8.3.3 Custom Initialization Method Toplogical Depth

This subsection explains the custom initialization method for attaining any sample F_t. Let K_i be the set of topology ids in $i \in T$. Let $w_{ij} \subseteq K_i$ be the set of topology ids having depth $j \in \{0, 1, 2, 3\}$. We define a function $f(i, d)$:

$$f(i, d) = \bigcup_{p=0}^{d} w_{ip} \tag{8.4}$$

This formula is then used to randomly create a chromosome for any number of switches d explained in Algorithm 2 below.

Algorithm 2 Randomly Create Chromosome with Topological Depth d

Input: Depth value d
Output: Solution array C
Initialize set $K' = \{f(i, d) \mid \forall i \in T\}$
Initialize array C with length 27
for each i in T **do**
    Randomly select topology t from K'
    $C[i] = t$
end for
return C

8.4 Fitness Formulation

The fitness function defines the criteria that the population must adapt to meet. It serves as the foundation for selection, guiding the algorithm towards progressively better solutions by quantifying which chromosomes are better [10]. To assign a fitness value, a chromosome is typically converted from its genotype to its phenotype, and then its quality is assessed. For example, if

the objective is to find an integer x that maximizes x^2, the fitness of a binary-encoded genotype would be determined by decoding it to its integer form and then squaring that integer [10]. In most cases, the fitness function is equivalent to the objective function(s) or a slight variation of it.

In case of the congestion management problem, the fitness function should evaluate a population based on the objectives defined in subsection 6.4.2. Objectives 5 and 6 are evaluated using the corresponding objectives' functions on the chromosome genotype. The other objectives are calculated using matrix multiplication m which is done as follows. Let $V_t = \{V_{tj} \mid \forall j \in J_t\}$ A sparse design matrix $\mathbf{B}$ contains V_t on the diagonal and is constructed to represent the topologies J_t for each timestep in the set T This matrix is constructed in the following way:

$$\mathbf{B} = \operatorname{diag}(V_1, V_2, \ldots, V_{|T|}) = \begin{bmatrix} V_1 & 0 & \cdots & 0 \\ 0 & V_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & V_{|T|} \end{bmatrix}$$

$\mathbf{a}_1$ is a sparse boolean vector that indicates which topologies in J_t 6.4.2 have been used. A chromosome is then represented as:

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{bmatrix}$$

A population consists of multiple $\mathbf{a}$ vectors, which are represented by $\mathbf{A}$. Using matrix multiplication, the load flow $N-0$, $N-1$ or topology depth can be calculated represented by matrix $C = AB$. The final step consists of taking the maximum over every row. The last step is to apply the corresponding objective functions 1-4 in subsection 6.4.2.

8.5 Selection Method

The selection method chooses which chromosomes is passed on to the next generation, typically favoring those with higher fitness scores to ensure that good traits are preserved and propagated. In literature, there is differentiated between three kinds of selection algorithms. Single optimization algorithms, multi-objective, and many objectives. The difference between single and multi is that multi can solve up to two objectives at the same time, while many can solve more than two. Since our stated problem consists of two and many objectives, there is focused on in the latter two.

In a comparative study by Ma et al. (2023), five different frameworks were compared, consisting of Nondominated Sorting Genetic Algorithm (NSGA-II), NSGA-III, Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), Indicator-Based Evolutionary Algorithm (IBEA), and Multi-Objective Particle Swarm Optimization (MOPSO), [8], [7], [4], [5], and (Comparative study:[24]). NSGA-II and NSGA-III stand out for their use of the Pareto dominance principle and crowding distance, which are explained in more detail in Section 8.5.1. These methods ensure the population maintains a diverse set of solutions by preserving non-dominated solutions and distributing solutions uniformly across the solution space. The other algorithms require some additional configurations, such as a pre-defined weight vector for MOEA/D or certain particle choices in MOPSO. These are undesirable because they introduce complexity and

the potential for bias in the optimization process. Pre-defined weight vectors or specific particle choices require careful tuning and may affect the fairness in treating all objectives equally. This can lead to suboptimal performance if not configured correctly. In contrast, NSGA-II's simpler setup allows for more straightforward implementation and reduces the risk of introducing such biases.

In recent years, the number of applications that use NSGA-II has been steadily increasing, with more than 600 publications in 2022 alone [24]. Several of these publications focus on applications in the electrical power industry. Some works closely related to the congestion management problem are discussed in the literature review subsection 5.1.3. For these reasons, the NSGA framework has been chosen. Its proven effectiveness in electrical power field applications and its easy setup compared to other selection algorithms make it a suitable choice for our problem. The following subsections will explain how these algorithms work and highlight their advantages.

8.5.1 NSGA-II

NSGA-II, is an improved version of the original NSGA. The algorithm is based on two key components:

- Nondominated sorting
- Crowding distance

Each method is explained in the following two subsections.

Nondominated sorting

Nondominated sorting works by assigning non-dominated levels to each solution. This is done using the following formula.

$$\phi(x) = \sum_{y \in P} I(x, y)$$

where

- P : All solutions in population
- x : A solutions in the population
- y : A solution where x is compared against.
- $I(x, y)$: Defined as follows:

$$I(x, y) = \begin{cases} 1 & \text{if } y \text{ dominates } x, \\ 0 & \text{otherwise.} \end{cases}$$

Ranks are assigned based on the non-dominated levels as follows:

1. Calculate $\phi(x)$ for each solution x in the population P.
2. Sort the solutions in ascending order based on their $\phi(x)$ values.

3. Assign ranks to the solutions such that solutions with the same $\phi(x)$ value share the same rank. Solutions with lower $\phi(x)$ values receive higher (better) ranks.

4. Group the solutions into fronts F_i, where each front F_i contains solutions with the same rank.

Points having level 1, belong to the Pareto front.

Crowding distance

The crowding distance is a distance metric between two poiints and is calculated bu the following formula he crowding distance $d(x)$ is determined by:

$$d(x) = \sum_{i=1}^{k} \left(f_i^+(x) - f_i^-(x) \right) \tag{8.5}$$

where

- $d(x)$: The distance metric for solution x.
- k: The number of objectives or criteria.
- $f_i^+(x)$: The positive contribution of the i-th objective for solution x.
- $f_i^-(x)$: The negative contribution of the i-th objective for solution x.

A large crowding distance means that a solution x is located in a less crowded region of the objective space. This implies that the solution is relatively isolated from other solutions, making more diverse compared to solutions with smaller crowding distances.

NSGA-II process

Having explained the key elements of NSGA-II, an explanation of the process can be given. Let t denote the generation. The process starts with a population P_t of size N and an offspring population Q_t of size N. The combined population R_t is formed by merging the parent population P_t and the offspring population Q_t, resulting in a population of size $2N$. The population R_t is then sorted into non-dominated fronts 8.5.1.

If the size of the best nondominated set F_1 is smaller than N, all members of F_1 are chosen for the new population P_{t+1}. The remaining members of the population are chosen from subsequent nondominated fronts in order of their ranking until no more sets can be accommodated. If the last non-dominated set F_k is only partially accommodated, the solutions within F_k are sorted using the crowding distance to select the most diverse solutions to fill the the new population P_{t+1}. The process is illustrated in the following Figure 8.4.

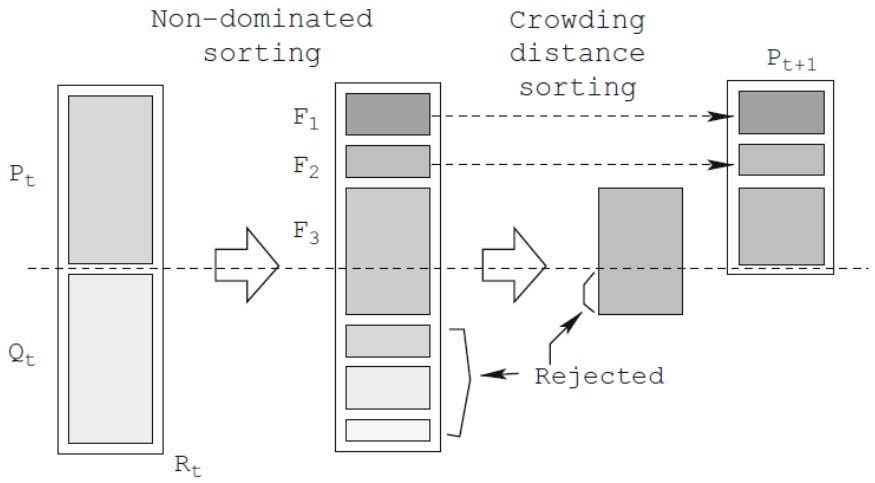


Figure 8.4: Process of NSGA-II algorithm. Uses non-dominated sorting and crowding distance to generate new population P_{t+1}

This iterative process ensures that each new generation of solutions maintains high quality and diversity, progressing towards an optimal Pareto front.

8.5.2 NSGA-III

NSGA-III builds on the framework of NSGA-II but introduces a new approach to maintain diversity in the population, specifically developed for many-objective optimization problems. Instead of relying on crowding distance to ensure diversity, NSGA-III generates a set of reference points on a hyperplane. These reference points are created using distribution methods that aim to evenly distribute them across the hyperplane. Two popular distribution methods in the literature are the Das-Dennis method and the Riesz s-Energy method [6], [14]. In his research, the Riesz s-Energy method is used to generate reference points. The Riesz s-Energy method ensures a more uniform distribution of reference points across complex objective spaces, scales more effectively as the number of objectives increases, and is more adaptable to different types of objective landscapes compared to the Das Dennis method[14]. In Figure 8.5 an illustration is provided of the constructed reference points using Riesz s-Energy method in a three dimensional space.

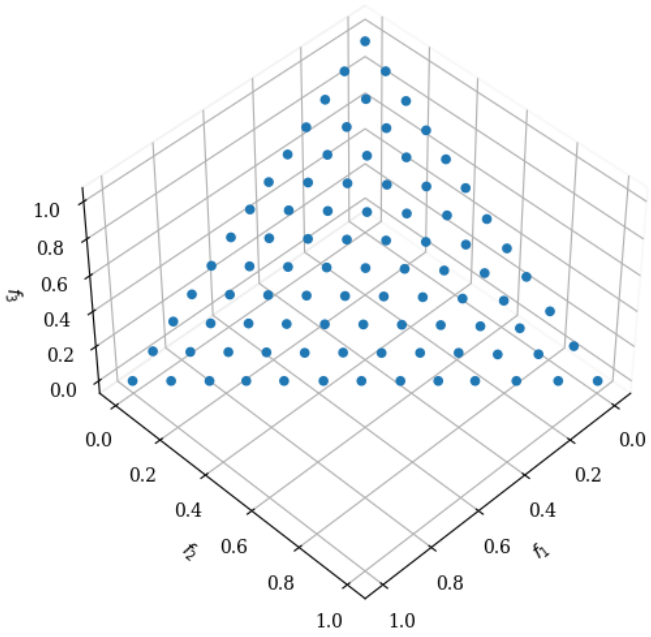


Figure 8.5: Reference poins using Rienz's-Energy method [3].

Having initialized the reference points, the next step is to standardize the solutions. This process involves calculating the nadir, which are the worst points and finding the extreme points. In order to calculate these points, the following set and parameter are defined:

- m: Number of objectives
- S: A set of solutions, where each solution has m objective values

Each solution $x \in S$ is represented as a vector of objective function values:

$$f(x) = \{f_1(x), f_2(x), \ldots, f_m(x)\}$$

Where $f_i(x)$ is the i-th objective value for the solution x. The nadir points can be calculated as follows

$$\mathbf{z}^{\text{nad}} = \left[\max_{x \in S} f_1(x), \max_{x \in S} f_2(x), \ldots, \max_{x \in S} f_m(x),\right] \tag{8.6}$$

To find extreme points for each objective axis, the Achievement Scalarization Function (ASF) is used. The ASF helps in identifying solutions that are extreme or dominant in one particular objective direction [7]. This is done as follows: For a solution $x \in S$, minimize the ASF:

$$ASF(x, w) = \max_{i=1,\ldots,m} \left\{ \frac{f_i(x) - z_i^{\text{nad}}}{w_i} \right\} \tag{8.7}$$

where $w = (w_1, w_2, \ldots, w_m)$ is the weight vector representing the axis direction. For each weight vector w, the ASF function identifies a solution that is extreme in the i-th objective direction. This results in an extreme objective vector $z_{i,\max}$. The extreme vectors $\{z_{1,\max}, z_{2,\max}, \ldots, z_{m,\max}\}$ are used to construct an m-dimensional linear hyperplane. The intercept a_i of the i-th objective

axis and this hyperplane can be computed [7]. Using the intercepts and the nadir points, the objective functions are normalized as follows:

$$f_i^{'}(x) = \frac{f_i(x) - z_i^{\text{nad}}}{a_i - z_i^{\text{nad}}} \tag{8.8}$$

Figure 8.6, provides an illustration of the explanation above.

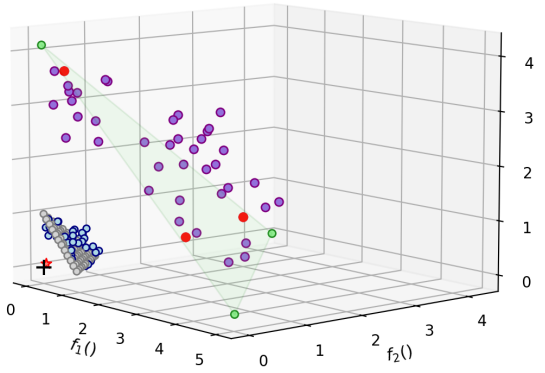


Figure 8.6: The solutions (purple) are scaled based on the nadir(red) and extreme points (green) to the normalised points (blue) [26].

Subsequently, each solution $f_i^{'}(x)$ is associated with a reference point. Each reference point corresponds to a line connecting the reference point with the origin, known as a reference direction.

For each solution $f_i^{'}(x)$, the the perpendicular distance to every reference direction w_j is calculated, using the formula:

$$d^{\perp}(f_i^{'}(x), w_j) = f_i^{'}(x) - \frac{w_j^T f_i^{'}(x)}{||w_j||} \tag{8.9}$$

Each solution $f_i^{'}(x)$ is associated with the reference point that corresponds to the reference direction w_j for which the perpendicular distance is smallest.
Figure 8.7 provides an illustration of this association process.

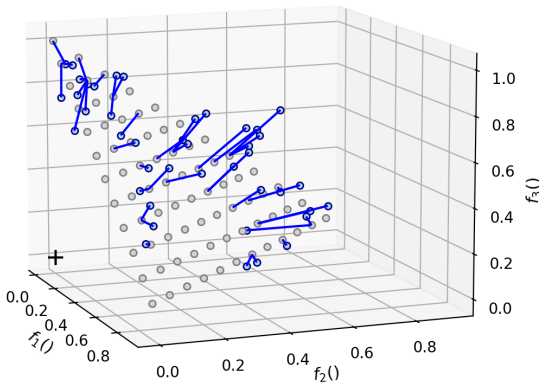


Figure 8.7: Association of solutions to the reference points. Each solution is associated with the closest reference point [26].

Finally, the number of points associated to a reference point is counted. This is known as niching and is used to determine which solutions to retain. Instead of sorting solutions by their crowding distance, the algorithm looks at how many solutions are associated with each reference point. Solutions associated with reference points with a low niche count are preferred to fill the remaining slots in the population.

The selection process is similar to NSGA-II, where the combined population R_t is sorted into non-dominated fronts using non-dominated d-order. Let F_l be the front that cannot be accommodated. Then all population members from the non-dominated front level 1 to level l are first included in S_t. When the last front F_l is partially accommodated, the algorithm looks at the fraction $\frac{F_l}{S_t}$ to determine which solutions from F_l should be included to complete the population P_{t+1}. This ensures that the final population maintains both diversity and convergence to the Pareto front. This new population P_{t+1} is used for the next generation.

8.6 Genetic Operators: Crossover and Mutation

This section explains crossover and mutation operators that are used. Crossover and mutation drive the processes of exploitation and exploration, respectively. Crossover involves combining the genetic information of two parent solutions to produce offspring. This operator facilitates exploitation by allowing the algorithm to mix and match the best traits from different parents, thereby focusing on and refining promising areas of the search space. The goal of crossover is to generate new solutions that inherit the desirable features of their parents, increasing the likelihood of improving the overall quality of the solution. Mutation, on the other hand, introduces random changes to individual solutions. This operator is used for exploration as it enables the

algorithm to maintain genetic diversity within the population and prevents premature convergence to local optima. By randomly altering parts of a solution, mutation helps the algorithm explore new and potentially better regions of the search space that might not be reachable through crossover alone.

8.6.1 Crossover

The challenge in choosing a crossover operator is that the resulting offspring must remain valid solutions. In Section 7, there is explained that the dataset has an uneven distribution in the number of available topologies per timestep. This uneven distribution can create problems when chromosomes are combined during crossover, potentially resulting in offspring with invalid topology values for certain timestamps. Therefore, a crossover method should be chosen that will not cause this problem. One such method is k-point crossover. This method works by selecting k crossover points within the chromosome, the parent chromosomes are then divided into $k+1$ segments. The offspring chromosomes are created by alternating these segments between the two parents. Here follows a mathematical explanation.

- Let P_1 and P_2 be two parent chromosomes, each represented as a sequence of genes:

$$P_1 = [p_{1,1}, p_{1,2}, \ldots, p_{1,n}]$$

$$P_2 = [p_{2,1}, p_{2,2}, \ldots, p_{2,n}]$$

- Let C_1 and C_2 be the resulting offspring chromosomes.
- Let K be the number of crossover points, where K is a positive integer less than n.
- Let $c_1, c_2, \ldots, c_K$ be the crossover points, where $1 \leq c_1 < c_2 < \ldots < c_K < n$.
- K crossover points $c_1, c_2, \ldots, c_K$ are randomly selected along the length of the parent chromosomes.
- The offspring chromosomes C_1 and C_2 are created by alternating segments between the two parents at the crossover points.

$$C_1 = [p_{1,1}, p_{1,2}, \ldots, p_{1,c_1}, p_{2,c_1+1}, \ldots, p_{2,c_2}, p_{1,c_2+1}, \ldots, p_{1,c_3}, \ldots, p_{2,c_K+1}, \ldots, p_{2,n}]$$

$$C_2 = [p_{2,1}, p_{2,2}, \ldots, p_{2,c_1}, p_{1,c_1+1}, \ldots, p_{1,c_2}, p_{2,c_2+1}, \ldots, p_{2,c_3}, \ldots, p_{1,c_K+1}, \ldots, p_{1,n}]$$

Figures 8.8 and 8.9 are also examples of k-point crossover with $k = 1$ and $k = 2$ respectively.

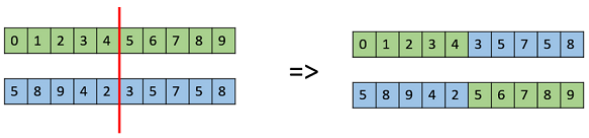


Figure 8.8: Example of one-point crossover: Genes are exchanged between two chromosomes at the randomly selected red line crossover point [38].

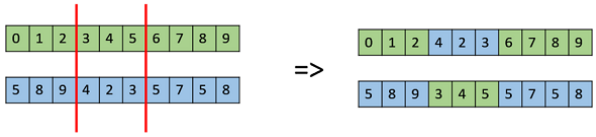


Figure 8.9: Example of multi-point crossover: Genes are exchanged between two chromosomes at several randomly selected crossover points, shown by red lines [38].

8.6.2 Mutation

The same challenge that exists when choosing a crossover operator also applies to mutation, the remaining offspring must remain valid solutions. One effective mutation operator that achieves this is a random reset mutation. In this method, a gene is randomly chosen, and a new value from the set of permissible values is chosen as new allele for the gene which ensures that the chromosome remains valid.

The reset mutation operator works as described in Algorithm 3:

Algorithm 3 Mutate Population

1: **Input:** Population matrix X, Set J_t 6.4.2, and mutation probability p
2: **Output:** Mutated population matrix X
3: random_matrix $\leftarrow$ Initialize a random matrix with values between 0 and 1 that has the same dimensions as X
4: mutation_indices $\leftarrow$ find indices where random_matrix $\leq p$
5: mutation_values $\leftarrow$ select topologies at mutation_indices based from corresponding set J_t
6: X[mutation_indices] $\leftarrow$ mutation_values
7: **return** X

Chapter 9

Methodology

This chapter provides an overview of the methodology employed in this thesis, focusing on applying an EA to TenneT's congestion management problem. The goal is to explore how different objectives influence the performance and diversity of solutions within both the objective and feature spaces. This is achieved through a detailed testing approach and specific evaluation measures. This methodology helps to identify what methods can efficiently provide the best and most diverse solutions for operators to apply to the electricity grid.

9.1 Objectives

In this section, it is explained what objectives are used, how many algorithms are compared, and what algorithms optimize which objectives. Each algorithm uses the implementation of the EA described in Chapter 8. A total of six different algorithms are compared. Each algorithm is optimized for a subset of the objectives defined in subsection 6.4.2. The six EAs, and the corresponding objectives they optimize, are detailed in Table 9.1.

Table 9.1: Objectives optimized by different EA.

Objective	EA1	EA2	EA3	EA4	EA5	EA6
max $N-1$	x	x	x	x	x	x
max topo depth	x	-	-	x	-	x
# switching actions	-	x	-	x	x	x
# reference topo	-	-	-	-	-	x
mean topo depth	-	-	x	-	x	-
$N-0$	-	-	-	-	-	-

9.1.1 Motivation for comparing EAs

In this section, a motivation is given for why the different EAs are chosen to be compared. The motivation is partly based on the correlation matrix between the objectives, which can be seen in Figure 9.1 below.

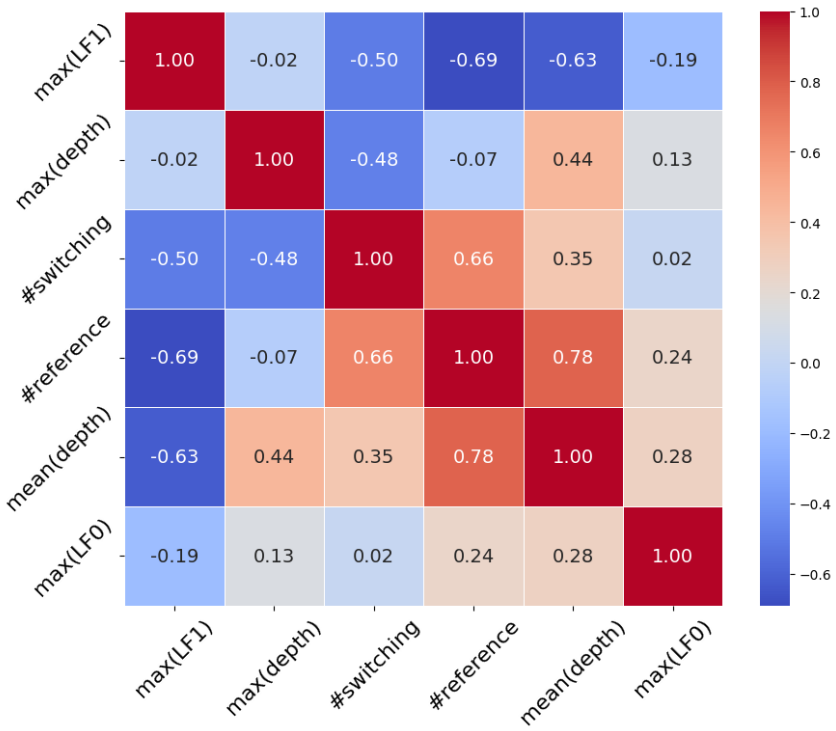


Figure 9.1: Correlation matrix of objectives.

The decision of the chosen objectives is based on the following reasons:

- **N − 0 is not chosen as an optimization objective**
  Table 9.1 shows that each algorithm is not optimised on $N-0$. $N-0$ and $N-1$ must both be below a certain threshold. In practice, $N-0$ is always less than $N-1$, meaning that if $N-1$ is below the threshold, $N-0$ is as well. Consequently, this makes optimising on $N-0$ unnecessary.

- **Difference between selection algorithms NSGA-II and NSGA-III**
  Three algorithms are optimized on two objectives using the NSGA-II selection algorithm, while three other EAs were optimized on three or more objectives using the NSGA-III selection algorithm. The results of these different optimization strategies are compared to evaluate how solutions optimized for two objectives perform relative to those optimized on three or more objectives. This analysis aims to determine whether the Pareto front projected onto a two-objective space from a many-objective (three or more objectives) optimization is equivalent to the Pareto front obtained from the EA optimized directly for those two corresponding objectives. This helps us to understand if the multi-objective approach captures the same quality of solutions in the two-objective space or if dedicated optimization on just two objectives provides distinct advantages.

- **Difference between aggregation functions**
  From the correlation matrix 9.1, it can be seen that max(depth) is not significantly correlated with max$(N-1)$. However, mean(depth) shows a strong negative correlation with max$(N-1)$. By optimizing max (depth) and mean (depth) separately, it can be analyzed

how much the mean or max operator influences the end result and determine which operator would be best to optimize to achieve a balanced solution between topological depth and $N-1$.

- **Combining objectives due to high correlation**
  The correlation matrix 9.1 shows a high correlation between the mean(depth) and the reference objective. The chosen EAs allow for an analysis of whether these objectives can be effectively combined, potentially simplifying the optimization process or if there is a need to optimize them separately.

- **Relaxing reference condition**
  One of the objectives defined by TenneT is to maximize the duration spent in the reference topology. This is the topology in which operators spend the most time and, therefore, feel more comfortable handling situations. From a research perspective, it is interesting to observe the results if this objective were relaxed. This relaxation could reveal whether the end results differ significantly when this objective is not strictly prioritized.

- **All objectives of interest for perator**
  EA6 optimizes all objectives that are relevant to the operator.

9.2 Testing Approach

This section describes the methodology for testing the different EAs to ensure that each algorithm is evaluated fairly and consistently. First of all, each EA seen in Table 9.1 is run 100 times on different seeds. A seed is a setting that initializes the random number generator to ensure reproducibility of the results. Running an EA 100 times, the effect of randomness on the outcome can be analyzed. Furthermore, this mitigates the influence randomness has on our end solutions, ensuring that the observed performance is due to the algorithm's inherent capabilities rather than random chance. This approach provides a more robust and reliable evaluation of the EA's performance.

Secondly, each EA is optimized on different objectives, and the final solutions of each EA will reside in their respective objective spaces. To compare the results between the EAs, a common objective space is defined by the following objectives: maximizing $N-1$, maximizing depth, minimizing the number of switches, and minimizing the number of references. The final results of each EA, are projected on this common objective space. This projection to a common objective space can be mathematically explained as follows: let $\mathbf{f}^u(\mathbf{x})$ denote the objective vector of a solution $\mathbf{x}$ evaluated by EA u in its specific objective space, that is called the **Target space**. To enable comparison across different EAs, we define a common objective space where each solution $\mathbf{x}$ is re-evaluated for the common objectives.
Thus, every s solution $\mathbf{x}$ from EA u is mapped to the common objective space by reevaluating $\mathbf{x}$ with the following function (9.1):

$$\mathbf{g}(\mathbf{x}) = [\max(N-1), \max(\text{depth}), \#\text{switches}, \#\text{reference}] \tag{9.1}$$

This re-evaluation ensures that all solutions are compared based on the same set of criteria, making it possible to analyze and compare the performance of different EAs.

9.2.1 Key Definitions

The following definitions support the comparison analysis and are used in the next sections of the thesis.

- **Combined population**: This is a population for a specific generation, formed by combining all the solutions from different seeds.
  Mathematically, let Z_j^i represent the set of solutions from seed $i \in M = \{1, \ldots, 100\}$ at generation $j \in \{0, \ldots, n\}$. The combined population C_j can be defined as:

$$C_j = \bigcup_{i \in M} Z_j^i \tag{9.2}$$

- **Combined Pareto front**: This is the Pareto front of the final combined population, consisting of the non-dominated solutions from the last generation.
  Mathematically, let $Pf(\cdot)$ be the function that returns the Pareto front of a set of solutions. The combined Pareto front PC_n for the final generation n is:

$$PC_n = Pf(C_n) \tag{9.3}$$

  where $Pf(\cdot)$ is defined in subsection 9.2.1.

These definitions can be used within the own objective space $\mathbf{f}^u(\mathbf{x})$ or in the common objective space $\mathbf{g}^u(\mathbf{x})$ of any EA u. The following definitions apply only to the common objective space.

- **Super population**: This is a population for a specific generation, formed by combining the population solutions of the common objective space from all different EAs.
  Mathematically, let C_j^u be the combined population of EA $u \in \{1, \ldots, 6\}$ at generation $j \in \{0, \ldots, n\}$. The super population S_j can be defined as:

$$S_j = \bigcup_{u \in \{1,\ldots,6\}} C_j^u \tag{9.4}$$

- **Super Pareto front**: This is the Pareto front of the super population from the final generation, consisting of the non-dominated solutions of the common objective space across all EAs.
  Mathematically, the super Pareto front PS_n for the final generation n is:

$$PS_n = Pf(S_n) \tag{9.5}$$

9.3 Evaluation Measures

The performance of the algorithms are assessed using several evaluation metrics. This section gives a brief overview of the evaluation measures used. Section 10 gives a more detailed explanation of the measures and a motivation for why they have been used. These measures are used on the combined Pareto front, in the objective spaces $f(\mathbf{x})^u$ and $g(\mathbf{x})^u$ for each EA u, respectively.

- **Hypervolume**
  The hypervolume measure is one of the most frequently applied measures to compare the results of evolutionary multi-objective optimization algorithms. The hypervolume indicator is appealing because it is compatible with any number of problem objectives, only needs one reference point, and requires no prior knowledge of the true Pareto-optimal front [33].

- **Inverted Generational Distance Plus (IGDplus)**
  IGDplus calculates the average distance from each point in the true or a reference Pareto front to the nearest point in the obtained set of solutions. This metric effectively combines the aspects of convergence, how close the obtained solutions are to a true or reference Pareto front, and diversity, how well these solutions cover the entire Pareto front[19].

- **Non-uniqueness**
  For a population of solutions, it can happen that different solutions represent the same point in objective space. This indicator measures the number of non-unique solutions.It indicates the amount of diversity in a set of solutions that is beneficial in a decision support context.

- **Coverage**
  Coverage is a measure for comparing different sets of PF approximations. It can be used to measure the contribution of a set S_1 to another solutions set S_2, also taking into account the non-unique points. This can be useful in two ways:

  1. **Seed sensitivity** In the own objective space of an EA, the coverage can be calculated of a set S_1 of EA run with seed i, to the combined Pareto front set S_2. In this way, an analysis of the seed sensitivity can be made.
  2. **Contribution super Pareto front** The number and specific points an EA shares with the super Pareto front can be analyzed. This analysis will indicate how well an EA is able to find the best solutions. solutions.

- **Topological diversity**
  This measures the number of different topologies used in a set of solutions. It indicates the most relevant topologies employed by an EA. A comparison can be made between a combined Pareto front and a Pareto front generated with a specific seed to see which topologies in which the strategies differ. Topological diversity can also be analyzed across the different EAs to compare the difference in topologies used in different EAs, where they overlap and where they are different.

- **Substation diversity**
  This metric measures the number of different substations utilized. Similar to topological diversity analysis, it allows for comparison between the EAs to evaluate differences in substations used.

- **Computation time**
  Finally, we can look at computation time. EAs that optimize on more objectives will generally require more time to compute. This difference in computation time can impact the feasibility of applying an EA,especially when there is a need for quick decision-making.

Table 9.2: Overview of the evaluation measures and corresponding evaluation domain.

Name	Domain
hypervolume	in objective space
non-uniqueness	on solution set
coverage	in objective space
substation diversity	in descriptor space
topological diversity	in descriptor space
computation time	per algorithm

9.4 Hyperparameters

In order to test the algorithms, every EA is run with the same hyperparameters to ensure a fair comparison. The hyperparameter settings for each EA can be seen in Table 9.3.

Table 9.3: Hyperparameters and their values.

Hyperparameter	Value
number of seeds	100
selection method	NSGA-II/ NSGA-III
crossover operator	4-point crossover
mutation operator	reset mutation
population size	722
crossover rate	0.9
mutation rate	0.037 (1/27)
termination parameter	700 generations

Table 9.4: NSGA-iii reference points configuration.

NSGA-iii parameters	Value
reference points method	Rienz s-Energy
number of reference points	300

9.5 Procedure of Testing the Different EAs

The methodology for analyzing the different EAs will have the following procedure:

1. **Run EAs**
   An EA, defined in Section 8, is applied to the congestion management problem, optimizing the specified objectives in Table 9.1. Every EA will use the same hyperparameters defined in subsection 9.4.

   This step involves running the algorithm multiple times, each time targeting different objectives. The algorithm is run on 100 different seeds to mitigate the influence of randomness when comparing different EAs and to analyze how much randomness has on the end result.

2. **Analysis of solutions using evaluation measures**

    - **Solution space** The solutions are analyzed within the target and common space. This analysis will help us understand the distribution and quality of the solutions, highlighting the effectiveness of the algorithm in achieving the desired objectives. The quality and distribution of the solutions are assessed using hypervolume and IGDplus.
    - **Feature space** The solutions will also be examined in the feature space, focusing on non-uniqueness, coverage, substation diversity, and topological diversity. This analysis will provide a deeper understanding of the characteristics of the solutions.

3. **Conclusions** Based on the analysis, we will determine: 1) if EA are effective for addressing congestion management problems, and 2) which specific EA(s) are best suited for this application.

By following this methodology, we aim to gain valuable insights into the influence of different objectives on the performance and diversity of solutions.

Chapter 10

Evaluation Measures

This chapter presents evaluation measures to assess the performance of each EA. It covers hypervolume, inverted generational distance plus (IGDplus), and computation time, which are standard measures in the EA literature [18]. Furthermore, custom measures designed for this problem have been defined. These are non-uniqueness, coverage, substation, and topology diversity, these are explained in detail in the corresponding subsections.

10.1 Hypervolume

The hypervolume indicator is the most widely used in the evolutionary multi-objective optimization community [18]. Firstly, it is the only measure that is Pareto compliant, which means that optimizing based on hypervolume leads directly to the Pareto front, capturing the best trade-offs among objectives [11]. Secondly, for this indicator, only one reference point must be set. Other performance indicators, such as IGD, require a set of reference points [33]. The hypervolume works by calculating the area of a defined reference point to a set of non-dominated solutions.

The left part of Figure 10.1 illustrates how that looks. In the left graph, two objectives are minimized, f_1 and f_2, a reference point is set at point r. This value should always be dominated by solutions, which are $a1, a2, a3, a4$ in the example illustration. Each point of the points $a1, a2, a3, a4$ dominates a region in the objective space that extends from the point itself to the reference point r. These regions form rectangles whose areas can be calculated. The total hypervolume is the sum of these individual areas. The hypervolume area is denoted by $HI(\{a_1, a_2, a_3, a_4\})$.

The same principle works for calculating the hypervolume in 3 dimensions.

In the right part of Figure 10.1 there is an additional objective f_3. Instead of an area, the volume of the set of non-dominated points Y is calculated.
The generalized method for calculating the hypervolume for the $(m \geq 2)$ numbers of objectives can be as follows. Let $S = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\}$ be a set of n non-dominated solutions, where any $\boldsymbol{a}_i$ represents a solution with values $(a_1, a_2, \ldots, a_k)$. A reference point $r = (r_1, r_2, \ldots, r_k)$ is chosen such that it is dominated by all the solutions in S. For each solution $\boldsymbol{a}_i$, the objective space, the volume V_i of the hyper-rectangle dominated by this solution and bounded by the reference

point r is given by the following equation:

$$V_i = \prod_{j=1}^{m} (r_j - a_{ij}). \tag{10.1}$$

The total hypervolume HV is the sum of the volumes of the individual hyper-rectangles for all solutions in the set $S = \{\boldsymbol{a}_1, \boldsymbol{a}_1 \dots \boldsymbol{a}_n\}$, which is given by the following equation:

$$HV(S, \mathbf{r}) = \sum_{i=1}^{n} V_i = \sum_{i=1}^{n} \left(\prod_{j=1}^{m} (r_j - a_{ij}) \right) \tag{10.2}$$

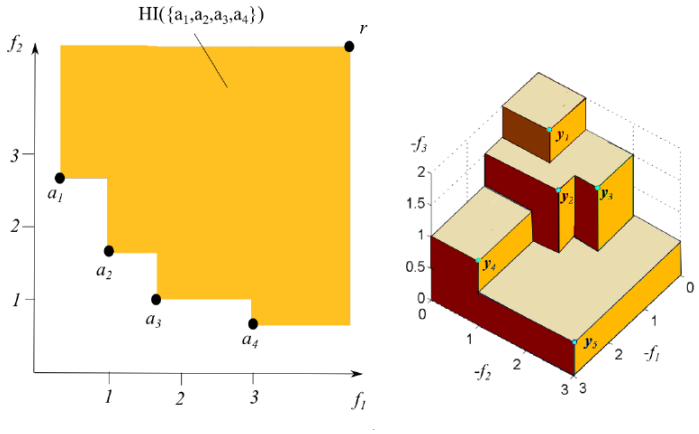


Figure 10.1: Hypervolume indicator in two dimensions for set $A = \{a_1, a_2, a_3, a_4\} \in \mathbf{R}^2$ (left) and in three dimensions for a set $A = \{y_1, y_2, y_3, y_4\} \in \mathbf{R^3}$ (right) [33].

10.1.1 Implementation of Hypervolume to Congestion Management Problem

For interpretability reasons, the solutions have been standardized by scaling each objective between 0 and 1. This process involves defining the following sets:

- $\mathbf{W} = \{\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_n\}$: A set of solutions representing the initial population of the super solutions, where each solution W_i is a vector of objective values
- $\mathbf{B} = \{\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_n\}$:A set of solutions representing the final population of super solutions, where each B_i solution is a vector of objective values

With these sets, the nadir and optimal points are calculated, these are used to scale the solutions. The set of nadir points z^{nadir}, can be calculated as follows:

$$\mathbf{z}^{\text{nad}} = \begin{bmatrix} \max(W_{11}, W_{21}, \dots, W_{n1}) \\ \max(W_{12}, W_{22}, \dots, W_{n2}) \\ \vdots \\ \max(W_{1m}, W_{2m}, \dots, W_{nm}) \end{bmatrix} \tag{10.3}$$

Where n is the population size and m is the number of objectives. Similarly, the set of optimal points can be calculated as follows:

$$\mathbf{z}^{\text{opt}} = \begin{bmatrix} \min(B_{11}, B_{21}, \ldots, B_{n1}) \\ \min(B_{12}, B_{22}, \ldots, B_{n2}) \\ \vdots \\ \min(B_{1m}, B_{2m}, \ldots, B_{nm}) \end{bmatrix} \tag{10.4}$$

Having found the worst and best points across all EAs, the solution sets can be standardized. The following formula is used to find the set of standardized solutions $S^{'}$ of the set of solutions S.

$$S^{'} = f(S) = \frac{S - z^{opt}}{z^{nad} - z^{opt}} \tag{10.5}$$

Now, a reference point $\mathbf{r}$ can be determined simply by: $r = (r_1, r_2 \ldots, r_m) = (1, 1, \ldots, 1)$ for m objectives since every solution a is at least better than r. The highest hypervolume that can be achieved is 1, meaning that a solution set S covers the whole objective space. Figure 10.2 is an example of the hypervolume in the standardised solutions space in two dimensions, represented by x and y. The hypervolume for this example is $HV(\mathbf{a}) = (a_2 : (0.6 \times 0.8) + a_1 : (0.4 \times 0.3) + a_3 : (0.1 \times 0.2)) = 0.47$ Solution $\mathbf{a}$, in this case covers almost half the space.

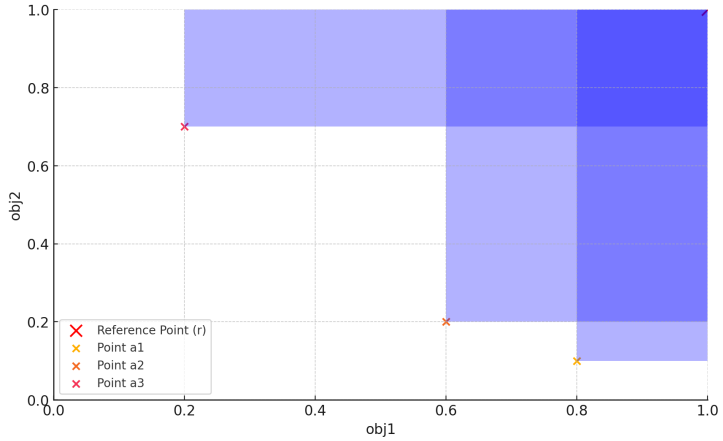


Figure 10.2: Hypervolume indicator in two dimensions for set $A = \{a_1, a_2, a_3\} \in \mathbf{R}^2$ in a standerdised objective space

10.2 Inverted Generational Distance Plus (IGDplus)

The second evaluation measure used is Inverted Generational Distance Plus (IGDplus). IGDplus evaluates the performance of a set of obtained solutions by comparing them with a set of reference points on the true or reference Pareto front. IGDplus is an improvement over the original IGD metric. The original IGD measures the average distance from a set of reference points to the nearest solution in the set of obtained solutions. However, it does not account for the dominance relationship between solutions and reference points. IGDplus solves this by making use of directional distance [18]. Instead of calculating the distance between a reference and nearest solution point, it calculates the distance to the nearest non-dominated space the set

of solutions covers. Figure 10.2 illustrates the difference. The directional distance is calulated using the equation below (10.6):

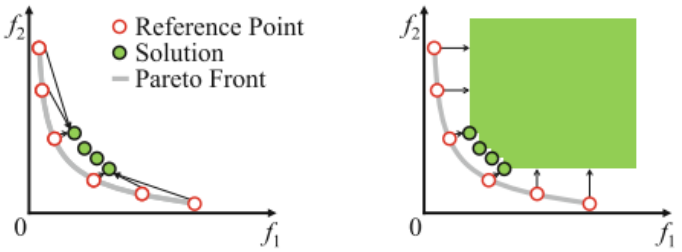
$$d^{+}(\mathbf{r}, \mathbf{a}) = \sqrt{\sum_{i=1}^{m} \max(r_i - a_i, 0)^2}, \tag{10.6}$$

where r_i and a_i are the i-th objective values (components) of the reference point $\mathbf{r}$ and the solution $\mathbf{a}$, respectively. The $\max(r_i - a_i, 0)$ function ensures that only positive deviations are considered, focusing on the areas where the reference point dominates the solution.
The IGDplus value is the average of the minimum directional distances over all reference points.

$$IGD^{+}(A, R) = \frac{1}{|R|} \sum_{\mathbf{r} \in R} \min_{\mathbf{a} \in A} d^{+}(\mathbf{r}, \mathbf{a}) \tag{10.7}$$

Where:

- A : A set of solutions $A = \{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}\}$
- R : A set of solutions $R = \{\mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_n}\}$



(a) IGD calculation. (b) IGD^{+} calculation.

Figure 10.3: Illustration of the IGD (a) and IGDplus (b)indicators for a minimization problem [18].

The additional use of IGDplus makes the solutions more interpretable. While hypervolume shows how much of the solution space is covered, IGDplus indicates how far off solutions are from the Pareto front. This combination provides a clearer understanding of both the extent and quality of the solutions, balancing coverage with convergence and distribution.

10.2.1 Implementation of IGDplus to Congestion Management Problem

The same standardization method is used as in the implementation of the hypervolume. The solution sets A and R are standardized using equation (10.5), resulting in sets $A^{'}$ and $R^{'}$, using the same nadir and optimal points calculated from (8.6) and (10.4), respectively. The IGDplus distances from sets $A^{'}$ and $R^{'}$ are calculated using equation (10.7); $IGD^{+}(A^{'}, R^{'})$.

10.3 Non-Uniqueness

Non-uniqueness measures how many distinct solutions represent the same point in the objective space but use a different strategy. This metric provides insight into the diversity of the solutions in terms of the strategies used. For each EA, the analysis involves calculating the number of non-unique solutions within their respective Pareto fronts. This is done by comparing the objective values and identifying solutions with identical objective values but different topological configurations. Formally, it can be denoted as the difference between the number of strategies and the number of unique objective points, which are the different values that are not equivalent in fitness. Non-unique points are important for several reasons:

- **Operational flexibility:** Decision-makers benefit from a wider range of options leading to more informed and confident decision-making.
- **Insight into algorithm behavior:** Analyzing non-unique points helps in understanding how the algorithm explores the solution space. It indicates the algorithm's ability to find diverse solutions and underscores the exploration-exploitation balance.
- **Adaptability:** Different operational scenarios might require different strategies. Non-unique points ensure that there are alternative strategies available that can be quickly adapted to changing conditions or requirements.

For each EA, the total number of unique points, as well as the mean and maximum number of unique points per objective point, are shown.

10.4 Coverage

In this section, the coverage metric is defined and explained. Coverage is defined as the intersection of a set X with any subset Y:

$$\text{Coverage}(X, Y) = X \cap \bigcup_{y \in Y} y \tag{10.8}$$

Where X and Y are the can be a super, combined or seeds sets. A hierarchical one-to-many relationship exits between these sets. one super Pareto front, consisting of 6 (many) EA combined Pareto fronts. Each combined Pareto front consists of 100 (many) seed Pareto fronts. Using the coverage, there can be analyzed how many seed sets it takes to get a combined set. This is when $\text{Coverage}(X, Y) = X$ meaning, the set X and the intersection set are the same. Similarly, the performance of an EA can be analyzed by calculating the coverage of a combined set with that of the super set. The larger the resulting intersection set $\text{Coverage}(X, Y)$, the better it is able to find the best solutions. Figure 10.4 shows the hierarchical one-to-many relationship between seed, combined, and super sets within the solution/feature space.

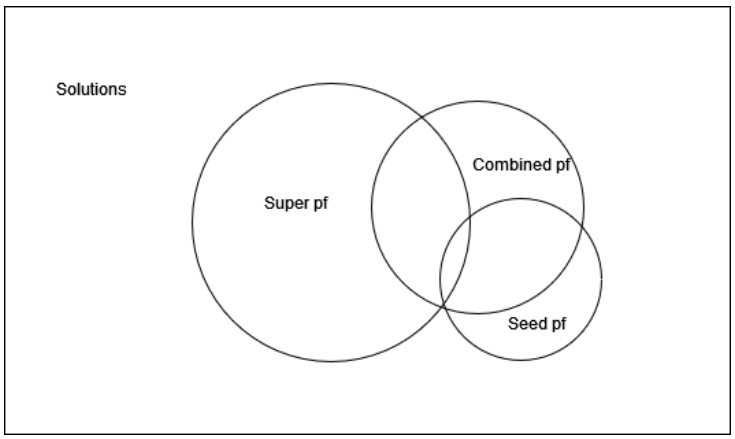


Figure 10.4: Hierarchical one-to-many relationship between seed, combined, and super sets within the solution/feature space.

The coverage metric is used in two analyses; the first analysis concerns the relationship between the final seed sets and the combined set. Due to randomness, the final results between seed runs may vary. A specific run can contain different strategies compared to a run executed with a different seed setting. To analyze this behavior, two seed analysis methods are used. The first method examines the distribution of how many runs need to be combined to form the combined set. The second method analyzes the relationship between the number of combined seed runs and the number of points and strategies. These methods are explained in more detail in subsections 10.4.1 and 10.4.2, respectively.

The second analysis focuses on coverage within the common objective space of the super-Pareto front with the combined sets. Although IGDplus and hypervolume metrics provide a clear indication of how close the solutions of a combined EA are to the super Pareto front, they do not offer insight into the number of unique points contributed by each algorithm. In this analysis, the aim is to determine which algorithm contributed the most strategies to the super Pareto front. The more an EA contributes, the better it performs.

10.4.1 Seed Set Analysis for Combined Solution Set Coverage

In the first method, the number of different seed sets required to match the number of combined solutions is analyzed. This is done by conducting 1,000 experiments, where for each experiment, sets of different seeds are randomly selected and incrementally combined until the resulting solutions in the combined solution set equals the solutions in the combined set. The number of sets needed to achieve this equality is recorded. This process yields a probability distribution of the number of seeds required to achieve the combined set. This probability distribution helps determine the confidence level that all solutions have been found given the number of seeds combined. A detailed description of the algorithm can be found in Algorithm 4.

Algorithm 4 Seed set analysis for combined solution set coverage

Input: Final solution sets of random seeds S, number of experiments $n = 1000$
Output: Probability distribution of the number of seed sets needed to achieve the combined set
num_sets_needed $\leftarrow$ []
for experiment from 1 to n **do**
    $S' \leftarrow S$
    $C_{\text{current}} \leftarrow \emptyset$
    while Coverage(C_{current}, T) $\neq C$ **do**
        $T \leftarrow$ Randomly select a subset from S'
        $S' \leftarrow S' \setminus \{T\}$
        $C_{\text{current}} \leftarrow C_{\text{current}} \cup T$
    end while
    Append number of sets in C_{current} to num_sets_needed
end for
return num_sets_needed

10.4.2 Incremental Seed Aggregation and its Effects on Strategies and Points found

In the second analysis, the effect of incrementally combining more seeds on the number of strategies and points found is investigated. Initially, an experiment consisting of 100 runs is performed, each with a random seed. The number of points and strategies included in the combined set is recorded. The experiment is then repeated by combining the results of two seeds, then three seeds, and so forth, until all seeds are combined. For each step, the mean and standard deviation of the number of points and strategies found across the 100 runs are calculated. The results demonstrate how the algorithm converges in terms of both the number of strategies and the number of points found. Algorithm 5 provides a detailed description of how this analysis works.

Algorithm 5 Incremental Combination Analysis

Input: Set S of all EA seed Pareto fronts, number of runs $n = 100$
Output: Mean and standard deviation of the number of strategies found at each step
mean_strategies ← []
std_dev_strategies ← []
for k from 1 to $|S|$ **do**
    strategies_found ← []
    for run from 1 to n **do**
        selected_seeds ← Randomly select k seeds from S
        combined_result ← $\bigcup_{s \in selected\_seeds} s$
        Append number of strategies in combined_result to strategies_found
    end for
    mean_strategies_k ← Mean(strategies_found)
    std_dev_strategies_k ← StandardDeviation(strategies_found)
    Append mean_strategies_k to mean_strategies
    Append std_dev_strategies_k to std_dev_strategies
end for
return mean_strategies, std_dev_strategies

10.5 Topological Diversity

In this section, methods for analyzing topological and substation diversity are discussed. The following analyses are performed.

1. The final topology result is displayed using sankey diagram.
2. The topologies used most often are identified and the distribution of topology count is given and analyzed.
3. The dominance of topologies per timestamp is assessed.
4. The different substations used is identified.

These analyses are conducted within their own optimized objective (target) space and a common objective space.

10.5.1 Final Topology Result

The final topology result are represented by a Sankey diagram, where the flow of strategies is visualized over the different timestamps. In this diagram, nodes represent different topologies, and the width of the arrows between nodes is proportional to the number of strategies that transition from one topology to another. This representation clearly shows the dominant topologies and the flow of strategies between them, illustrating the algorithm's exploration and utilization of different configurations.

10.5.2 Distribution of Topologies

This analysis investigates which distributions are predominantly used in the Pareto front of the combined EA. This is done in the following way. Let V be the set of all topologies. Let $I_{i,j}$ be an indicator function that represents that topology j is present in strategy i. Let C_j be the count of the Topology $j \in V$. The count of any topology j is calculated as follows (10.9):

$$C_j = \sum_{i \in Z} I_{i,j} \tag{10.9}$$

For each topoly $j \in V$, the count is calculated. Then, the counts are sorted, based on their frequency. This gives an indication of what the most important topologies are and how each EA favors different topologies.

10.5.3 Topology Dominance per Timestamp

The Simpson Diversity Index (SDI) is a measure used in ecology to quantify the biodiversity of a habitat [36]. It accounts for both the number of species present (species richness) and the relative abundance of each species (species evenness). In addition to its application in nature, this metric can also be used to determine how rich and diverse the number of topologies is in a certain timestamp are. The value of the Simpson diversity index ranges from 0 to 1. A value closer to 1 indicates a rich and balanced distribution of topologies, whereas a value closer to 0 indicates a unrich and unbalanced distribution of topologies.

The Simpson Diversity Index is calculated using the following formula:

$$D_t = 1 - \left(\sum_{i=1}^{S_t} \frac{n_{i,t}(n_{i,t} - 1)}{N_t(N_t - 1)} \right) \tag{10.10}$$

Where:

- D_t is the Simpson Diversity Index at timestamp $t \in T$.
- S_t is the total number of distinct topologies at timestamp $t \in T$.
- $n_{i,t}$ is the number of topologies of type i at timestamp $t \in T$.
- N_t is the total number of topologies at timestamp $t \in T$.

Additionally, the total difference between the SDI of a combined population and that of the super population is calculated to determine how much the topology distribution of the combined population deviates from the ideal set of solutions. Let, t_1 and t_2, be the SDI of two time frames the total difference is calculated using the following formula (**??**):

$$\mathbf{\Delta D} = \sum_{t \in T} |D1_t - D2_t| \tag{10.11}$$

Where:

- $\mathbf{\Delta D}$: is the total difference in SDI
- $\mathbf{D1}, \mathbf{D2}$:SDI of **D1** and **D2** in space $\mathbf{R}^{|T|}$

10.5.4 Substation Diversity

In addition to analyzing the topologies used, it is also important to analyse the substations activated withing each strategy. Some topologies require specific substations to be activated. To analyze substation diversity, a frequency diagram is created. This diagram will show the fraction of each substation's usage across all topologies.

10.6 Computation Time

Finally, an analysis of the computation time is performed. EAs that optimize for multiple objectives typically require more computation time. This increased computation time can affect the practicality of using an EA, especially in situations where rapid decision-making is crucial. The most computationally expensive parts of the algorithms described in Section 8, are the selection algorithms NSGA-ii and NSGA-iii and the evaluation method. NSGA-ii and NSGA-iii both have a time complexity of $\mathcal{O}(MN^2)$, where M is the number of objectives and N is the population size. This complexity arises due to the non-dominated sorting procedure, which dominates the computational effort of the algorithms [18]. This is the time complexity for running one generation. Figure 10.5, illustrates how the time complexity scales for any N and M.

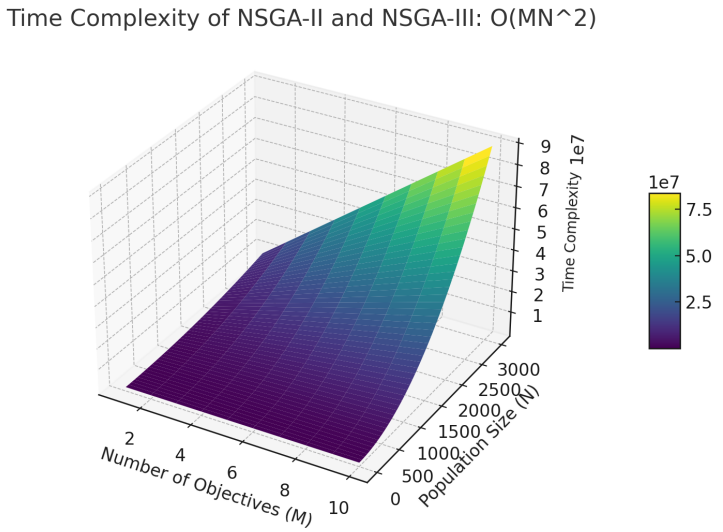


Figure 10.5: Time complexity of $\mathcal{O}(MN^2)$.

For each EA, the time it takes to finish a run, which takes 700 generations, is recorded. Then the mean of all 100 different seed runs is taken. Using this mean, the runtime of 1 generation can be calculated. Based on our seed analysis and termination criteria(number of generations), it can be calculated how long it would take for an EA to find all the points and strategies.

Chapter 11

Results

In this chapter the result from our evaluation measures are presented. First the final results projected in each solution space is given, then the results of the new initialisation method proposed in Section 8.3 are shown and finally the results of each evalutaion measure defined in Chapter 10 for each EA is given.

11.1 Final Solutions

For EA, a scatter plot (Figure 11.1) between $N - 1$ and the other objectives of the combined Pareto fronts is presented.

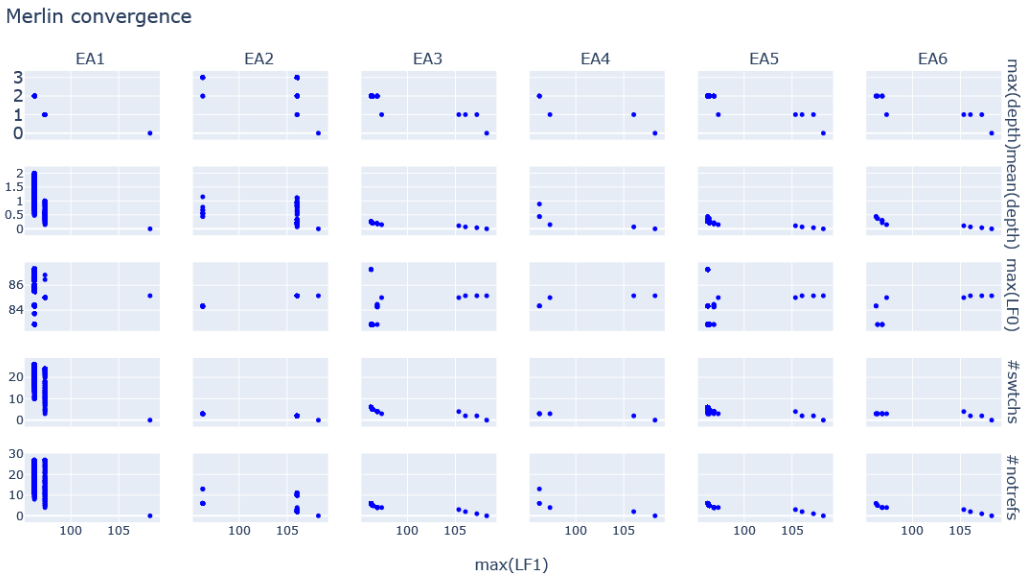


Figure 11.1: Final populations of each EA projected in each objective space.

Figure 11.1 gives the following first insights:

- EA1 has many solutions outside its target space. However, most of these points are sub

optimal.

- EA2 is not able to find all the Pareto optimal points for max depth, suggesting a limitation in its optimization capability for this objective.
- EA5 and EA6 appear to find similar solutions, indicating that the optimization criteria for these two EAs lead to comparable results.
- EA3 performs considerably better than EA1 and EA2, which are optimized on two objectives. It finds similar solutions to EA5 and EA6, indicating strong performance across multiple objectives.

11.2 Initial Population Results

In this section, the result of the selection method proposed in Section 8.3 is given. EA4 optimizes switches and maximum depth. This algorithm is run twice, once using random initialisation and once using the proposed initialisation method defined in Section 8.3. For both runs, the same hyperparameters are used. The first and final population are plotted to see the difference between the initialization methods. The initial populations can be seen in Figure 11.2 and the final population in Figure 11.3. Figure 11.2 shows that for random initialisation the values for depth and switches are 3 and 26 respectively. In contrast, our custom method gives a better spread across the solution space for both objectives.

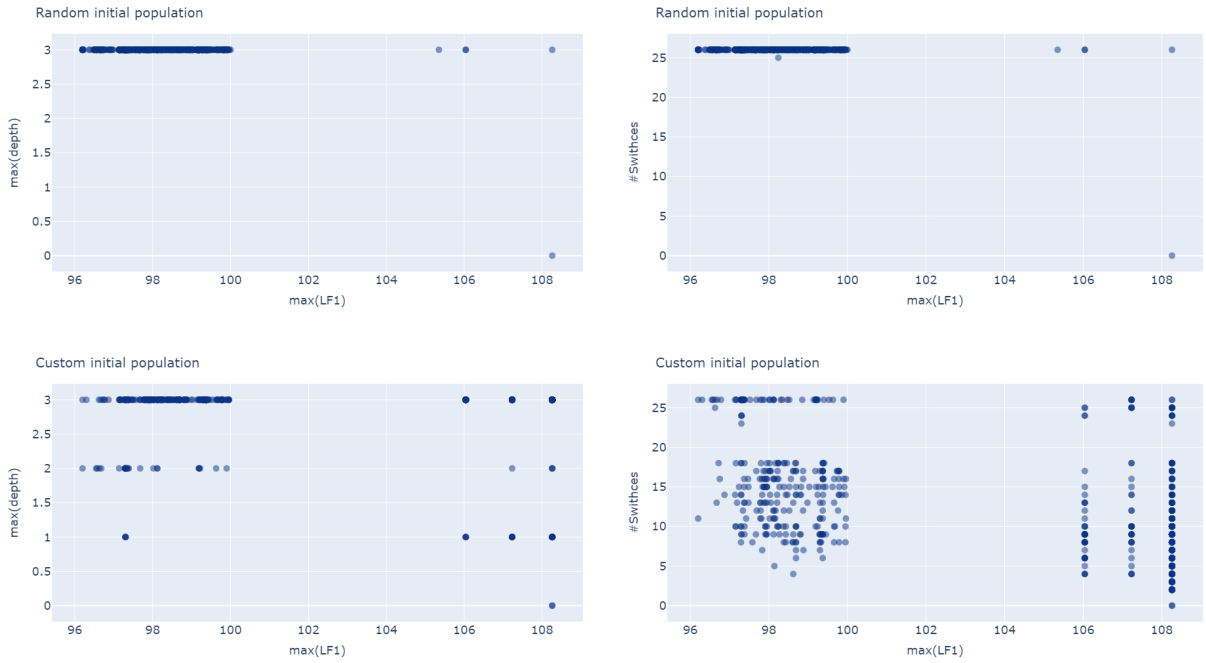


Figure 11.2: Results of initial population using random initialisation (top row) and custom initialisation (bottom row) for max depth and switches.

EA4 using the custom initialisation method method found 7 different objective values containing

41 strategies for both depth and switches, the randomly initialized population only found 5 objective values containing 17 strategies. EA4 also contains better solutions. The depth plot contains the solutions (97,3;1) instead of (106;1) and (96.1;3) instead of (96,9; 3) in switches plot. This shows the importance of having a diverse initial population and how it can affect the final result.

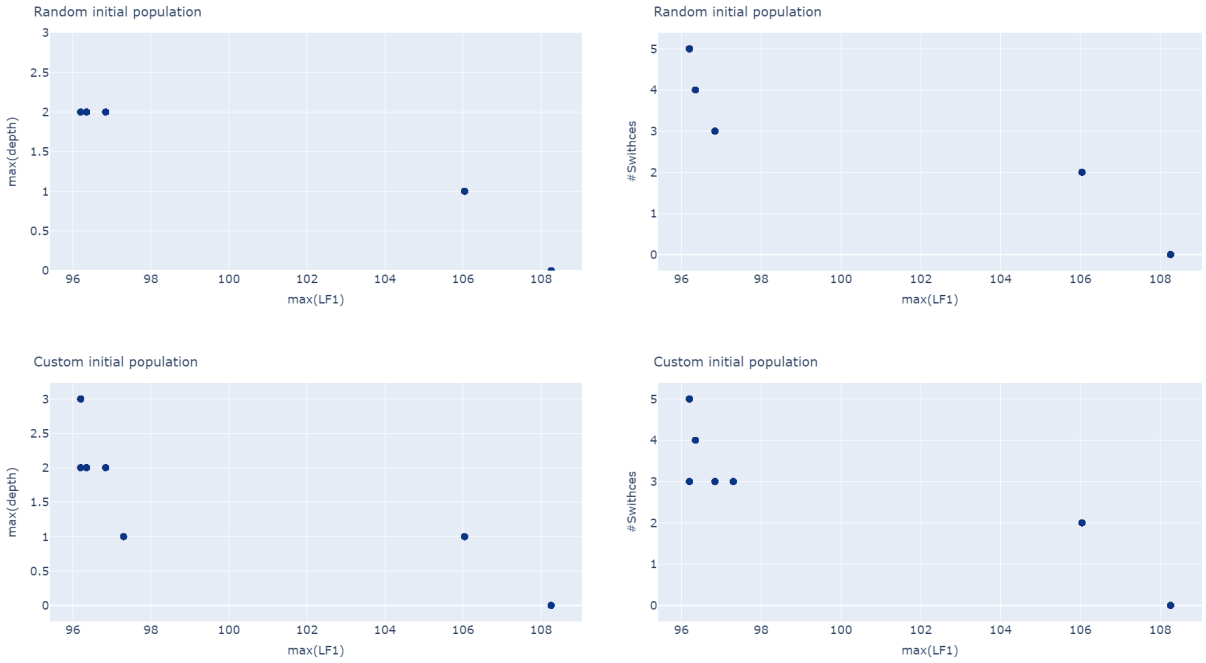


Figure 11.3: Results of final population using random initialisation (top row) and custom initialisation (bottom row) for max depth and switches.

11.3 Hypervolume and IGDplus

In this section, the Hypervolume and IGDplus results are presented. The implementation details can be found in subsection 10.1.1. The result is given for both the target space and the common space, respectively.

11.3.1 Results Target Space

In Figure 11.4, the hypervolume for each EA in the target space is shown. Every EA shows convergence. After 100 generations, most EAs appear to have converged; however, there are still some small improvements that are made. After 400 generations, all EAs appear to have converged. This indicates that running the algorithm for more generations will likely not result in better solutions. It is important to note that differences in the values to which the EAs converge do not imply that one EA outperforms the others. The maximum hypervolume that can be achieved in each solution is different.

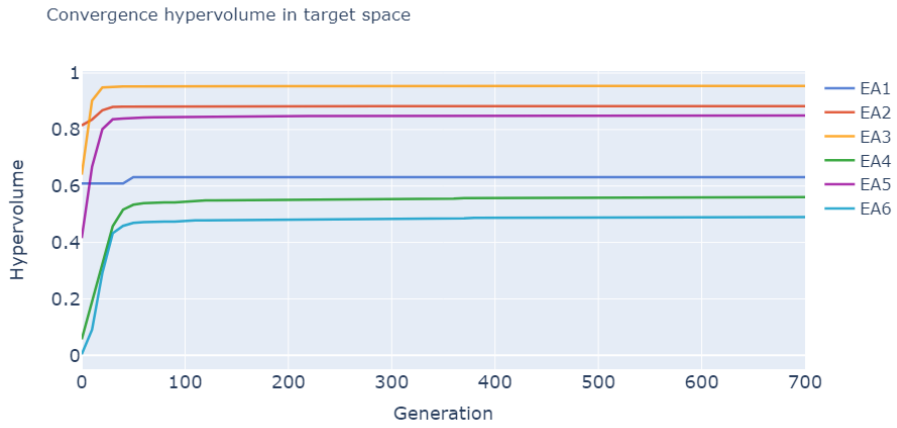


Figure 11.4: Convergence hypervolume in target space of the 6EAs.

In Figure 11.5, the IGDplus for each EA in the target space is shown. EA3 and EA5, the algorithms optimized on mean depth, achieve a 0 IDG+ distance, indicating the best results compared to the combined Pareto front. These algorithms also find the best solutions rapidly, with EA3 converging after 50 generations and EA5 after 300 generations. Furthermore, EA1 and EA2, which are optimized for two objectives, perform the worst compared to the super front. Other EAs find better results within their respective target spaces, outperforming EA1 and EA2 in their target spaces.

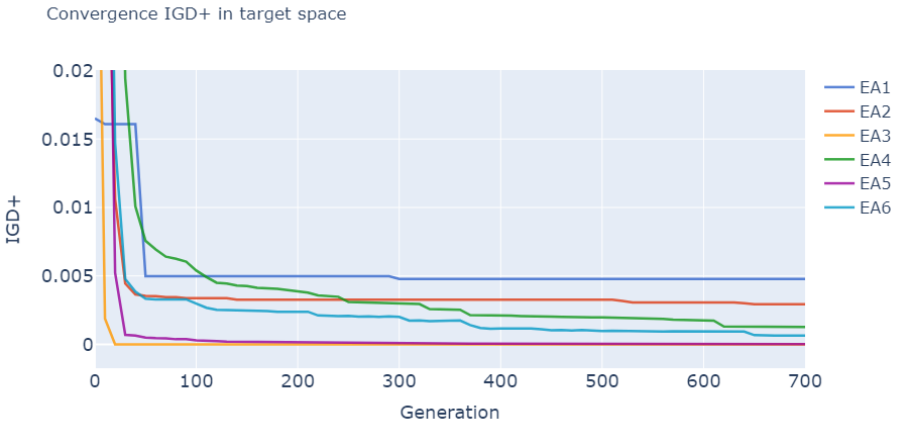


Figure 11.5: Convergence IGDplus in target space of the 6EAs.

11.3.2 Results Common Space

In Figure 11.6, the hypervolume for each EA in the common space is shown. EA1 and EA2 perform poorly in the common objectives space, as expected, since they only optimize two out of the four objectives in this space. The rest of the EAs achieve a hypervolume around 0.5 after 700 generations. Remarkably, EA3, which is optimized for two objectives achieves a similar

hypervolume to that of EA5 and EA6.

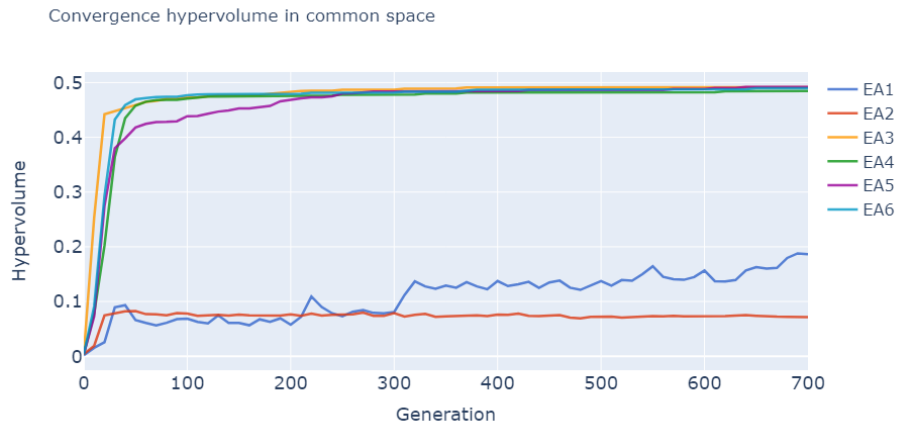


Figure 11.6: Convergence hypervolume in common space of the 6EAs.

In Figure 11.7, the IGDplus for each EA in the common space is shown. This graph tells a similar story as the hypervolume Figure 11.6. The standard deviations for each EA are also plotted. The path of EA1 appears random, indicating inconsistent performance and difficulty in optimizing the common objectives.

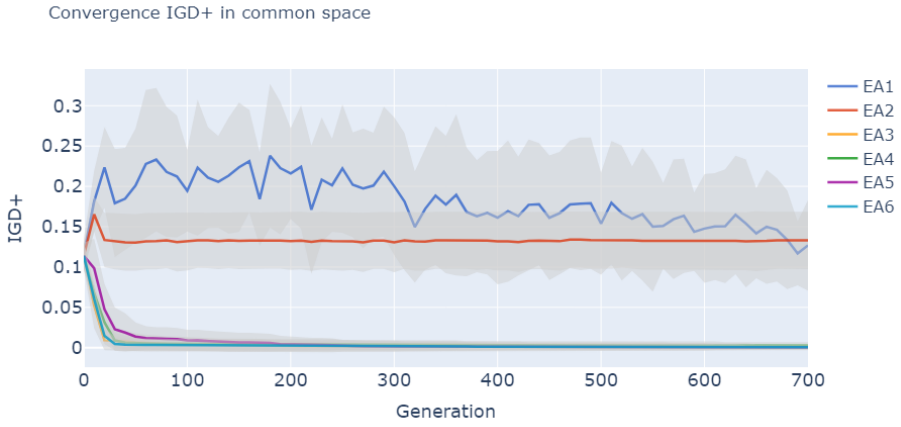


Figure 11.7: Convergence IGDplus in common space of the 6EAs.

11.4 Coverage Combined Population

In this section, coverage is looked at. First, we look at the results of our seed analysis methods to see how many runs are required to obtain all the optimal points and strategies, as explained in Section 9.2.1. Then the results are given for the coverage between the combined and super Pareto front as explained in Section 9.2.1.

11.4.1 Target Space

In this subsection, the results of coverage on the target space are shown. Figure 11.8 illustrates the convergence of the number of nondominated points within the target space.

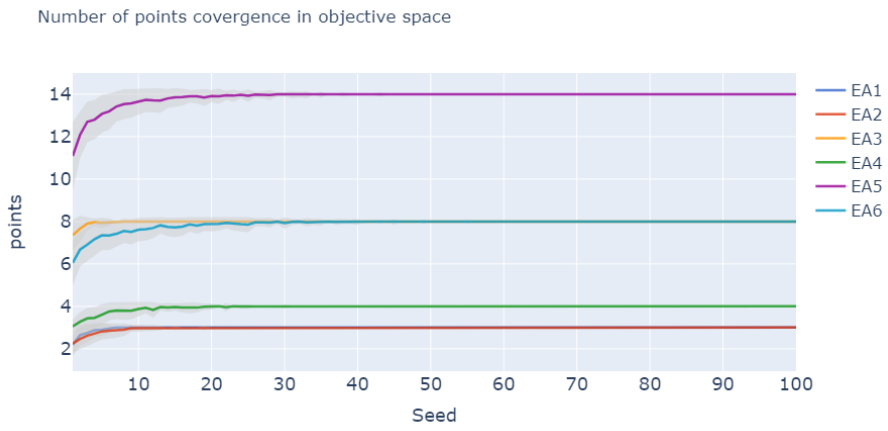


Figure 11.8: Convergence of the number of points in target space (=objective space) of the 6EAs.

It can be observed that each EA converges to a certain number of nondominated solutions. A combined seed set of 30 is sufficient for all algorithms to find all the nondominated solutions. Figure 11.9 below illustrates the convergence of the number of strategies. EA1, EA2, EA3, and EA5 do not converge. Only the EAs that have converged, EA4 and EA6, are shown in Figure 11.9 for scaling reasons, as the non-converged EAs continue to increase in the number of strategies. The results for the non-converged EAs can be found in the Appendix 14.2 .

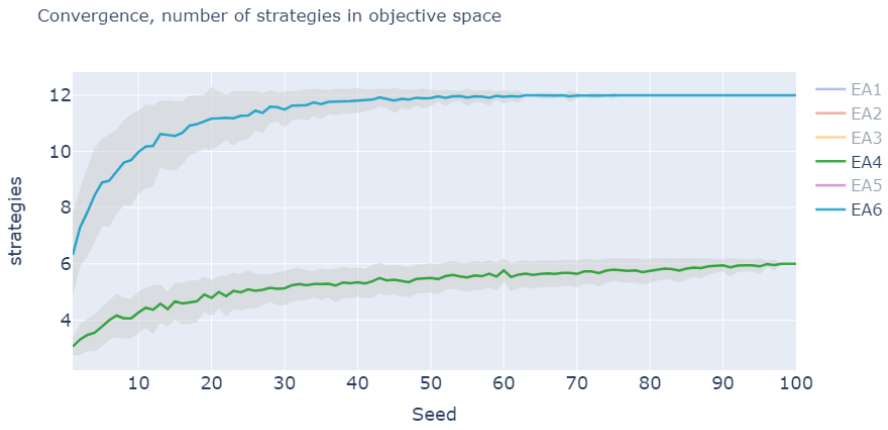


Figure 11.9: Convergence of the number of strategies in target space (=objective space) of the 6EAs.

EA4 converges slowly to 6 strategies. Almost all the seed results have to be combined. EA6 converges much faster. This coverag analysise in the target space shows that EA6 is able to

find 8 non-dominated containing 12 strategies. EA4 finds 4 non-dominated soluitions containing 6 strategies. The other EAs are keep finding more strategies as the number of seed sets are combined. In Appendix 9.2.1 all the EAs are plotted.

11.4.2 Common Space

In this subsection the results of coverage on the common space is shown. Figure 11.10 illustrates the convergence of the number of non-dominated points within the common space and Figure 11.11 represents the convergence of the number of strategies in the common space of the 6EAs.

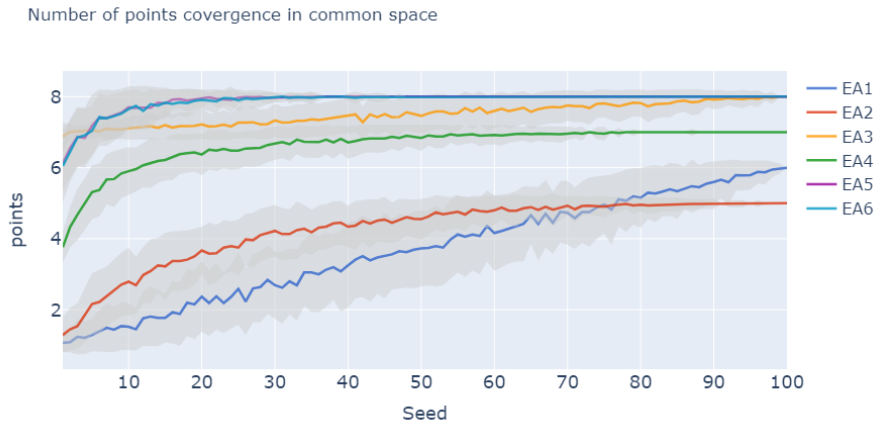


Figure 11.10: Convergence of the number of points in common space of the 6EAs.

Figure 11.10 shows that EA6, EA5, EA4 and EA2 have converged to a certain number of nondominated points. EA5 and EA6 have almost the exact same convergence rate. It seems that EA3 is almost coverged. Adding more seed runs together will likely show EA3 converging. In contrast, EA1 does not seem to increase, having an linear trend and high standard deviation.

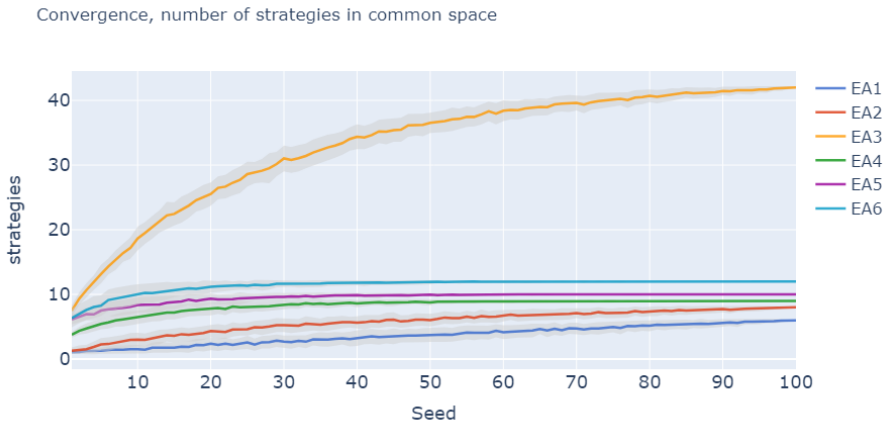


Figure 11.11: Convergence of the number of strategies in common space of the 6EAs.

Figure 11.11 shows that EA3 is able to find way more strategies, than the rest of the algorithm. It seems that, just like the number of points, it is not fully converged yet. Additional seeds might help in finding more solutions. To see the behaviour of the other EAs, the EA3 is left out in Figure 11.12.

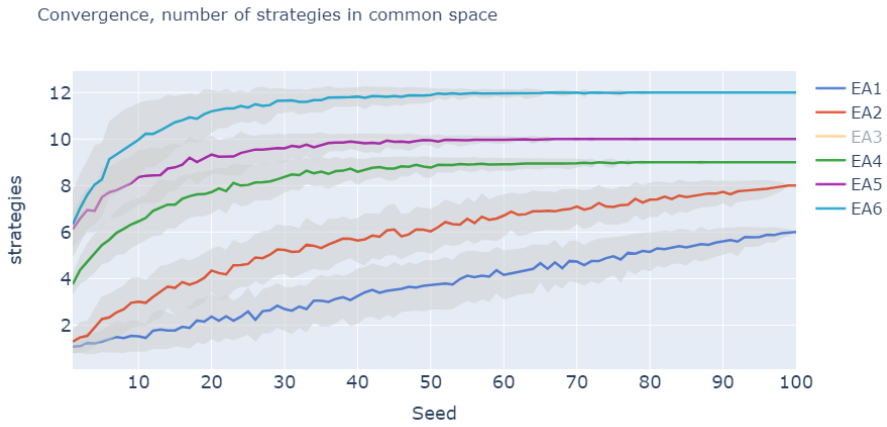


Figure 11.12: Convergence of the number of strategies in common space of the 5EAs leaving out EA3.

There can be seen that EA6, EA5 and EA4, have fully converged, while EA2 and EA2 have not yet. EA6 and EA5 are able to find the same nunber of points, but EA6 is able to find more strategies.

11.4.3 Density Function

In this section, the results of the Combined Solution Set Coverage analysis(CSSC) (10.4.1) are presented. Only the probability distributions of the EAs that have converged, as discussed in 11.4.2, are investigated. Figure 11.13 shows the results of running the CSSC analysis for the number of strategies of EA6.

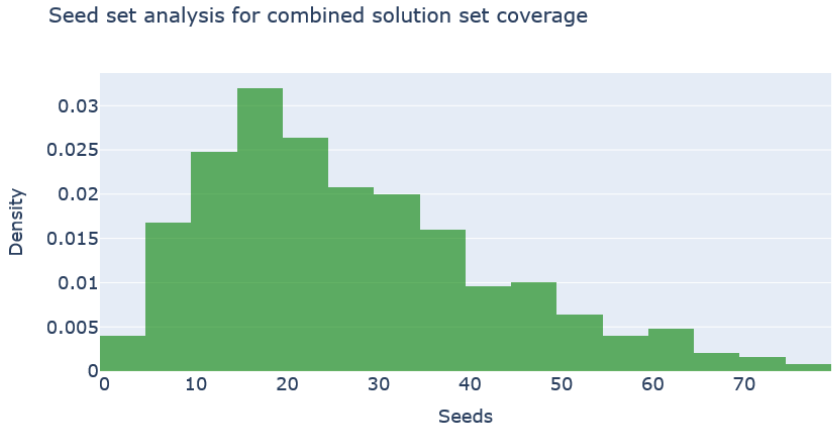


Figure 11.13: Results of the CSSC analysis on the number of strategies of EA6

The distribution looks resembles a gamma distribution. To test this hypothesis, the Kolmogorov-Smirnov (KS) test is used. The KS test compares the sample distribution to a theoretical distribution, measuring the maximum distance between their cumulative distribution functions. A small KS statistic and a high p-value indicate that the sample distribution is similar to the gamma distribution [20]. This test has been used on each EA, and confirms that the data is gamma distributed. In Figure 11.14, the fitted gamma distributions of EA4 EA5 and EA6 are shown.

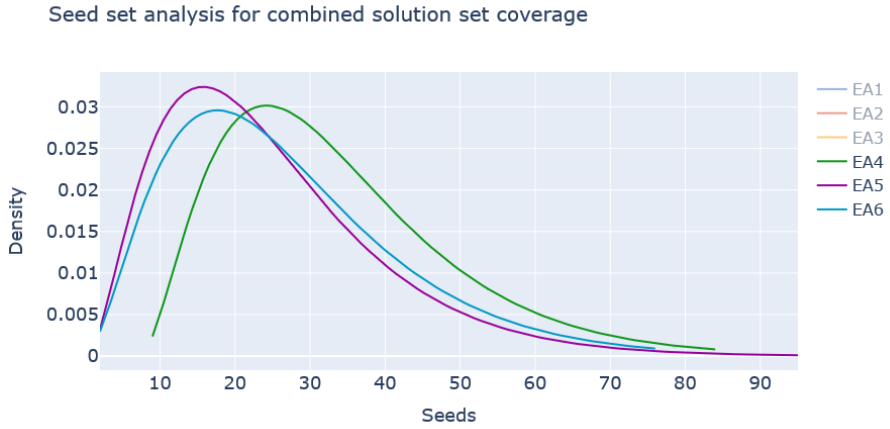


Figure 11.14: Gamma distribution fit of EA4, EA5, and EA6.

The CSSC analysis is also performed for the number of points. However, the results for EA4, EA5, and EA6 do not follow a particular distribution. The results are plotted in Appendix 14.4. Table 11.1 provides the 95th and 99th quantiles for EA4, EA5, and EA6 based on a given number of seed runs. These quantiles offer insight into how many seed runs need to be combined to achieve 95% and 99% certainty in identifying all strategies and points. For strategies, the quantiles are based on the gamma distributions, while for points, they are derived directly from the data.

Table 11.1: 95th and 99th Confidence Results from CSSC Analysis

Algorithms	**EA4**	**EA5**	**EA6**
Strategies			
Parameters	(2.77, 7.37, 9.47)	(2.71, 0.39, 8.97)	(3.12, -0.35, 9.09)
95th Quantile	64	53	59
99th Quantile	83	72	78
Points			
95th Quantile	63	23	27
99th Quantile	78	33	36

11.5 Coverage Super Pareto Front

This section evaluates the performance of the EAs against the super Pareto front. Table 11.2 presents the data points and their occurrence in the super Pareto front (SP) of each EA. Table 11.3 summarizes the coverage of strategies and points for each algorithm. The objectives in the points columns are arranged as follows: $\max(N-1)$, max(depth), number of switches, and number of not reference topo.

Table 11.2: Coverage of each EA with the super Pareto front.

Index	Points	SP	EA1	EA2	EA3	EA4	EA5	EA6
1	(92.20; 2; 3; 6)	2		1		2	2	2
2	(96.35; 2; 3; 3)	2					2	2
3	(96.84; 2; 3; 4)	3					1	3
4	(97.30; 1; 3; 4)	1	1		1	1	1	1
5	(105.35; 1; 4; 3)	1			1		1	1
6	(106.04; 1; 2; 2)	1		1	1	1	1	1
7	(107.23; 1; 2; 1)	1			1		1	1
8	(108.26; 0; 0; 0)	1	1	1	1	1	1	1

Table 11.3: Coverage fraction of strategies and points with super Pareto front.

	EA1	EA2	EA3	EA4	EA5	EA6
Strategies	0.17	0.25	0.42	0.42	0.83	1
Points	0.25	0.375	0.625	0.5	1	1

- EA6 demonstrates the highest effectiveness, achieving 100% coverage of both strategies and points in the super Pareto front. EA5 shows comparable performance to EA6, covering 83% of the strategies and 100% of the points.
- EA5 achieves twice the coverage of EA4, suggesting that optimizing for mean topological depth is considerably more effective than focusing on max topological depth. This is further concluded by EA3, which also outperforms EA4 in number of points found.

- Certain points appear more frequently in the solution sets of the algorithms. For example, point 8 is present in the solution sets of all EAs. Points 2 and 3 only appear in the solution sets of EA5 and EA6, indicating that these solutions are harder to find.

11.6 Non-Uniqueness

In this section, the non-uniqueness results are shown in Table 11.4.

Table 11.4: Non-uniqueness of every EA in common objective space and target space.

	Converged	Points	Strategies	Non-unique	CP Points	CP Strategies	CP Non-unique
EA1	no	3	24836	24833	6	6	0
EA2	no	3	117	114	5	8	3
EA3	no	8	86	78	8	42	34
EA4	yes	4	6	2	7	9	2
EA5	no*	14	265	251	8	10	2
EA6	yes	8	12	4	8	12	4

From Table 11.4 the following insights can be derived:

- EA1 has found a significantly higher number of strategies compared to the other EAs. Optimizing only for maximum depth is too relaxed, leading to an excessive number of strategies, many of which are sub-optimal.
- EA3 and EA5, which are optimized for mean depth, can identify more non-unique solutions compared to the other algorithms in the target space. This is because mean depth can take any positive real number, offering a continuous range of potential solutions. In contrast, the objective of max depth is limited to discrete values up to three, resulting in fewer possible values.
- EA3 has 42 strategies in the common space, however Table 11.2 shows that only 5 strategies are part of the super front.

11.7 Topology Diversity

This section presents the results of the analysis of topological diversity. Topological diversity is explained in more detail in Section 10.5. First, the results of topology dominance using the Simpson Diversity Index are provided. Then, the five most commonly used topologies are highlighted. Finally, the distribution of topologies and substations is shown.

11.7.1 Simpson Diversity Index

In Figure 11.15 the total difference of the Simpson Diversity Index between the different EAs and the super solution is shown.

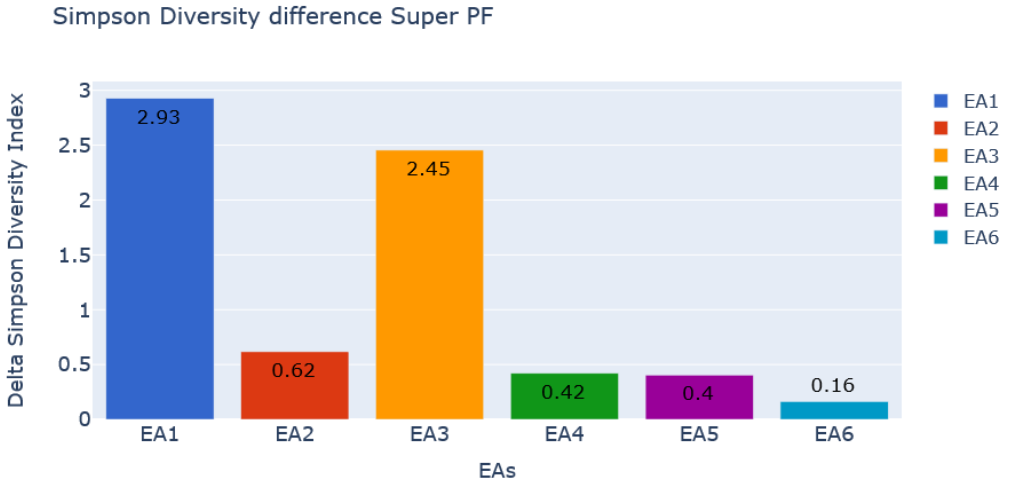


Figure 11.15: Difference of Simpson Diversity index with super Pareto front and the 6EAs.

From Figure 11.15 the following insights can be derived:

- The difference between EA4, EA5 and EA6 is low, with EA6 the lowest. This means that that the topology richness and evenness is almost the same as the super Pareto front.
- EA3 shows a high deviation from the super Pareto front, which is interesting since it performs similarly to EA3 in terms of Hypervolume and coverage. This suggests that EA3 employs more topologies than the other EAs at certain timesteps in its strategies.

Figure 11.16 shows the SDI of each EA and the SDI of the super Pareto front. EA1 and EA2 are left out from the graphs because of readability reasons. In Appendix 14.1, the SDI of all EAs are plotted.

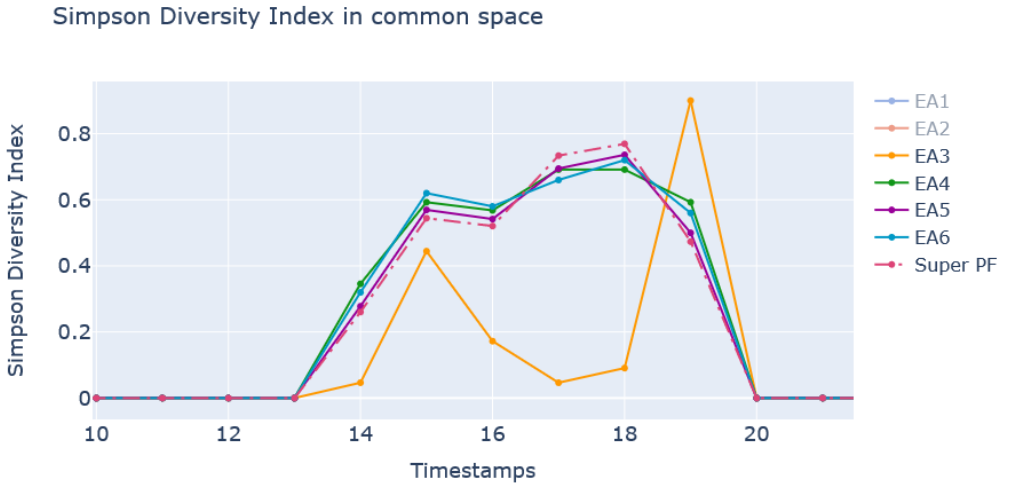


Figure 11.16: Simpson diversity index of EA3, EA4, EA5, EA6, and the super Pareto front.

From Figure 11.16 the following insights were derived:

- The SDI is high between time step 13 and 20, meaning that most of the switching takes place between these intervals. The timeframe of congestion identified in Section 9.2.1 aslo lies with in that timeframe.
- In contrast, the SDI values are low from time step 0 to 13 and from 20 to 27, indicating that strategies in the combined population use the same topology during these time periods.
- The SDI for EA4, EA5, and EA6 is almost identical, indicating that performance differences are the result of differnt topology choices or their interconnections.

11.7.2 Frequency Distribution

Figure 11.17 shows the results of the 5 most frequently used topologies.

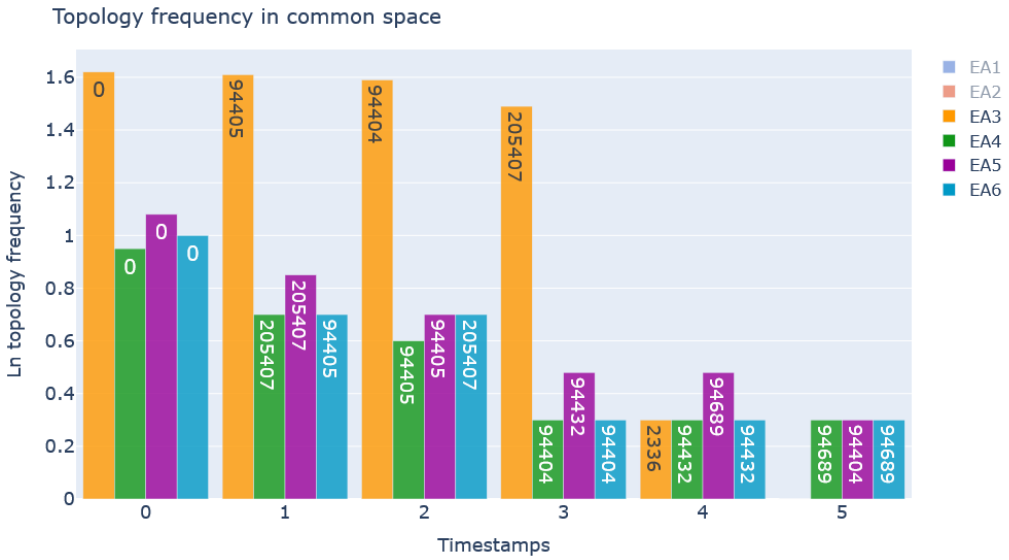


Figure 11.17: Natural logarithm (Ln) frequency of the top 5 topologies used for EA3, EA4, EA5, and EA6.

From Figure 11.17 the following insights were derived:

- Topology 0 is the most frequently used in each EA, and topology 94405 is in second or third place.
- There is overlap in the frequently used topologies This overlap suggests a convergence in the search space of these EAs, where they tend to find and exploit similar topological solutions despite the different objectives they have been optimized on.
- EA3 uses the most frequently used topologies more than the other Eas. Furthermore EA3 uses topology 2336 and the other do not.

11.7.3 Topology Distribution

This subsection highlights only the final topological results of EA5 and EA6 in Figure 11.18 and Figure 11.19, as EA5 and EA6 are the best-performing algorithms. The topological distributions of the other EAs can be seen in the Appendix 14.4.

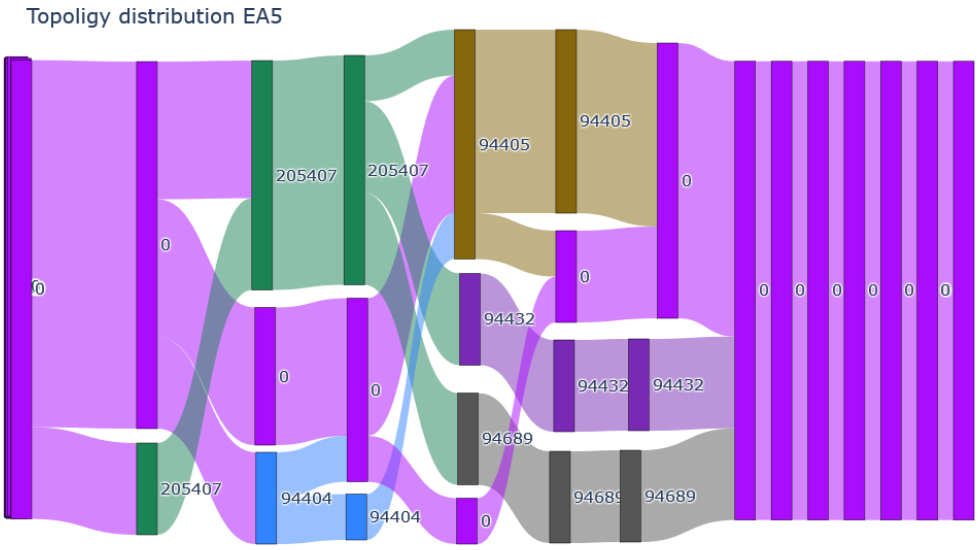


Figure 11.18: Topology distribution of EA5.

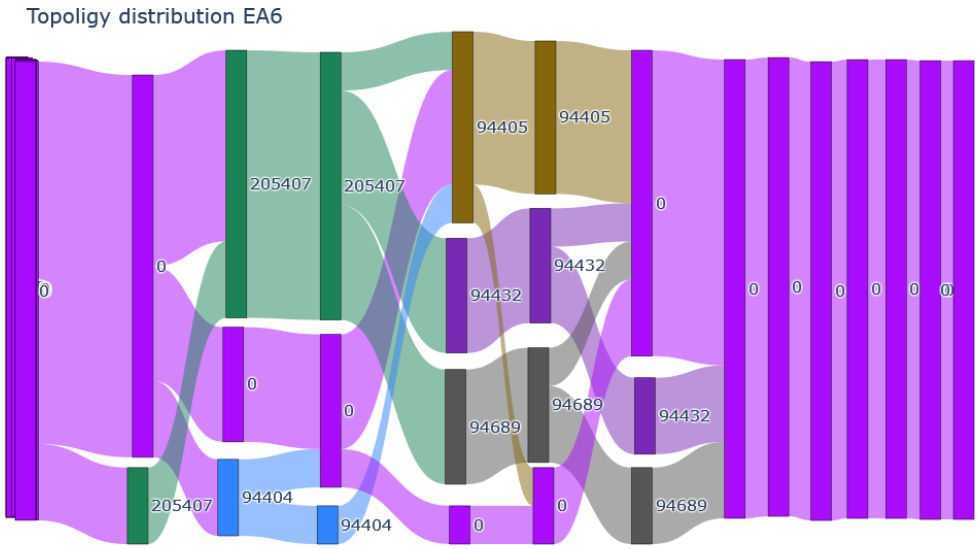


Figure 11.19: Topology distribution of EA6.

From Figure 11.18 and Figure 11.19 it is observed that the topologies used at each timestamp are identical, as indicated by the SDI values which show the same distributions and evenness across the timesteps 11.16. The frequency plot 11.17 further corroborates this, demonstrating

that the same top 5 strategies are utilized. Therefore, the difference in strategies found can only be attributed to the interconnections of these topologies This difference is highlighted by the extra flows present in Figure 11.19 of EA6 that were not observed in EA5 11.18. One flow transitions from topology 94689 (gray) to topology 0 (purple). The other flow transitions from topology 94432 (dark purple) to topology 0 (purple). These additional flows account for the two extra strategies that EA6 is able to identify, as mentioned in Section 11.5.

In Figure 11.20, the substation diversity is shown for both EA5 and EA6. Since EA5 and EA6 use the same topologies at each timestamp, the substation diversity is also identical.

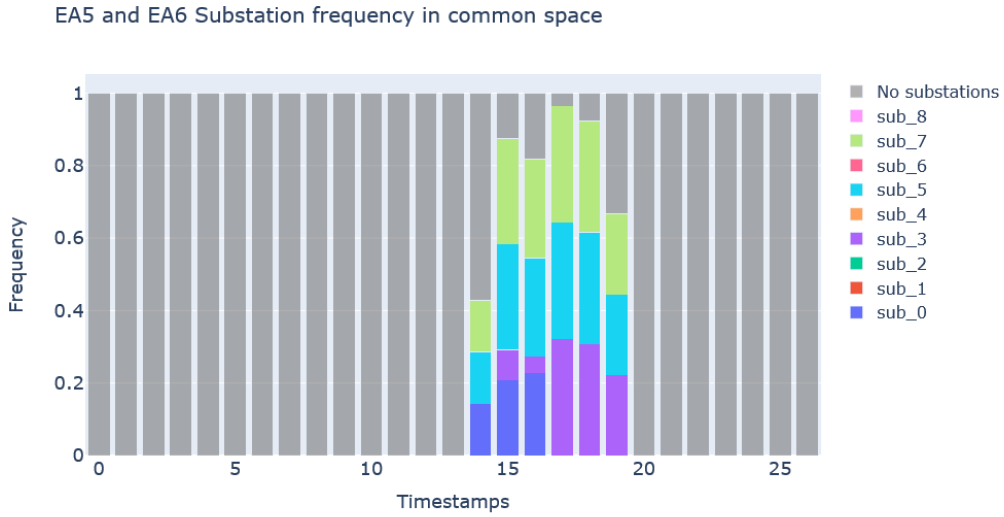


Figure 11.20: Substation diversity of EA6

11.8 Computation Time

In this section, the computation times for each EA are given. Table 11.5 below summarizes the computation times using the hyperparameters detailed in Section 9.4.

Table 11.5: Runtime statistics for each EA

EA	Total in min	Avg per seed in min	Per generation in milliseconds
EA1	126	1.26	108
EA2	129	1.29	111
EA3	137	1.37	117
EA4	158	1.58	135
EA5	156	1.56	134
EA6	216	2.16	185

Table 11.5 shows that increasing the number of objectives also increases computation time as expected. For EA6, almost 1 hour extra is needed. However, the computation time can be reduced by decreasing the number of generations and the number of seed runs. In Section 11.3,

the hypervolume results indicated that only 400 generations are needed for each EA to converge. Furthermore, based on the results of our CSSC analysis in subsection 11.4.3, the number of seed runs required to find all the points or strategies can also be reduced. Table 11.6 reflects the total computation time for EA5 and EA6 for each confidence level. It can be seen that the computation time for EA6 is longer than for EA5. It takes the EA6 1.5 times more time to find all the points and strategies than EA5.

Table 11.6: Computation times of EA5 and EA6 for 400 generations given the number of combined seeds from the CSSC analysis.

Algorithms	EA5	EA6
Strategies		
Computation time 95th CL	47.2	72.8
Computation time 99th CL	64.2	96.3
Points		
Computation time 95th CL	20.5	33.3
Computation time 99th CL	29.4	44.4

Chapter 12

Conclusion

In this research, we investigated how EA can be applied to a congestion management problem as defined in Section 6.4.2. The following research question was investigated:

> "How can a multi-objective evolutionary algorithm be applied to effectively optimize the congestion management problem?"

This question involves two sub questions:

1. How to apply an EA to a congestion management problem?
2. What EA method is most effective?

How to apply an EA to a congestion management problem?
We examined the essential components of an EA and adapted each for the specific requirements of the congestion management problem. Here, we discuss each component:

- **Chromosome formulation:** The length length equal to the number of timesteps where each gene is represented by a topology ID. This direct formulation allowed for an efficient mapping between solutions and their representations.
- **Initial population:** A new initialization method is developed to ensure diversity, as random initialization did not provide sufficient variety. This method ensures a broader exploration of the solution space from the outset.
- **Fitness function:** The fitness function is designed using the objectives defined in Section 6.4.2. An matrix multiplication method is proposed to efficiently calculate the fitness values for each member of the population.
- **Selection method:** NSGA-II and NSGA-III are chosen for optimizing problems with two and multiple objectives, respectively. These algorithms are selected for their effective use of the Pareto dominance principle and their proven success in similar applications within the electrical power field [24].
- **Operators:** K-point crossover and reset mutation are used as operators. These methods ensure that offspring remained valid solutions, respecting the uneven topology distribution of the dataset.

- **Stopping criteria:** The algorithms is run for 700 generations to allow for sufficient evolution and convergence of solutions. This can be reduced to 400 as all algorithms will have converged.

In addition to tailoring the components of an EA to the congestion management problem, we also investigated the number of runs required to capture all points and strategies effectively. The results included the 95th and 99th confidence intervals, providing insight into the robustness and reliability of the solutions.

What EA method is most effective?
To determine the most effective EA method, six different algorithms were run on various objectives. Subsection 9.1.1 provides the motivation for the selected algorithms. The performance of each algorithm was evaluated based on the following criteria:

- Hypervolume and IGDplus
- Coverage
- non-uniqueness
- Topology analysis

From the results in Chapter 11, it can be concluded that EA5 and EA6 are the algorithms that perform the best. The main difference between them lies in their performance and computation time. EA6 shows better performance, being able to find more strategies than EA5 in the super Pareto front (Section 11.5). However, EA5 has a faster computation time, 1.5 times faster than EA6 (Section 11.6). The determination of which algorithm is better depends on the priorities of the application. For this specific application, which involves creating strategies for the next day the night before, there is enough time to run the algorithm, which makes a computation time of 72 to 96 minutes acceptable. Therefore, it can be concluded that performance is more favorable than computation time, making EA6 the best algorithm.

Chapter 13

Discussion and Recommendations

13.1 Discussion

This research has provided insight into the application of EAs to optimize congestion management in power grids. However, several limitations and areas for further investigation should be considered.

One significant limitation is that the testing was conducted on a dataset for a single day. Although this provides initial insight, it may not capture the variability and different congestion characteristics that can occur on different days. Future research should extend the analysis to datasets from various days to ensure the robustness and generalizability of the findings. Moreover, the DC approximation (Section 6.2), which is used to determine load flows, is known to deviate from actual load flows [21]. This variability was not considered when evaluating solutions, which can affect the robustness of the results. Taking into account this variability in the optimization process could improve the reliability and robustness of the proposed strategies.

On top of that, load flow calculations are dependent on weather forecast data. In a comparative study, Rahman et al. (2024) evaluated six different machine learning models for predicting weather. The results showed that these models achieved 90% accuracy in predicting precipitation and temperature [31]. While this is a good performance for weather forecast models, the remaining 10% uncertainty can still significantly affect load flow results. Just as in the DC approximation, this uncertainty was not taken into account. Including this uncertainty can also improve the reliability of the proposed strategies.

Furthermore, the current objectives for evaluating grid management strategies may not fully capture all necessary aspects of grid system performance. This limitation indicates that the strategies suggested by our model might not be entirely reliable. Certain strategies might include topology sequences that increase the probaility for failure. An example of an objective that is not currently included in our model is the Loss of Load Probability (LOLP). LOLP measures the likelihood that the power system will not meet demand, which is a critical aspect of the reliability of the grid. As can be read in our Literature Review Section 5, Zhang et al. (2014) proposed a model that optimizes grid performance by incorporating LOLP, which demonstrated the importance of including such an grid reliability metric [41]. Integrating LOLP into our model

would lead to a more comprehensive assessment of grid stability and result in other strategies being proposed. The exclusion of LOLP and other metrics implies that our current model might overlook potential risks associated with certain topological configurations. This oversight can lead to strategies that seem optimal under the existing evaluation criteria, but pose a higher risk of grid failure. Therefore, improving our model to include LOLP and other reliability metrics is important to develop more robust and reliable grid management strategies.

In addition, the true Pareto front of the multi-objective optimization problem remains unknown. This study used NSGA-II and NSGA-III to approximate the Pareto front; however, the real optimal trade-offs between objectives may be different. Further research could involve benchmarking the results against other optimization techniques to further evaluate the effectiveness of the model.

Finally, in our analysis of the super Pareto front, we identified 8 objective points that consist of 12 different strategies (Table 11.2). From the frequency distribution plot, it was observed that the topologies most frequently used are consistent in each EA (see Figure 11.17). If one of these topologies becomes unavailable, the operator cannot use most of the proposed strategies. Future research should ensure diversity not only in the objective space, but also in the feature space. Several studies have addressed this issue. For example, Wineberg and Oppacher (2003) proposed a genotype diversity metric based on allele frequencies, this metric ensured the diversity different allel values [40]. Implementing such a metric could help prevent over-reliance on specific topologies and ensure that a variety of strategies remain viable even if some topologies become unusable.

.

13.2 Recommendations

13.2.1 TenneT

1. **Compare results with current tool:**
   It is recommended to compare the results obtained from the EA-based optimization with TenneT's current tool that uses brute force optimization. This comparison will help in evaluating the performance improvements, efficiency, and accuracy of the proposed method. Analyzing the differences in results can provide insights into the strengths and weaknesses of both approaches.

2. **Integrate into TenneT's GridOptions Tool:**
   To leverage the benefits of the EA-based optimization, it should be integrated into TenneT's GridOptions tool. This integration will enable the system operators to utilize the optimized strategies in real-time operations, thereby improving the decision-making process for congestion management. The enhanced tool will provide a more comprehensive and efficient solution for managing the electricity grid's dynamic topology.

3. **Use proposed evaluation measures:**
   The evaluation measures proposed in this research should be adopted to assess solutions from third-party companies. These measures include hypervolume, Inverted Generational

Distance Plus (IGDplus), coverage, and topological diversity. By applying these standardized metrics, TenneT can objectively evaluate and compare the effectiveness of various solutions offered by external vendors, ensuring that only the most robust and efficient strategies are implemented.

13.2.2 Future Research

Enhance model mobustness by incorporating uncertainty in load flow calculations: To enhance the reliability and robustness of the proposed congestion management strategies, it is recommended to incorporate uncertainties associated with both the DC approximation and weather forecasts into the optimization process. This can be achieved by integrating variability data and modeling the problem as a stochastic optimization problem rather than a deterministic one.

- Implementing robust optimization techniques that account for potential discrepancies in load flow calculations due to DC approximation errors and weather forecast uncertainties.
- Enhancing data accuracy through advanced forecasting techniques and real-time data collection to reduce uncertainties.
- Performing sensitivity analyses to understand the impact of these uncertainties on load flow results and strategy effectiveness.
- Collaborating with meteorological experts to refine weather data inputs, ensuring that the model aligns closely with real-world conditions.

By addressing these factors, the congestion management strategies can be better equipped to handle real-world variabilities, ultimately improving grid stability and operational efficiency.
By addressing these factors, the congestion management strategies can be better equipped to handle real-world variabilities, ultimately improving grid stability and operational efficiency.

Bibliography

[1] H. Behnia and M. Akhbari. "Integrated Generation and Transmission Maintenance Scheduling by Considering Transmission Switching". In: *International Transactions on Electrical Energy Systems* 29.4 (2019), e2792. DOI: 10.1002/etep.2792.

[2] Lilla Beke, Michal Weiszer, and Jun Chen. "A Comparison of Genetic Representations and Initialisation Methods for the Multi-objective Shortest Path Problem on Multigraphs". In: *SN Computer Science* 2.3 (2021). ISSN: 2661-8907. DOI: 10.1007/s42979-021-00512-z.

[3] J. Blank and K. Deb. "pymoo: Multi-Objective Optimization in Python". In: *IEEE Access* 8 (2020), pp. 89497–89509.

[4] Jingjing Chen, Qingfu Zhang, and Genghui Li. "MOEA/D for Multiple Multi-objective Optimization". In: *Evolutionary Multi-Criterion Optimization*. Ed. by Hisao Ishibuchi et al. Cham: Springer International Publishing, 2021, pp. 152–163. ISBN: 978-3-030-72062-9.

[5] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga. "Handling Multiple Objectives with Particle Swarm Optimization". In: *IEEE Transactions on Evolutionary Computation* 8.3 (2004), pp. 256–279. DOI: 10.1109/TEVC.2004.826067.

[6] Indraneel Das and J. E. Dennis. "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems". In: *SIAM Journal on Optimization* 8.3 (1998), pp. 631–657. DOI: 10.1137/S1052623496307510. URL: https://doi.org/10.1137/S1052623496307510.

[7] Kalyanmoy Deb and Himanshu Jain. "An Evolutionary Many-objective Optimization Algorithm Using Reference-point-based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints". In: *IEEE Transactions on Evolutionary Computation* 18.4 (2014). ISSN: 1089-778X. DOI: 10.1109/TEVC.2013.2281535.

[8] Kalyanmoy Deb et al. "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197. DOI: 10.1109/4235.996017.

[9] Matthias Dorfer et al. "Power Grid Congestion Management via Topology Optimization with AlphaZero". In: (Nov. 2022). URL: http://arxiv.org/abs/2211.05612.

[10] A. E. Eiben and J. E. Smith. *Natural Computing Series Introduction to Evolutionary Computing*. Tech. rep. URL: https://www.springer.com/series/.

[11] J. G. Falcón-Cardona, M. T. M. Emmerich, and C. A. Coello Coello. "On the Construction of Pareto-Compliant Combined Indicators". In: *Evolutionary Computation* 20.10 (Aug. 2022), pp. 1–28. ISSN: 1530-9304. DOI: 10.1162/evco_a_00307.

[12] Mitsuo Gen and Lin Lin. *A New Approach for Shortest Path Routing Problem by Random Key-based GA*. Tech. rep. 2006. URL: http://www.brunel.ac.uk/.

[13] G. Granelli et al. "Optimal Network Reconfiguration for Congestion Management by Deterministic and Genetic Algorithms". In: *Electric Power Systems Research* 76.6-7 (Apr. 2006), pp. 549–556. ISSN: 0378-7796. DOI: 10.1016/j.epsr.2005.09.014.

[14] D. P. Hardin and E. B. Saff. "Minimal Riesz Energy Point Configurations for Rectifiable d-dimensional Manifolds". In: *Advances in Mathematics* 193.1 (2005), pp. 174–204. ISSN: 0001-8708. DOI: 10.1016/j.aim.2004.05.006. URL: https://www.sciencedirect.com/science/article/pii/S0001870804001537.

[15] M. W. Hassan et al. "Optimal Generation Expansion Planning Considering Renewable Energy Integration". In: *Renewable Energy* 36 (2011), pp. 123–130.

[16] Fangguo He, Huan Qi, and Qiong Fan. *An Evolutionary Algorithm for the Multi-objective Shortest Path Problem*. Tech. rep.

[17] Christian Horoba. *Exploring the Runtime of an Evolutionary Algorithm for the Multi-objective Shortest Path Problem*. Tech. rep.

[18] Hisao Ishibuchi et al. "Comparison of Hypervolume, IGD and IGD+ from the Viewpoint of Optimal Distributions of Solutions". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 11411 LNCS. Springer Verlag, 2019, pp. 332–345. ISBN: 978-3-030-12598-1. DOI: 10.1007/978-3-030-12598-1_27.

[19] Hisao Ishibuchi et al. "Modified Distance Calculation in Generational Distance and Inverted Generational Distance". In: *Evolutionary Multi-Criterion Optimization*. Ed. by António Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello. Cham: Springer International Publishing, 2015, pp. 110–125. DOI: 10.1007/978-3-319-15892-1_8.

[20] "Kolmogorov–Smirnov Test". In: *The Concise Encyclopedia of Statistics*. New York, NY: Springer New York, 2008, pp. 283–287. ISBN: 978-0-387-32833-1. DOI: 10.1007/978-0-387-32833-1_214. URL: https://doi.org/10.1007/978-0-387-32833-1_214.

[21] Hong Wei Li et al. "Two Successive Approximate Models and Linear Formulations for Power Flow Analysis of DC Grid". In: *Journal of Electrical Engineering and Technology* (2024). ISSN: 2093-7423. DOI: 10.1007/s42835-024-01908-6.

[22] P. Li et al. "A Connectivity Constrained MILP Model for Optimal Transmission Switching". In: *IEEE Transactions on Power Systems* 36.5 (Sept. 2021), pp. 4820–4823. DOI: 10.1109/TPWRS.2021.3089029.

[23] X. Li et al. "Generation Expansion Planning with Environmental Constraints". In: *IEEE Transactions on Power Systems* 28 (2013), pp. 456–463.

[24] Haiping Ma et al. "A Comprehensive Survey on NSGA-II for Multi-objective Optimization and Applications". In: *Artificial Intelligence Review* 56.12 (Dec. 2023), pp. 15217–15270. ISSN: 1573-7462. DOI: 10.1007/s10462-023-10526-z.

[25] Antoine Marot et al. "Learning to Run a Power Network Challenge for Training Topology Controllers". In: *Electric Power Systems Research* 189 (2020). ISSN: 0378-7796. DOI: 10.1016/j.epsr.2020.106635.

[26] Luis Martí. *NSGA-III in Python*. https://nbviewer.org/github/lmarti/nsgaiii/blob/master/NSGA-III%20in%20Python.ipynb. 2023.

[27] M. Murugan et al. "Transmission Constrained Generation Expansion Planning Problem". In: *Journal of Power Systems* 34 (2019), pp. 123–145.

[28] M. Numan et al. "Mobilizing Grid Flexibility through Optimal Transmission Switching for Power Systems with Large-scale Renewable Integration". In: *International Transactions on Electrical Energy Systems* 30.3 (Mar. 2020), pp. 1–15. DOI: 10.1002/2050-7038.12211.

[29] Muhammad Numan et al. *The Role of Optimal Transmission Switching in Enhancing Grid Flexibility: A Review*. 2023. DOI: 10.1109/ACCESS.2023.3261459.

[30] R. P. O'Neill et al. "Economic Analysis of the N-1 Reliable Unit Commitment and Transmission Switching Problem Using Duality Concepts". In: *Energy Systems* 1.2 (Jan. 2010), pp. 165–195. DOI: 10.1007/s12667-009-0005-6.

[31] Md Saydur Rahman et al. "Comparative Evaluation of Weather Forecasting Using Machine Learning Models". In: (Feb. 2024). URL: http://arxiv.org/abs/2402.01206.

[32] Erica van der Sar, Alessandro Zocca, and Sandjai Bhulai. *Multi-Agent Reinforcement Learning for Power Grid Topology Optimization*. 2023.

[33] Ke Shang et al. *A Survey on the Hypervolume Indicator in Evolutionary Multiobjective Optimization*. Feb. 2021. DOI: 10.1109/TEVC.2020.3013290.

[34] R Shigenobu et al. "Multi-objective generation expansion planning using evolutionary algorithms". In: *Electric Power Systems Research* 154 (2018), pp. 45–52.

[35] Umair F. Siddiqi et al. "A Memory Efficient Stochastic Evolution Based Algorithm for the Multi-objective Shortest Path Problem". In: *Applied Soft Computing Journal* 14.PART C (2014), pp. 653–662. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2013.09.008.

[36] Edward H. Simpson. "Measurement of Diversity". In: *Nature* 163 (1949), p. 688.

[37] Walmart Global Tech. *Genetic Algorithm Demystified Part 1*. 2021. URL: https://medium.com/walmartglobaltech/genetic-algorithm-demystified-part-1-776e0f53703f.

[38] TutorialsPoint. *Genetic Algorithms - Crossover*. https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_crossover.htm. Accessed: 2024-08-07. n.d.

[39] Jan Viebahn et al. *GridOptions Tool: Real-World Day-Ahead Congestion Management Using Topological Remedial Actions 10528 C2 Power System Operation and Control PS2 Changes on System Operation and Control Considering the Energy Transition*. Tech. rep.

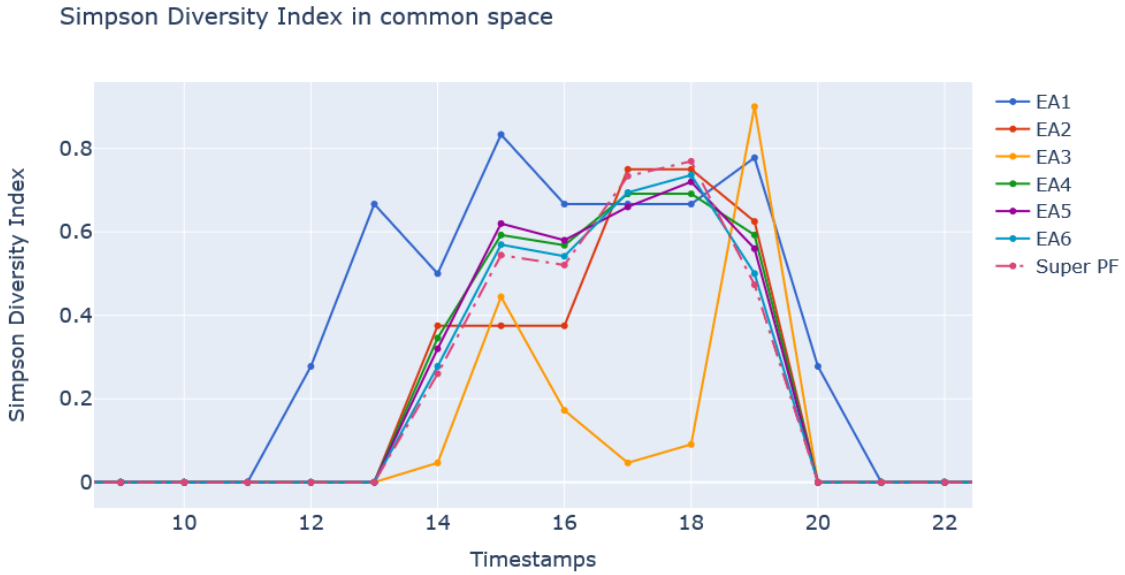
[40] Mark Wineberg and Franz Oppacher. "Distance between Populations". In: vol. 2724. July 2003, pp. 1481–1492. ISBN: 978-3-540-40603-7. DOI: 10.1007/3-540-45110-2{_}20.

[41] Chi Zhang and Jianhui Wang. "Optimal Transmission Switching Considering Probabilistic Reliability". In: *IEEE Transactions on Power Systems* 29.2 (Mar. 2014), pp. 974–975. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2013.2287999.

Chapter 14

Appendix

14.1 SDI Figures



14.2 Coverage strategies

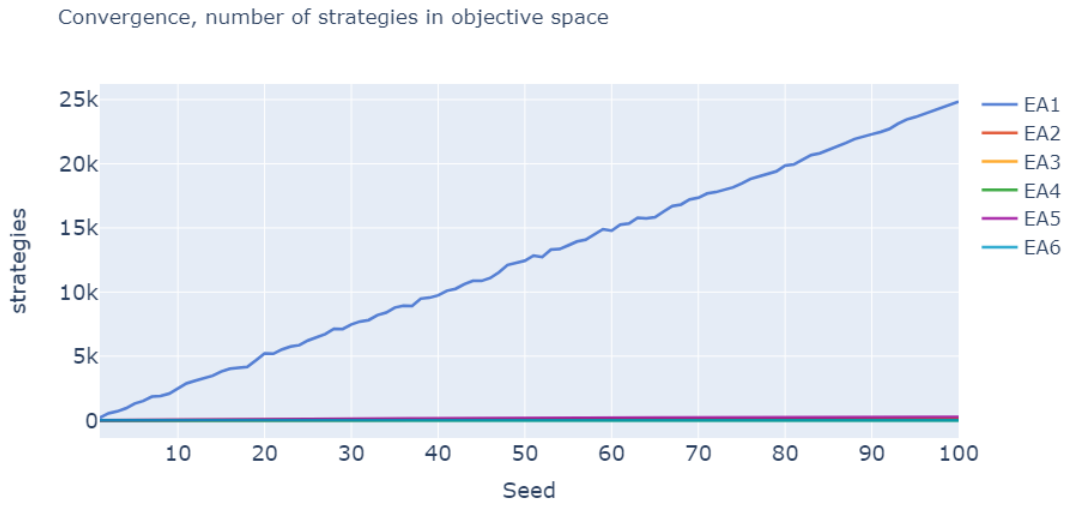


Figure 14.1: Topological diversity of EA1 in common space.

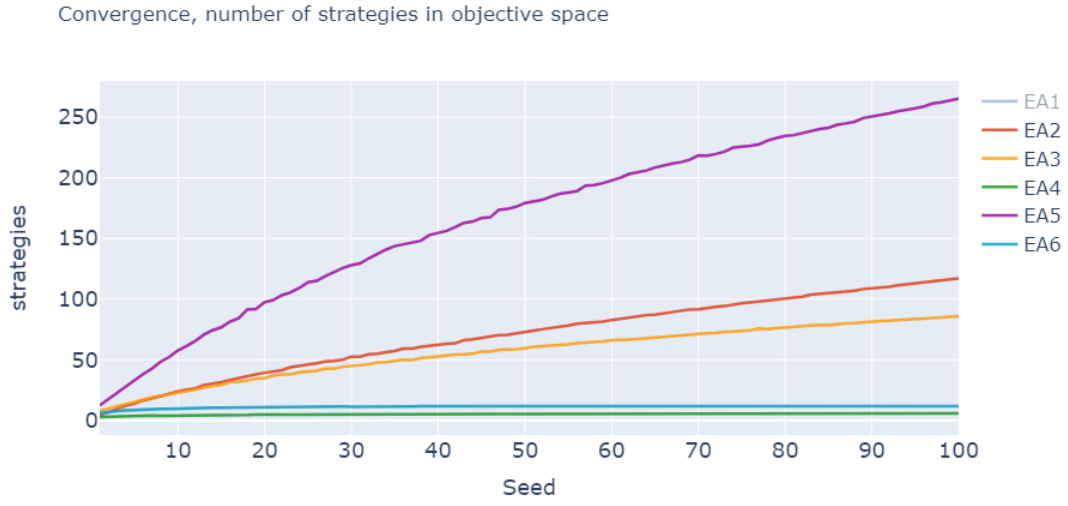


Figure 14.2: Topological diversity of EA1 in common space.

14.3 CSSC Point Figure

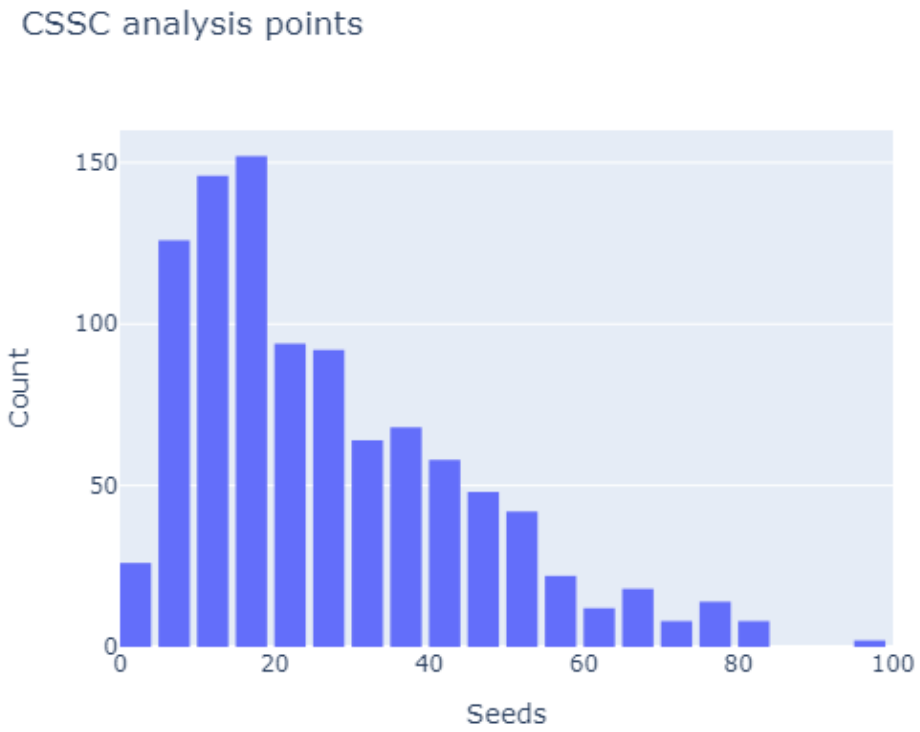


Figure 14.3: CSSC of points EA4.

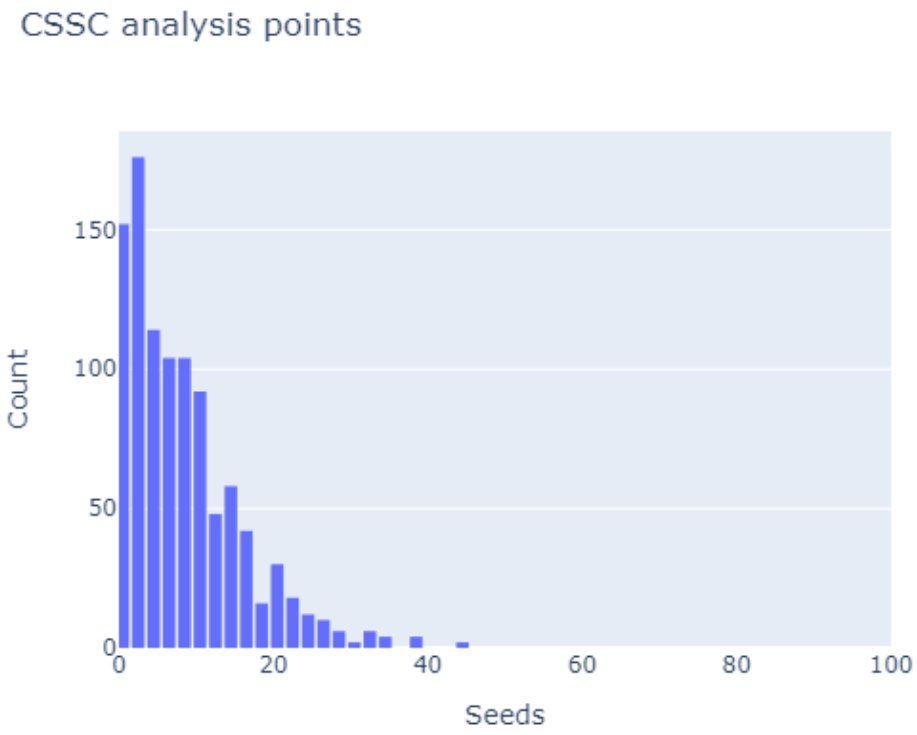


Figure 14.4: CSSC of points EA5.

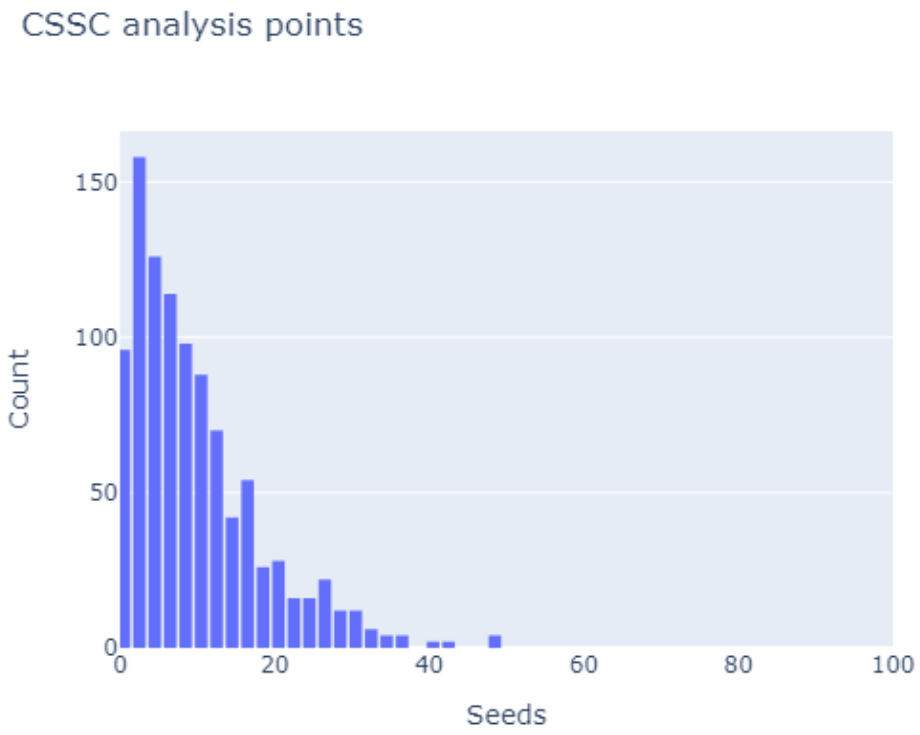


Figure 14.5: CSSC of points EA6.

14.4 Topology and substation diversity

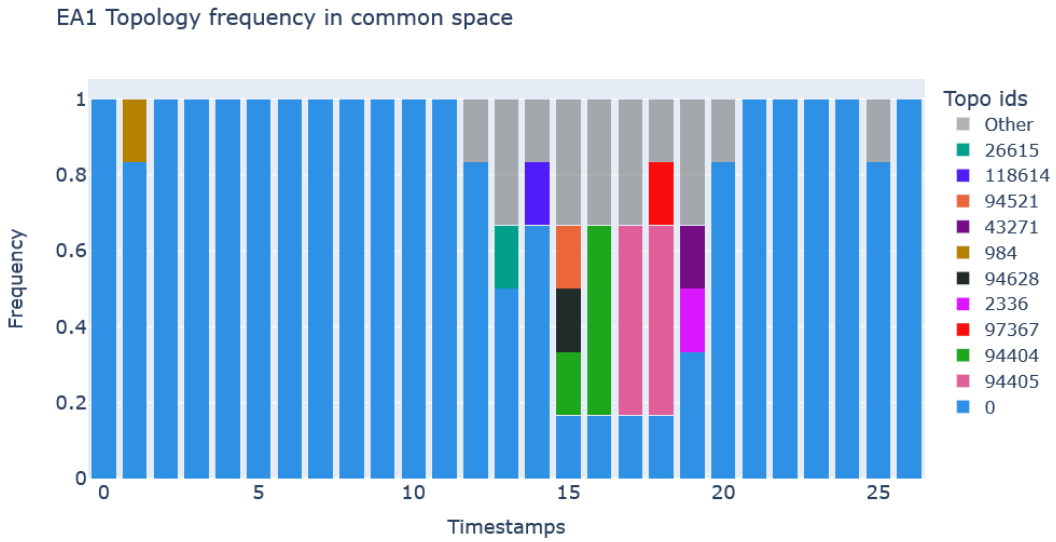


Figure 14.6: Topological diversity of EA1 in common space.

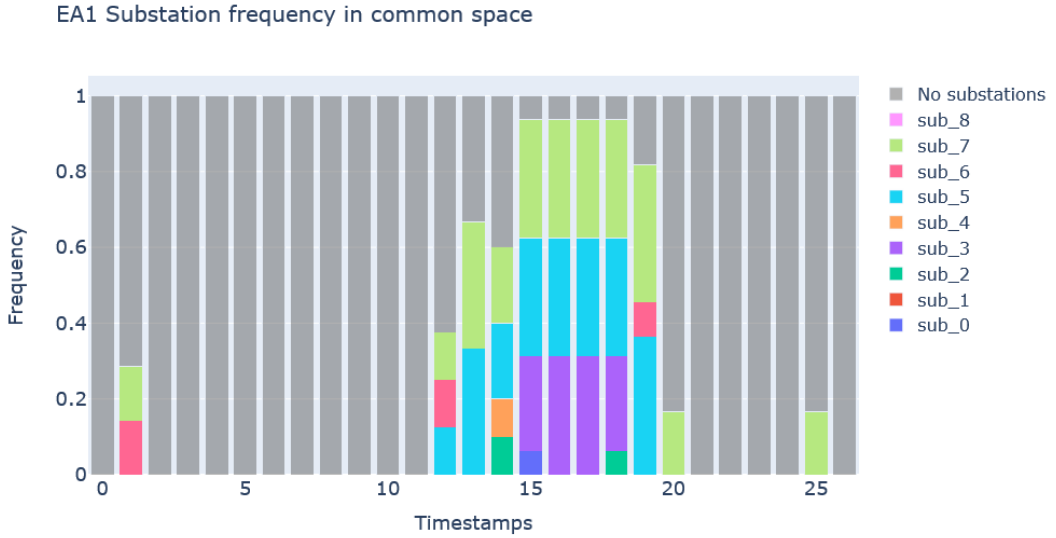


Figure 14.7: Substation diversity of EA1 in common space.

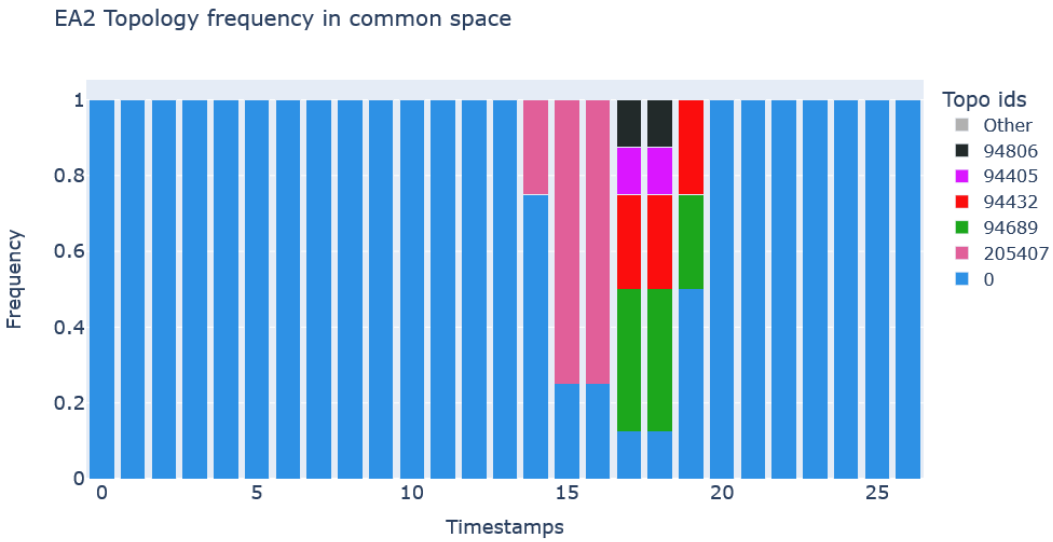


Figure 14.8: Topological diversity of EA2 in common space.

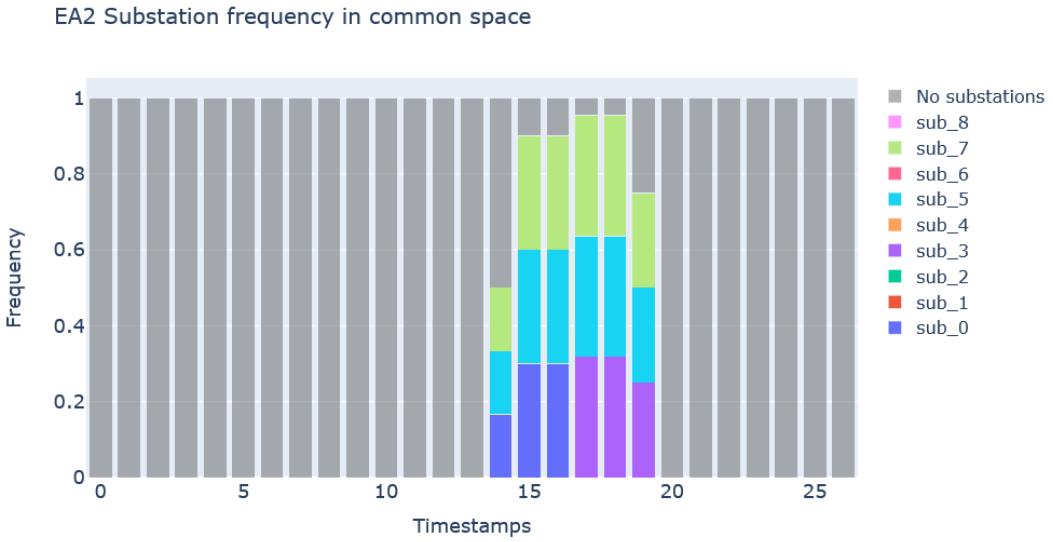


Figure 14.9: Substation diversity of EA2 in common space.

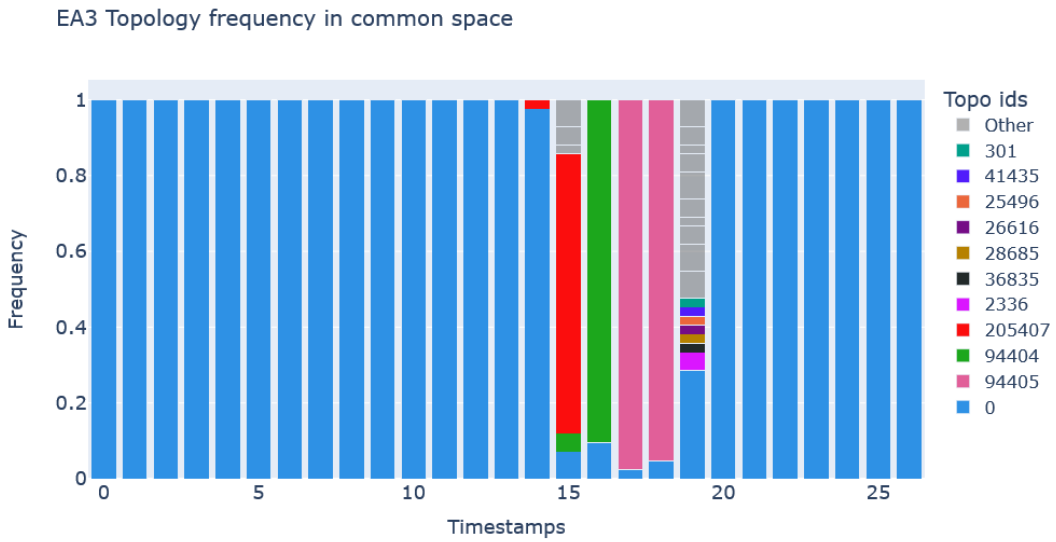


Figure 14.10: Topological diversity of EA3 in common space.

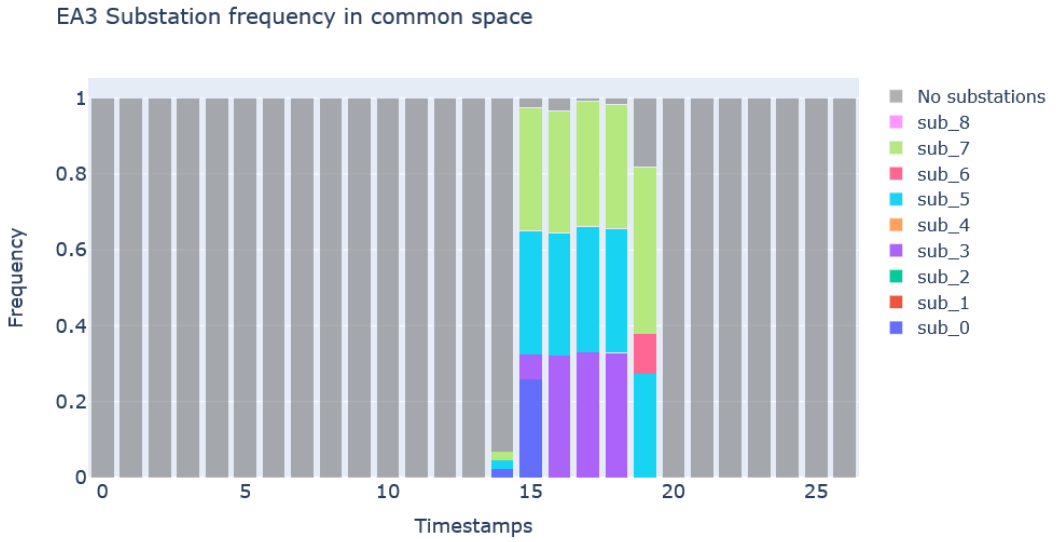


Figure 14.11: Substation diversity of EA3 in common space.

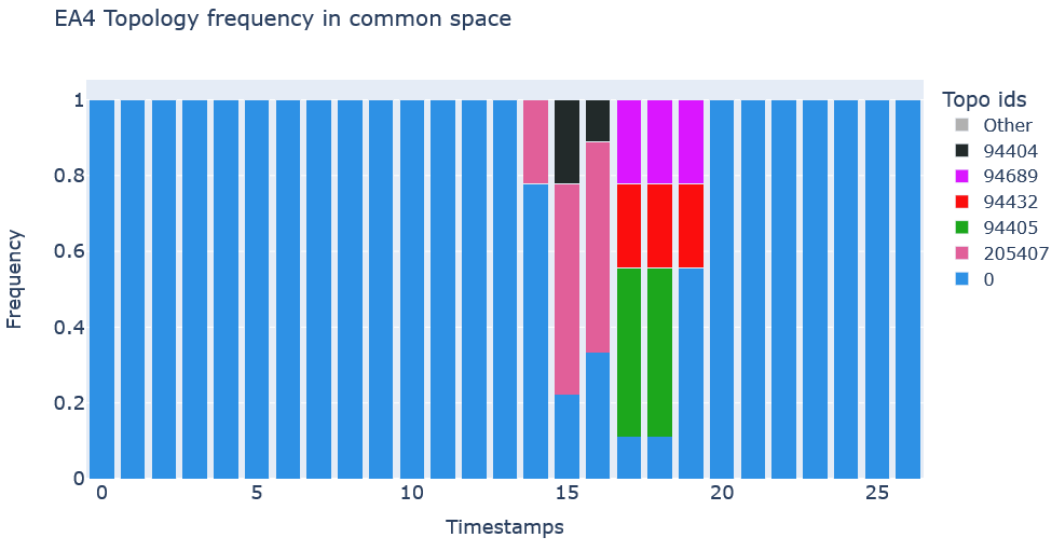


Figure 14.12: Topological diversity of EA4 in common space.

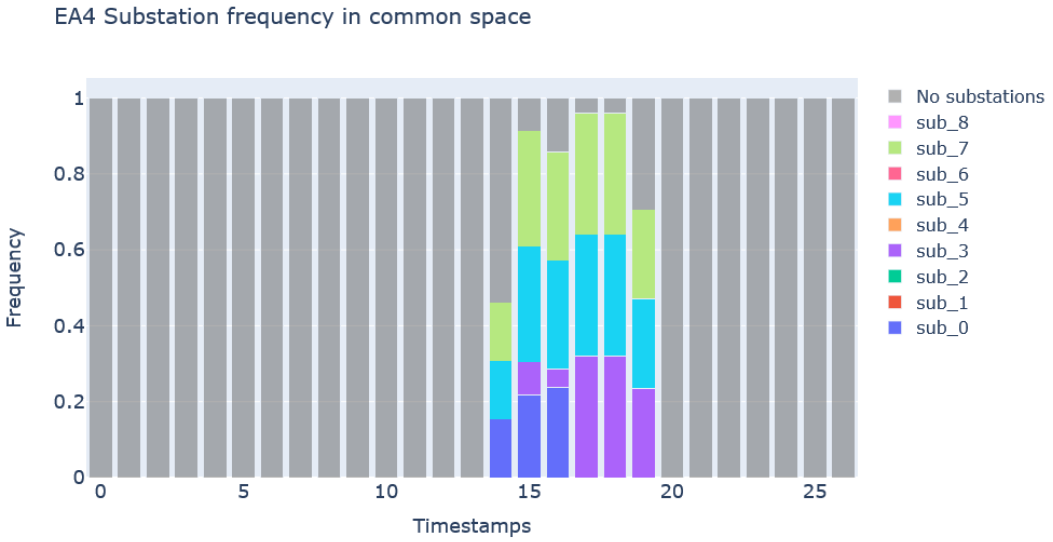


Figure 14.13: Substation diversity of EA4 in common space.