Automatic Power System Restoration

Application of a Search Algorithm

Lars Lindgren

LUND UNIVERSITY

Licentiate Thesis
Department of Measurement Technology and
Industrial Electrical Engineering

2009

Department of Measurement Technology and Industrial Electrical Engineering
Faculty of Engineering
Lund University
Box 118
SE-221 00 LUND
SWEDEN

http://www.iea.lth.se

ISBN 978-91-88934-52-9
CODEN:LUTEDX/(TEIE-1060)/1-144/(2009)

Printed in Sweden by Media-Tryck
Lund University
Lund, 2009

MALMÖ UNIVERSITY

School of Technology
Malmö University

Abstract

Large blackouts are rare but can have a high impact on society and the economy. This impact strongly depends on their duration and therefore speedy restoration is important.

More distributed and renewable generation and more interconnected power systems that are operated close to their limits will lead to new challenges in power system restoration. At the same time society is becoming ever more dependent on electric power supply. The ability to restore the power supply after blackouts is therefore vital and means to improve it should be investigated.

For the operators in a control centre, restoration of a power system involves taking a number of actions. If possible these follow written instructions, but often also needs to be improvised. This thesis proposes that automation is also applied to this task. An algorithm that finds a suitable sequence of actions to restore the power supply after a blackout can be used both to assess different strategies or potential restoration difficulties before any real event and to support the operators during an actual restoration situation.

This thesis shows that the restoration problem is NP-hard and proposes a simple heuristic restoration algorithm based on the well known tree search algorithm A^*. The proposed algorithm is applied to a simplified power system model using static load flow analysis.

NORDIC32, which is used as test system, is a fictitious power system model with 32 substations and 36 generators. It attempts to broadly represent the synchronized Nordel transmission system.

The algorithm has been found to be robust against variations in both the search parameters and the power system model. For black start in the base case the algorithm typically finds a sequence of around 750 actions that restore all loads after the algorithm has tested about 2000 actions. The actions include network switching, startup and setpoint changes of generators and load reconnections. On a 2.2 GHz AMD Opteron processor this computation takes less than 30 seconds.

The fact that the computation time is as short as 30 seconds indicates that the algorithm realistically could handle more complex systems either larger or with more detailed modelling.

Keywords

Power system restoration, Load flow analysis, Tree searching, A^*-search, Bulk power system, Black start, Power system control, Power systems, Search methods, Automation, Predictive control, NP-hard.

Sammanfattning

Omfattande strömavbrott är ovanliga men kan få stora konsekvenser för samhället och ekonomin. Eftersom konsekvenserna är starkt beroende av avbrottens längd är snabb återstart av kraftsystemet viktig.

Fler distribuerade och förnybara energikällor och mer sammanlänkade elkraftsystem som drivs närmare sina gränser kommer att leda till nya utmaningar vid återstart av elkraftsystemet. Samtidigt har samhället blivit allt mer beroende av tillförlitlig elförsörjning. Förmågan att återställa elförsörjningen efter strömavbrott är därför viktig och möjligheterna att förbättra denna bör undersökas.

För kontrollrumspersonalen innebär återstart av kraftsystemet att ett stort antal åtgärder vidtas. Detta följer om möjligt en skriven instruktion men kräver vanligen mycket improvisation. Denna avhandling föreslår att även automatisering utnyttjas. En algoritm som finner en lämplig sekvens av åtgärder för att återställa elförsörjningen efter ett strömavbrott kan användas både för att utvärdera olika strategier eller bedöma eventuella svårigheterna vid en återstart innan någon verklig händelse inträffat och för att stödja operatörerna under en verklig återstart.

Denna avhandling visar att återstartsproblemet är NP-hårt och föreslår en enkel heuristisk återstartsalgoritm baserad på den välkända trädsökningsalgoritmen A*. Den föreslagna algoritmen arbetar med en förenklad modell av kraftsystemet baserad på statiska effektflödesberäkningar.

Som testsystem har NORDIC32 använts , vilket är en fiktiv kraftsystemsmodell med 32 kopplingsstationer och 36 generatorer. NORDIC32 är ett försök att återspegla huvuddragen hos det synkroniserade Nordel-systemet.

Algoritmen har visats vara robust mot variationer både i sökparametrar och i kraftsystemsmodellen. För dödnätsstart med de grundparametrar som använts genererar algoritmen typiskt en sekvens med omkring 750 åtgärder

för att återställa elförsörjningen till alla förbrukare. För att finna dessa åtgärder testar algoritmen ca 2000 åtgärder. Dessa åtgärder innefattar kopplingar i nätet, start och ändring av driftläge för kraftverksenheter samt belastningstillkopplingar. På en 2,2 GHz AMD Opteron processor tar denna beräkning mindre än 30 sekunder.

Det faktum att beräkningstiden är så kort som 30 sekunder visar att algoritmen bör kunna hantera mer komplexa system, antingen större eller med mer detaljerade modellering.

Acknowledgement

This project was initiated by Dr. Bo Eliasson who also supervised the beginning of this project and my B.Sc. project.

This thesis had not been possible without the support from people around me. I want to thank my supervisors at Lund University, especially:

Assoc. Prof. Olof Samuelsson, my main supervisor for his strong support even in periods when the work has progressed slowly and for his help with both content and language.

Prof. Sture Lindahl, for using his expertise to give me many new views and insights.

Prof. Gustaf Olsson and Magnus Akke for their enthusiasm and insights.

My colleagues at Malmö University and at the Department of Measurement Technology and Industrial Electrical Engineering at Lund University have been very important in the work with this thesis. I want to especially thank Prof. Bengt J. Nilsson for his help with the computer science part of this work and fellow postgraduate student Tech. Lic. Christian Andersson for our close cooperation in the beginning of the project. I also want to thank John Niubo at Malmö University for his support.

Energinet.dk, previously Eltra, has financed this project which is gratefully acknowledged. Two persons at Energinet.dk have been especially important, they are Assoc. Prof. John Eli Nielsen and Dr. Per Lund.

Of course I also want to thank my family and others for their support during this work.

I have learned very much during this work.

Terminology

Active power The average power over a cycle in the AC voltage.

Bottom up restoration Restoration from several points and then synchronizing the different islands.

Circuit breaker A mechanical switch that can operate fast and is able to interrupt fault current at short-circuits without being damaged.

Disconnector A mechanical switch that only can be opened and closed when the current or the voltage difference is very low. Disconnectors are used to change the substation configuration, often after a circuit breaker has been used to turn off the current. They are also used to safely isolate parts for repair and maintenance.

Distributed slack bus A slack bus in a power flow calculation is the bus where the active power is adjusted to get the active power balance right. In a power flow calculation with distributed slack bus this adjustment is distributed to several buses according to some weighting factor.

Distribution network The low voltage and medium voltage networks that distribute the power to small and medium sized customers. It is common that distribution networks are operated in a radial configuration since that simplifies the protection system.

Distributed generation Small generating units that are connected to the distribution network.

Electrical island A part of a power system that is disconnected from the rest but remains energized. The main energized part of the power system can be seen as special case of an island.

Energy not delivered The energy that customers wanted to buy but could not because of a power outage. Normally calculated as the predicted consumption in the area that is without power.

Evaluation function A function that is used to evaluate and compare how good states or solutions are.

Flat start Flat start means that all voltages are set to a fixed voltage, usually the nominal value (1 p.u.) and all voltage phase angles are set to 0 before the first iteration in a power flow calculation.

Generating unit A power plant can have one or more generating units, each mainly consisting of a turbine and a generator. Each generating unit can be started independent of the other generating units.

Heuristics Method based on experience such as "rule of thumb".

In a search algorithm context, heuristics cannot be guaranteed to generate the correct answer but are useful in some cases.

Load flow analysis See Power flow analysis.

N-1 criterion A rule in power system operation that demands that the system shall be able to withstand the failure of any component without a collapse.

NP-complete A class of computation problems that are considered hard to solve efficiently.

NP-hard The class of all computation problems that is at least as hard to solve as any NP-complete problem, see section 5.1.

Power flow analysis A calculation of the active and reactive power flows in a power system given the generation and consumption in the system. Can include different regulators such as voltage regulators and tap changers. The most demanding step is to calculate the voltage magnitude and phase angle for all buses, which is done iteratively. After that the power flows on the lines and in the transformers are trivial to calculate.

Power plant A power plant consists of one or more co-located generating units.

Reactive power Power that over a cycle in the AC voltage moves in both directions so that on average no power is transferred but it still generates losses. In a system with inductive power lines the voltage can be controlled by controlling the reactive power.

Renewable generation Power generation that use renewable energy sources such as wind, sunshine or biomass. Often confused with distributed generation since renewable generation is often distributed.

Substation A location for switching equipment and transformers used to connect power lines to each other.

Switch A device that can close or open an electrical circuit, see circuit breaker and disconnector.

Synchronization The act of closing a circuit breaker between two electrical islands. Before the synchronization it is important that the difference between the voltage magnitudes, frequencies and voltage phase angles in the two parts are small enough. Synchronization is rather much like engaging a mechanical clutch with no slip in that the axes in the synchronous machines in the whole synchronized system will be spinning synchronously.

Three phase symmetry A situation where the currents and voltages in the three phases are identical except for a phase shift of 120 degree between the phases. In this case it is sufficient to calculate the power, voltage and current in one phase and then multiply the power by three to get the three phase power. High voltage AC systems are normally operated close to three phase symmetry.

Top down restoration Restoration from one island in the system and then expanding that island by energizing more and more components.

Topology calculation In power system analysis topology calculation is the calculation of how the system is connected given the state of the switches. It answers two questions:

- Which components are connected directly to each other and thus have the same voltage?
- Which electrical islands exists in the system?

It's a necessary pre-processing step before a power flow calculation can be performed.

Transmission system The high voltage part of a power system that transmits power over long distances.

Contents

Chapter 1

Introduction

The reliability of a power system depends on the frequency and the duration of power outages. Much of the research about power systems focuses on reducing the frequency of power outages.

The focus in this thesis is instead to reduce the duration of large power outages or blackouts by improving the restoration process.

1.1 Background

This research project deals with the electric power system and specifically how to manage a restoration situation in which large parts of the power system have lost voltage after a fault or a collapse. In such situations, the start-up and regulation of generating units must be accurately coordinated with the connection of loads and power lines, to eliminate the risk of another collapse.

This project was motivated by the growing amount of distributed power generation which gives new questions about how restoration should be performed. The large number of small generating units also raises questions about if restoration can be performed automatically at a lower level in the power system.

Five approaches to the study of these issues have been identified:

1. Theoretical arguments.

2. Analysis of actual sequences of events.

3. Simulation studies with sequences of actions produced manually by a person, without real-time requirements.
4. Realistic real-time simulation exercises in which a team of operators attempts to manage prepared scenarios unknown to them, to determine how well they manage the situation, given different strategies for the work.
5. To use automatic search algorithms to find suitable sequences of actions in different restoration situations and assessment of how difficult it is for the algorithm to start the power system in each situation. By varying algorithm parameters and comparing the results, conclusions as to which strategy should be applied can be reached.

It is difficult to develop theoretical arguments in such a way that they are applicable in practice and generally applicable to such complex problems.

Analyzes of actual sequences of events have already been done; see [4, 20], for example. The problem is that one can only investigate what actually happened; to determine whether alternative strategies would have been better is hard.

Simulation studies with sequences of actions produced manually by a person with or without real-time requirements must be structured in some way; otherwise they will lead to results that depend entirely on the person's skill and expectations in different situations. If the production of the sequence is structured according to a particular set of rules, it would bee far better to automate the sequence generation. It then falls into category 5 above.

Realistic real-time simulation exercises as mentioned in category 4 above would have been interesting, but would have required access to operators to act as experimental subjects and set up realistic scenarios in which all significant aspects of the situation were simulated. It was felt that this was not possible within the scope of the project.

This licentiate-thesis deals with the development of an algorithm to be used in approach number five above.

1.2 Scope of the thesis

In order to compare different power system restoration strategies or situations in terms of energy not delivered, it would be helpful to have an algorithm for finding a sequence of actions that restores a power system.

The idea is that while not necessarily giving the optimal solution the algorithm can be used for finding a feasible solution and this can be used to estimate the energy not delivered[1]. The aim of this thesis is to investigate such an algorithm. The algorithm is tested on a fictive power system.

The power system model is simplified by not including any time dependent or dynamic models and only modelling deterministic behaviour.

It is assumed that the protection system makes sure that the system state after the collapse contains no equipment that operates outside its specification. It is also assumed that the state of the system is fully known.

After all loads are restored some process is needed to reach a more normal operating point that fulfils the N-1 criterion and other constraints for normal operation this is not discussed in this thesis.

1.3 Previous work

Power system restoration has of course been a necessary part power system operation since the first power systems were built. The traditional method is to have written instructions with general rules that are executed manually [15]. Since the restoration situation can often not be known beforehand this includes more or less improvisation. Since the use of computers became common there have been attempts to automate the whole process or part of the process.

The book “Power System Restoration methodologies & implementation strategies“ by M. M. Adibi collects many of the key papers in the area and complements these with comments [8]. The book was published year 2000.

The power system restoration research can be categorized according to several different criteria:

Power System type Power system restoration needs to be done on different types of power systems and at different levels, such as transmission systems [18, 25, 36], distribution systems [19, 33, 36] and shipboard power systems [37].

Disturbance Some research focuses on the common case that only a small part of the system is without power [18], other research focuses on

---

[1] Defined as the energy that the customers are prevented from consuming by the blackout.

black start after a complete blackout [27, 25, 33].

Sub problems Much research focus on different sub problems in power system restoration such as generator start-up sequence [26], standing voltage phase angles [38], target switch configuration without finding a sequence of actions [18, 19, 27] and selecting suitable islands to restart [30].

Modelling The resolution of the model that is used when analyzing power system restoration is a compromise between many details to capture the behaviour of the real system or few details to keep down the implementation complexity and the computation time. Both static power flow calculations [18, 27] and dynamic electromechanical models [39, 22, 33] are used. Some aspects not normally included in power system models such as generating unit start-up times and ramping are important to consider in restoration studies.

Automation level Much research is on the fundamental issues in power system restoration and applies to manual restoration as well as automated restoration [9, 11]. Other research treats how to support the operators during restoration using simple prepared rules [15] or how to automate power system restoration [25, 33, 36] or parts of the process.

Algorithm Proposed algorithms for automatic restoration or operator support use many different algorithms such as expert systems [29, 22, 36], petri-nets [25], tree search [25, 33], genetic algorithms [19, 36], discrete particle-swarm optimization [27] and combined tabu search and a multi-agent system [18].

The most relevant articles found are listed below. The following criteria have been used when selecting the articles:

- The used algorithm can find a sequence of actions for restoration from black start or near black start of bulk power systems.
- Describes the used algorithm in enough details so the inner workings of the system can be understood.
- Needs no system specific manually prepared restoration plans or guidelines.

- Uses a significantly different approach from the other shown articles.

The found articles are:

- ”Backtracking based algorithm in hierarchical time-extended petri net model for power system restoration“ by Dong Liu et al. from 2005 [25] uses a petri net model and heuristic rules to choose which actions to test and backtracking when it does not find a way forward. The presented implementation does not perform a power flow calculation and can therefore not check voltage levels and so on.

- ”An agent approach to bulk power system restoration“ by T. Nagata et al. from 2005 [31] uses a decentralized multi-agent system where the different parts of the power system has associated computer programs that negotiate with each other. The presented implementation has no constraints on reactive power or voltage and does not perform a power flow calculation. An overview of an agent based system that includes power flow calculation is given in [24].

- ”Development of an interactive rule-based system for bulk power system restoration“ by Teo C.Y. and Wei Shen from 2000 [13]. This article is interesting in that it compare the values from the simulation with measured values and updates the model accordingly and then updates the model. It uses very fixed rules for how to restore the system for example ”IF any two subsystem have met 50% of their local load demanding, THEN these two subsystems are ready to synchronize.“ and the active power generation is only controlled by the local frequency. This probably makes the system unsuitable for comparing restoration in different situations.

- ”Generation capability dispatch for bulk power system restoration: a knowledge based approach“ by Liu C.-C. et al. from 1993 [40] focuses on finding a generator startup sequence given the capabilities of the production units. The A^*-algorithm is used to find paths in the power system directly. It does not check the power flow solution for the proposed solution steps.

- ”Development of a guidance method for power system restoration“ by Kojima Y. et al. from 1989 [12] uses an expert system to guide operators during restoration. First a target state is defined for the power system a few hours later that the operators can modify if they wish. Then the expert system tries to find actions that take the system to the target state.

An interesting article on restoration at the distribution level is ”New challenges in power system restoration with large scale of dispersed generation insertion“ by Thi Thu Ha Pham et al. from 2009 [33]. The focus in this article is on restoration at the distribution level by using distributed generation while waiting on the restoration at the transmission level. This article uses a depth first tree search to enumerate switching sequences that fulfils the power balance in each step. They are then ordered by how much load they restore and a dynamic simulation is preformed on each until a valid sequence is found. This method is interesting but probably results in too many switching sequences to test if applied on the transmission level. The article also describes solutions to a number of related problems.

None of these methods have gained widespread use. Furthermore there seems to be no commonly used test-cases to test algorithms against.

Finally it has been stated without proof that the power system restoration problem is NP-hard [21].

1.4 Contribution

In this work, a new tree search algorithm for power system restoration has been developed. It is based on the well-known A^*-algorithm. The algorithm has been shown to be able to do a full restoration of a rather complex power system with a simplified model. This is done without specifying any overall restoration plan or specific goal state beforehand.

The work has been published in the papers:

Lindgren, L. and Eliasson, B. Automated power system restoration, *Nordic Wind Power Conference*, November 1-2, 2007, Risø DTU, Denmark.

Samuelsson, O., Lindgren, L. and Eliasson, B. Simulated power system restoration, *43rd International Universities Power Engineering Conference*, IEEE, september 1-4, 2008, Padova, Italy.

The performance of the algorithm under different conditions has been investigated.

The power system restoration problem has been shown to be NP-hard. This was done by reduction from Steiner tree in graphs.

1.5 Applications

An automatic power system restoration algorithm can be used in different ways:

- Automatic power system restoration.
- Operator support, suggesting actions to the operator in real time.
- Off-line planning for restoration in advance of the blackout. This includes both comparing different restoration plans/strategies' and comparing restoration sequences after modifications in the net.

The focus in this thesis is planning for restoration in advance.

1.6 Outline of the thesis

Chapter 2 gives an overview of the power system restoration problem from a practical power system operation view point. Readers with previous knowledge of these issues can skip this chapter.

Chapter 3 and 4 describe the modelling of the power system components and the used test system. Especially the sections 3.4 about time calculation and 4.2 about power flow visualization are important in order to interpret the results.

Chapter 5 places the power system restoration problem in a computer science context and shows that the problem is NP-hard. It is rather independent from the rest of the thesis.

Chapter 6 introduces search algorithms, explains the used algorithm and the adaptation of the algorithm to the power system restoration problem.

Chapter 7 presents restoration sequences that are generated by the algorithm when applied to the test system.

Chapter 8 and 9 on sensitivity analysis show the effect of changing the search and model parameters respectively.

Chapter 10 summarizes and evaluates the results.

Chapter 11 proposes possible improvements to the presented algorithm.

Appendix A gives a brief overview of how the software for this project was developed.

Appendix B contains the model file that describes all the details of the used test system and the test cases.

Chapter 2

Overview

2.1 The operational states of the power system

In order to describe operating situations in a power system they can be categorized in five groups called operational states of the power system, see Figure 2.1 [16]. Each of these states has its own operational rules.

In the *normal* operational state the objective is to run the power system as economical as possible with enough security margins. Typically it is required that the system should be able to withstand any component failure at any time without any power interruption to the consumers. This is called the N-1 criterion.

When component failures and/or unexpected conditions lead to a state where the N-1 criterion is no longer fulfilled the power system is said to be in the *alert* state. In this state the goal for the operators is to bring the system back into the normal operating state by preventive control actions. In the alert state all or almost all load is still supplied.

Additional disturbances in the alert state can cause the system to enter the *emergency* state. This state is characterized by problems such as overloads, over/under-voltages, over/under-frequency and instability. Most actions in this state are taken by the protection system. There is not enough time for manual operations. The goal in this state is to reach a stable operating point where all equipment operates within its specifications.

The state *in extremis* means a partial or total blackout, the goal in this state is to stop the deterioration of the operation state and to reach a stable operating point from which to start restoration.

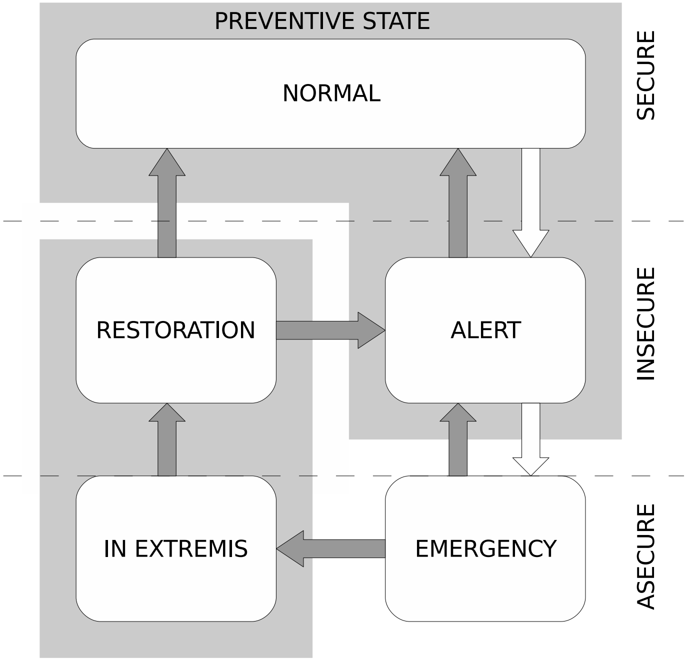


Figure 2.1: The operating states and state transitions of a power system, adapted from [16]. White arrows represents foreseen or unforeseen disturbance. Gray arrows represent control and/or protective actions.

When a stable operating point with partial or total blackout is reached the system enters the *restoration* state. The goal in this state is to reconnect all loads, when this is achieved the system either enters the alert or the normal operating state. This thesis focuses on the process of going from a stable operating point to the alert state.

2.2 Restoration

Power system restoration is a very complex task. The restoration is here described as seen from the perspective of a TSO (Transmission System Operator) such as Energinet.dk or Svenska Kraftnät.

The first problem facing the operators is to establish an overall picture of the state of the power system. Often the cause of the blackout is not fully known and it would take too long to analyze it before starting the restoration. The list of alarms will be very long and the clocks are often not synchronous so it can be hard to see what event that caused the other. It is likely that there are unknown faults in the system.

It is also not uncommon that there are problems with the SCADA (Supervisory Control And Data Acquisition) system so there can be problems both with reading remote measurements and taking remote actions. This is because there will be very many alarms in the SCADA system and also the backup power in substations could fail.

The operators are left with a situation where they may not know what has happened and why. They also do not know if there are any faults in the power system and the SCADA system may not be fully operational.

The operators often also need to communicate with control centres in adjacent transmission systems and with the power plant operators in order to get a picture of to what extent they can help or need help. There will also be many requests from media to get information of what happened and estimates of when the power supply will normalize.

Small blackouts, involving only a few substations, can often be handled rather easily since they occur more often than the larger blackouts so the operators has more experience with them and the load can often be reconnected as soon as the operational reserves are able to meet the demand.

Larger blackouts, affecting the whole or significant parts of the power system, are much harder to restore since large parts of the power system need to be energized and a significant part of the generating units will not be available.

In order to handle restoration situations the operators have prepared restoration plans and guidelines for some typical blackout situations. The operators will try to apply these to the current situation.

Two major strategies in power system restoration can be defined, bottom up vs. top down. Bottom up means that several smaller electrical islands are started in parallel; these are then used to energize the transmission system. The need to synchronize the islands can slow down the process. Top down means that the transmission system is energized from one point and all the lower voltage levels are energized from the transmission system and the whole energized power system is kept synchronized.

In Sweden the general plan is to restore the system from the North and southwards, which is an example of top down. The reason for this is that the hydropower in the North is normally much easier to use during the restoration and faster to start. Also it is very likely that some part of the Northern part of the system will not be affected by the blackout. Energinet.dk is working towards an ambitious bottom up strategy in their cell-project [28]. In power systems that are dominated by large thermal power plants it is common to have separate plans for the restart of each large steam turbine plant, which is a form of bottom up restoration.

The first priority is normally to energize all substations. This has several reasons. The remote controls of substations operate on battery power and are normally dimensioned for powering the substation control equipment a specified time, Svenska Kraftnät has for example chosen 12 hours in new substations [6], but this can be uncertain due to aging of the batteries. If the batteries run out before the substations are energized the restoration is further delayed due to the need to send service crew with mobile diesel generators to the substation to power the control system and recharge the batteries.

It is also important to get power to the power plants so they can initiate their preparation to start as soon as possible. If a thermal power plant is without power for some time it will cool down and it will take a considerable amount of time to restart it. Also as larger parts of the system become energized more resources become available to the operators. If there is some other prioritized loads in addition to substation control and power plant auxiliaries it is also necessary to energize the substations in order to be able to supply the prioritized loads. Another advantage of energizing substations early is that permanent faults in disconnectors, current breakers and transmission lines will often be found early.

During the energization the line currents will be very low so the reactive generation in the lines will give problems with very high voltages. In order to avoid this some loads may be reconnected during the energization.

Another way to handle this is to synchronize generating units as soon as possible, as many generating units cannot operate without load this is another reason to connect load during the energization phase. In order to limit the reactive generation in the lines, some lines may not be energized in this phase hence the system will have fewer redundant paths.

When the transmission system is energized the focus is shifted towards connecting the remaining load as soon as the power plants are able to supply it given the available transmission capacity. The behaviour of the load is very uncertain during situations like this. One reason is that it is very hard to get statistics that are valid for the outage duration and weather, time of day and so on.

The level of load as a function of the time since reconnection is called cold load pickup [9]. Thermostatically controlled loads such as electrical heating, air condition and refrigerators will have higher consumption after a blackout than before. The reason for this is that a much larger percentage of the thermostats will be on than normal since the diversity in switching cycles has been lost. On the other hand many machines and industrial processes will need manual interventions before they can be restarted so they will have lower consumption when reconnected compared to what they normally have.

When closing a mesh in the grid it is necessary to check so the voltage phase angle difference over the circuit breaker is not too large. Otherwise the transient could initiate a new collapse and damage some equipment, especially the axis in generating units could be damaged due to high torque. In order to avoid this; the power flow often need to be changed somehow since the voltage phase angle difference is mainly due to the active power flow through inductive power lines [38].

Some degree of automation of the power system restoration task is already in common use. For example Svenska kraftnät uses automatic zero voltage detection and automatic restoration equipment (In Swedish: DriftUppByggnadsAutomatik, DUBA).

The automatic zero voltage detection opens all circuit breakers in parts of the grid that are not energized [3]. This simplifies the restoration since one object can be energized at a time without manually opening all the circuit breakers first.

The DUBA system of each circuit breaker tries to re-close the breaker if it was closed before the event and some predefined local conditions are met [2]. The operations performed by the DUBA operation are delayed between 10 and 300 seconds after the conditions are fulfilled and after the last DUBA initiated action in the substation. After large blackouts the DUBA will help with the energization of the first substations but much of the restoration need to be done manually [4]. This is due to the fact that the DUBA mostly tries to restore the state before the event directly with very limited ability to adapt to the current situation.

2.2.1 Geographical aspects

The type of power plants and their location relative to the loads will give different power system different characteristics.

Hydro power units are often located far away from the load centers since they need to be placed were the water flows are. Thermal power plants are often located closer to the load since there is more freedom when placing them even if then need for cooling water can give some restrictions.

Industry that consumes very much power is often located in areas with good power supply while the power supply is a very minor concern when placing most other loads such as homes and offices.

The load will be very weather dependent in areas with much electrical heating or air conditioning.

In large cities where it is not feasible to use overhead lines high voltage AC cables are used. These have very high capacitance so the reactive generation can be a major concern. One other concern is that they often need power in order to maintain the oil pressure that is needed for the insulation.

Neighbouring power systems are often interconnected with EHV AC (Extremely High Voltage Alternating Current) lines and then form one AC power system from a physical point of view even if they sometimes are regarded as separate systems from an administrative view point.

If the systems are separated by a significant distance of water, have different frequencies or there are some other reason to not connect them by AC lines then they can be connected by HVDC (High Voltage Direct Current). This gives a much looser coupling between the systems.

In some cases it is also economical to use HVDC in parallel to an AC system as reinforcement. Unlike AC-lines the active power flow and the reactive power can easily be controlled by HVDC-lines.

All these aspects will give different power systems very different characteristics and therefore the restoration strategy needs to be adapted to the specific case.

2.3 Power plant characteristics

Power plants have an important role in bulk power system restoration. Different types of power plants have very different characteristics, the most important difference is that steam turbine power plants have much longer starting times than for example hydro power and wind power.

2.3.1 Hydro power

Hydro power has very short starting time (a few minutes) and the active power can be ramped up and down very fast (100 %/minute). It also runs with high efficiency on partial load, and for this reason it is often used as regulating power. A drawback when using hydro power as fast frequency regulating power is that the power output has a non minimum phase characteristic, e.g. when opening the guide vanes to give more power, the power will temporarily decrease due to the acceleration of the water in the waterways. This is especially true in power stations with long waterways.

It has very low marginal cost but since a river has a limited amount of water and the water can be stored in dams it is common to assign a price to the water according to the expected electricity prices at later times. If the water flow is too large it can be necessary to spill the water. If there are more hydro power stations after each other the production must be coordinated so that they use each others' water.

There are also legal restrictions on maximum and minimum water flows and levels in order to not disturb the ecology too much and not flood areas downstream. Hydro power generators are also very good at regulating voltage.

2.3.2 Wind power

Different types of wind turbine generators have very different characteristics. Since power system restoration needs to be done even if there is no wind, restoration plans cannot depend on wind but when it is available it can be used.

In systems with much wind power there will be less conventional production ready to start (in a hot state) after a blackout when large wind production is forecasted than otherwise.

Early wind turbines were stall regulated with directly connected induction generators. These could not control active and reactive power, which make them harder to use in a power system restoration situation. Newer wind turbines often use doubly-fed induction generators (DFIG) controlled by power electronics or synchronous generators with electronic frequency converters which make it much easier to use them in a power system restoration situation. Modern wind turbines can be used for regulating both frequency and voltage by controlling the power electronics; which can be done in the order of milliseconds.

Since the marginal cost is very low it is very seldom economical to run on less than their maximum output at the current wind speed, but this can be done in a power system restoration situation. Most wind turbines normally start automatically about 10 minutes after they got stable voltage after a blackout. Some wind turbines can use the power electronics to act as a STATCOM[1] even when there is no wind this can help regulating the voltage during restoration.

2.3.3 Diesel engine

Diesel engines are mostly used as backup power, they are very fast to start and regulate and also cheap to build but expensive to run. In sizes up to about 1 MW they can be mobile. They often have black start capability based on a battery.

---

[1] A power electronic device that can continuously control its generation or absorption of reactive power.

2.3.4 Gas turbine

Gas turbines are used for larger units than diesel engines and take 5-30 minutes to start. They are fast to control and are therefore often used as regulating power and emergency power. They can often be started in a black net by using compressed air or battery. The marginal cost is high but they are cheap to build.

2.3.5 Steam turbine

Steam turbines are used in most nuclear and coal fired power plants and produce about 80 % of the world's electricity. They often have very long starting times since it takes long time to heat the boiler. A steam turbine can be in different states of readiness, cold (starting time 4-48 h), hot (starting time 30 minutes - 2 h), house load (starting time 5 minutes). The thermal inertia makes steam turbines rather slow at regulating their active power output, especially when the active power is increased.

Power plants based on steam turbines need rather much external power in order to start.

When disconnected from the net due to faults in the net many units try to continue supplying the internal power consumption (house load). In order to achieve this, the power to the turbine must be decreased rapidly, which is often done by letting steam bypass the turbine and go directly to the condenser. This is a sensitive process and in about 50% of the cases the unit will trip when transition to house load is attempted [4].

A hot restart is only possible if the unit gets outside power within a specified time after the blackout.

Nuclear power plants also have to follow strict safety regulations and sometimes need to get permission before starting. In the Swedish blackout in 1983 the first nuclear generating unit was synchronized after 10 h and all nuclear production was completely restored after about 80 h [1, pages 45-50].

Often a gas turbine is combined with a steam turbine in a combined cycle power plant. In a combined cycle power plant the heat in the exhaust gases from the gas turbine is used to produce steam for the steam turbine, this gives higher overall efficiency.

2.3.6 Combined heat and power

Gas turbines, steam turbines and diesel engines generate large amount of heat. Therefore it is often economical to use the heat for e.g. district heating. In industrial processes it is common to use steam, often it is more economical to produce it at higher pressure and temperature than needed and let it pass a steam turbine in order to generate electric power before it is used.

These power plants are called CHP, combined heat and power. They are often dispatched based on the heat demand and the electricity is a by-product. They can often not run if there is no way to use up the heat.

Buildings without electricity cannot absorb the heat even if it is cold weather. The pipe network and heat storage tanks in district heating can often absorb some hours of heat production.

2.3.7 Properties of distributed generation

Small scale distributed generation could in theory start and respond faster than large plants since small systems have faster thermal and mechanical dynamics in general. But often it is too expensive to equip small power plants with special control equipment for unusual situations such as restoration.

If the power plant has remote control it often will not work during power outages and also the remote control will often only be available to the plant owner and not the network owner or TSO. This can lead to a situation where distributed generating units will go online faster than desired in a sensitive stage of the restoration.

Also the large number of plants with different characteristics will make the planning problem harder. This will require that they are aggregated in the same way that loads are.

2.4 Historical development of power system restoration

Restoration was straightforward in the first power systems as they were built around a power plant that of course had black start capability. Since then the power systems have become more and more interconnected over larger and larger areas. Many newer power plants are not built to be able to start independently of external power.

The way the power system is operated has also changed. The first systems were to a large extent operated by direct manual control of the primary components. Then the operation was moved to local control rooms and these have been more and more aggregated. Now most national power systems are operated by computerized remote control (SCADA) from one or a few centralized control rooms.

This means that the number of people that operates the system has decreased. Even if the operation of the system has been heavily rationalized there will be a very high work load on the operators in a restoration situation. It also means that the operators do not have the same level of local knowledge that was common before. On the other side they will have much better overview of the system.

In Sweden the operators of the local control rooms had prepared rules for what to do in the event of a blackout. When these control rooms were closed some of these rules have been implemented in automatic systems and this is what we now call DUBA.

Another important change is the deregulation of the electric market in large parts of the world. This has led to the legal and administrative separation of network and production. This makes it harder to coordinate the operations in uncommon situations where the market cannot operate as intended.

2.5 Problem formulation

The research problem in this thesis is to design an algorithm that generates a complete sequence of actions to restore a power system after a collapse. This will be done for a static power system model. The initial state is not predetermined but a result of an unknown power system incident. The algorithm should not rely on prepared plans or rules that are specific to the power system or the restoration situation. This is important since the algorithm is intended for comparison of different power system configurations from a restoration perspective.

Chapter 3

Modeling and simulator

This chapter describes the models of the different system components and how the power system is simulated. The objective of these models is that they are easy to implement and computationally cheap while still giving results that are relevant from a power system perspective. The model is used by the restoration algorithm to test different actions and to get the resulting values such as frequency, voltages and currents. The model also generates a list of available actions in each state. The model contains checks to determine if a state is feasible which means that all components operate within their specifications.

3.1 Simulation

The power system is simulated with a power flow calculation after every action in the search algorithm. Much computation is saved by avoiding dynamic simulations for every step. If this algorithm should be used for real restorations the most promising actions should be checked with a dynamic simulation, which has not been implemented yet. Only the positive sequence is modelled; in other words, three-phase-symmetry is assumed. For a transmission system without faults this should be a reasonable assumption. The power flow calculation includes frequency calculation and limits on active and reactive power at generating units. This is necessary in order to model the production distribution during the restoration. The frequency calculation is developed from the concept of distributed slack bus. The frequency calculation assumes that all power units can be assigned a frequency response in

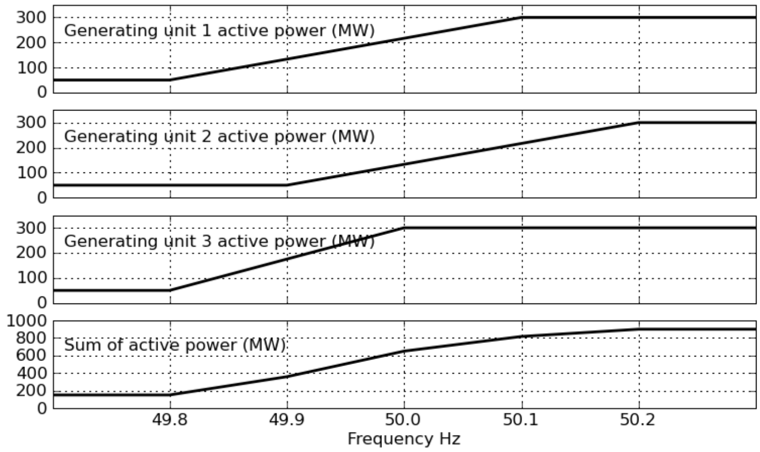


Figure 3.1: An example that shows how the frequency control efforts of several generating units are merged. All generating units in the diagram have a maximum power of 300 MW and a minimum power of 50 MW. Generating units 1 and 2 have a frequency response of 833 MW/Hz and power unit 3 has a frequency response of 1250 MW/Hz. Every Generating unit has a different set point for active power. The set point is the power obtained at 50 Hz. The frequency is calculated so that a balance exists between the production and the consumption including the active power losses.

MW/Hz that applies over the entire control range. Figure 3.1 illustrates how the MW/Hz characteristics of several generating units are combined.

The program also contains a complete topology calculation and handles several electrical islands running at the same time by performing a separate power flow calculation for each of them. Since only static power flows are calculated no proper synchronization checks can be done. The only check that is done when islands are synchronized is that the resulting power flow should represent a feasible state. The impedances of the components are independent of the deviation from nominal frequency. The power flow equations are solved with the Newton-Raphson method [23].

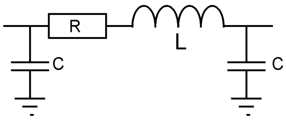


Figure 3.2: The model of a line, a π-link, so named because of the similarity of the diagram to the Greek letter π (pi). The model has series inductance (L), series resistance (R) and capacitance (C) to earth. Since three-phase symmetry is assumed, only one phase is modelled.

$$\begin{bmatrix} \Delta\mathbf{P} \\ \Delta\mathbf{Q} \\ \Delta\sum\boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} \frac{\partial\mathbf{P}}{\partial\boldsymbol{\theta}} & \frac{\partial\mathbf{P}}{\partial\mathbf{V}} & \frac{\partial\mathbf{P}}{\partial f} \\ \frac{\partial\mathbf{Q}}{\partial\boldsymbol{\theta}} & \frac{\partial\mathbf{Q}}{\partial\mathbf{V}} & (0...0)^T \\ 1...1 & 0...0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\theta} \\ \Delta\mathbf{V} \\ \Delta f \end{bmatrix} \tag{3.1}$$

Each power flow iteration solves the equation 3.1 to update angles by $\Delta\boldsymbol{\theta}$, voltages by $\Delta\mathbf{V}$ and frequency by Δf. $\Delta\mathbf{P}$ and $\Delta\mathbf{Q}$ are the mismatches in active and reactive power at each bus. Specifying $\Delta\sum\boldsymbol{\theta}$ drives the average of the bus voltage phase angels to zero, which is an alternative to selecting a reference bus. This is easier to handle when the topology changes than a reference bus. All power flow calculations are calculated from flat start.[1] The number of iterations in the power flow calculation has been limited to 50 and the tolerance for convergence are 10 kW active power error and 10 kvar reactive power error in each node. Incremental power flow calculations would decrease the time to calculate a power flow but would increase the memory requirements. Tripping constraints for protections etc. are not modelled. The nominal frequency is assumed to be the same throughout the power system. All connections and disconnections are made with explicitly modelled switches.

3.2 Power system components

3.2.1 Transmission lines

Transmission lines are modelled with a π-link with series resistance and reactance and only capacitance to earth. The capacitance is assumed to be symmetrically shared between the legs of the π-link; see Figure 3.2. The transmission lines have a highest permitted current. Transmission lines include both overhead lines and cables.

3.2.2 Transformers

Transformers are modelled as an ideal transformer in series with a π-link with series resistance and series reactance and with shunt reactance. Only real transformer ratios are modelled (no phase-shifting transformers). Transformers with more than two windings are not modelled. In this thesis no tap changers have been used. Transformers have a highest permitted current. It is assumed that the transformers can be energized from either side but in practice energization from the low voltage side is often avoided due to high inrush currents.

3.2.3 Shunt capacitors and reactors

Shunt capacitors and reactors are modelled as pure admittances. Any automatic control is assumed to be replaced by the presented algorithm during the restoration.

3.2.4 Loads

All the power consuming equipment connected to a power system is aggregated into larger objects called loads in simulations and calculations. This is done both since the net owner lacks information about details of the customer power consuming equipment and to simplify the calculations. At transmission level the load objects also include subtransmission and distribution nets and tap changing transformers. In reality the loads are the most complex part

---

[1] This means that all voltages are set to 1 p.u. and all voltage phase angles is set to 0 before the first iteration. The reason for this is that it is easier to handle for the search algorithm since values from all earlier net states need not to be stored.

of a power system. Transmission system operators normally do not reconnect loads themselves. Rather they tell the subtransmission system operators how much load they can connect in different areas. The power consumption after a blackout will be different from what it would have been if the blackout had not occurred since some equipment takes time to start and other equipment is controlled by thermostats and will consume more than normal. This is called cold load pickup [9].

In order to model cold load pickup some form of dynamic simulation would be needed. In this model loads are modelled only as constant active and reactive power. The full amount of load at each point in the transmission system is divided into a number of small parts that are reconnected by closing of switches.

3.2.5 Generating units

Generating units in the model can be on or off; black start units and plants that survived the blackout are on in the initial state of the model. When a generating unit that is off is energized from the net it will switch to the on-state. Generating units that are on will not be turned off even if they are disconnected. The model parameters and state variables can be seen in table 3.1 respectively 3.2. Active power is adjusted between P_{min} and P_{max} for frequency control. P_{set} is the active power at nominal frequency. See Figure 3.1. Reactive power is adjusted between Q_{min} and Q_{max} by the automatic voltage controller. V_{set} is the voltage with zero reactive power. P_{set} can be changed between P_{min} and P_{max} in 10 steps. V_{set} can be changed in steps of 0.01 p.u. as long as the reactive power is within its limits. When reactive power is above Q_{max}, V_{set} cannot be increased and when reactive power is below Q_{min}, V_{set} cannot be decreased. No changes can be made when the generating unit is off.

3.2.6 Power system node

A power system node is any location where two or more components are connected. All nodes have a highest and a lowest permitted voltage and frequency. A base voltage has also been defined for each node.

Table 3.1: Parameters in the generating unit model

P_{max}	Highest possible active output power from the unit
P_{min}	Lowest possible active output power from the unit
Q_{max}	Highest possible reactive output power from the unit
Q_{min}	Lowest possible reactive output power from the unit
dP/df	Frequency control characteristic, MW/Hz. Valid as long as the active power is between P_{max} and P_{min}. This is also known as the frequency response of the frequency controller. See Figure 3.1.
dQ/dV	Voltage control characteristic. Valid as long as the reactive power is between Q_{max} and Q_{min}.

3.2.7 Switch

A switch can be placed between two nodes. Switches have two states: on (closed) and off (open). Switches are modelled as capable of switching all occurring currents on and off. Open switches can only be closed if one side is energized; this simplifies the algorithm by avoiding to trying to close all switches that are not affecting the current power flow. All closed switches can be opened.

3.2.8 Faults

All components are assumed to be working properly. Any faulty components are assumed to be removed from the model before the simulation is started.

3.2.9 Substations

Substations are defined only as those components that can be connected to each other without impedance between them. This means that two groups of switchgear with transformer/transformers between them are regarded as two substations with different names. This has no relevance for the power flow calculation but is used when generating statistics for a net state.

Table 3.2: Variables in the generating unit model

P_{set}	Set point for active power, the power delivered by the power unit at a nominal frequency. P_{set} must be between P_{max} and P_{min} and can be adjusted during operation. The algorithm makes the adjustments in steps of 10 percent of the difference between P_{max} and P_{min}.
V_{set}	Set point for the voltage on the connection node of the power unit. This is the voltage that gives a reactive power of 0 Mvar and can be adjusted during operation. The algorithm makes the adjustments in steps of 1 percent of the nominal voltage.
On	Indicates whether or not the unit is in operation. Only islands that contain operating generating units need power flow calculation. Generating units that are or has been connected to a "live" island are assumed to be in operation.

3.3 Action sequence and states

The starting point is a power system with a specific (operating) state. From this state, the algorithm must find a sequence of actions that finally connects all loads. All the intermediate states in this sequence must be *feasible*. There may be several energized subsystems (islands). A feasible state is a state in which the voltage and frequency at all energized nodes are within set limits and the currents in all lines and transformers are below their thermal limits. In order to simplify the algorithm only discrete actions are modelled, even when changing continuous values such as active power set points. Available actions are shown in Table 3.3.

Switches in non-energized parts of the power system are assumed to be off (open) in their initial state. If automatic zero voltage protection is installed, as it is in Sweden, this is a very reasonable assumption. Since only actions on closed switches and energized components are considered, the search only needs to cover parts of the power system that have been energized at some time during the restoration. For a sequence of actions to be a valid solution to the problem, all intermediate states must be feasible.

A state is described by:

Table 3.3: Actions available to the algorithm.

Type of component	**Type of action**	**When applicable**	**Description**
Switch	Switch_open_action	If the switch is closed	Open the switch
Switch	Switch_close_action	If the switch is open and at least one side of it is energized	Close the switch
Generating units	PV_inc_P10	If the generating unit is energized and P_{set} is not at the upper limit	Increase the set point for active power at the generating unit by 10 % of the control range
Generating unit	PV_dec_P10	If the generating unit is energized and P_{set} is not at the lower limit	Decrease the set point for active power at the generating unit by 10 % of the control range
Generating unit	PV_inc_V1	If the generating unit is energized and Q is not at the upper limit	Increase the voltage set point by 1 %
Generating unit	PV_dec_V1	If the generating unit is energized and Q is not at the lower limit	Decrease the voltage set point by 1 %

- Switch states
- Voltage set points of generating units.
- Active power set points of generating units.
- The set of generating units that are in the On state.

Since the load in this model is determined only by the switch states, no extra state is needed for the load.

3.4 Time

In order to calculate the energy not delivered and plot results against restoration time an estimate of the time for each action is needed. This is arranged in the following approximate way. Increasing and decreasing active power set point by 10% on thermal non black start generating units are estimated to take 5 minutes. All other actions are estimated to take 1 minute. Since in real situations several actions can be taken in parallel by different operators, these estimated times are divided by 3 before they are used in the calculation. This is of course a very coarse model of the time consumption during the restoration. A more refined model would be needed to explicitly model the used parallelism and the time dependent behaviour of the components.

Chapter 4

Test system

Nordic32 shown in Figure 4.1 is selected as test system. Nordic32 is a fictitious power system model with 32 substations. It has been designed to broadly represent the Nordel system. Nordic32 is chosen because it is a well known test system so the data and results can be published freely and has enough complexity to give interesting restoration plans without being too complex. As a comparison this model is considered to be about one-third the complexity of the real power system in western Denmark (Jutland and Funen) down to and including 150 kV.

4.1 Adaptations and details

Nordic32 was originally proposed as a test power system by Svenska Kraftnät in a CIGRE report on Long term dynamics [7]. In that version it is a dynamic net model without station topology, but Svenska Kraftnät later added station topology, among other things, so that it could serve as an example for the ARISTO real-time simulator [14]. The ARISTO version of Nordic32 has been used in this project, with the following minor changes: The substation topology has been simplified to include only switches that are normally closed. Double busbar double breaker substations have been converted to single busbar substations. These changes will make the algorithm simpler. All the transformers have been fixed at the nominal tap ratio. The transmission system loads have been divided into several smaller loads between 8 and 60 MW. Since the loads can only be connected or disconnected it is necessary to be able to connect load in smaller steps. An extra 540 MW

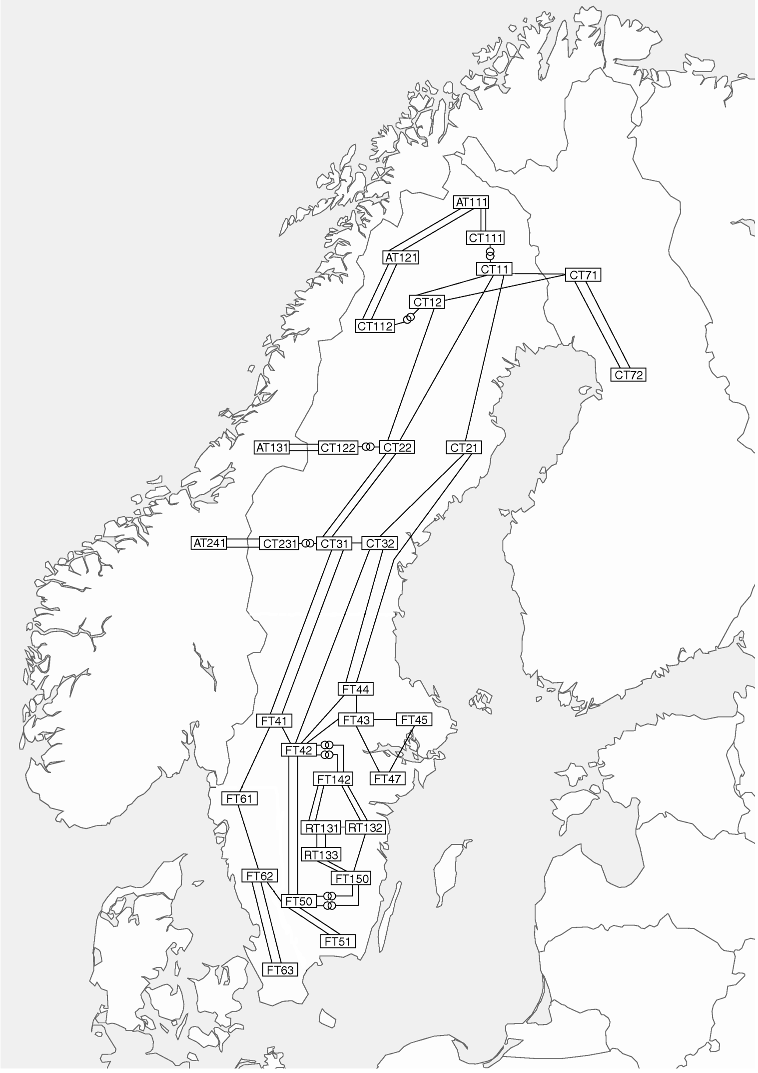


Figure 4.1: Nordic 32, a fictitious power system model with certain similarities with the “Nordel” power system. The diagram shows the stations with their designations and all lines and transformers with approximate fictive geographic positions.

generating unit with black start capability has been added in the South[1]. This shall represent gas turbines and other peaking generating units that are not included in Nordic 32. The model has 545 switches, 10 900 MW of load and 36 generating units. There are 17 256 MW generating capacity of which 10 866 MW are hydro power in the North the rest are thermal power in the South. Only the hydropower and the black start unit in the South are used for frequency control. The voltage regulators have almost perfect voltage regulation, the full reactive control range will be used for a voltage deviation of only 0.01 %.

The complete model file for this system can be seen in appendix B.

4.2 Visualization

To be able to easily present an overview of interesting power flows, a graphic representation of power flows has been developed. Given the coordinates of the substations, transformers and lines the visualization of a power flow is generated automatically. The visualization was developed to be suitable for viewing in black and white such as in this thesis. The visualization has also been of great value when running restoration searches interactively and when debugging the program. Table 4.1 and Figure 4.2 describes how the various parts of the visualization are interpreted. The normal state for the Nordic32 power system can be seen in Figure 4.3. In order to simplify the view the connections inside a substation are not shown. The shown production and consumption are the sum for the substation. Another example of a visualization of power flows can be seen in the commercial power flow solver Power World [32].

4.3 Test cases

The three different starting states shown in the Figures 4.4 - 4.6 are chosen to test the algorithm under different conditions.

- Restoration from the North after a complete blackout. Only one gen-

---

[1] This was done in order to be able to do black start in the South of the model. All the existing generating units in the Southern part of Nordic32 has a minimum active power generation and can therefore not be black started in the model. Another solution had been to modify one of the existing generators, this has not been tested.

Table 4.1: Representation of different variables in the oneline schematics. The graphical representation can be seen in 4.2.

Nr	Variable	Graphic representation
1	Active power flow	Size of black arrowhead in the middle of line.
2	Reactive power flow	Size of grey arrowhead in the middle of line.
3	Line or transformer current in p.u.	Thickness of the line is a constant plus a term proportional to the current.
4	Total load, active power	Area of unfilled circle
5	Current load, active power	Area of filled circle
6	Total generating capacity	Area of unfilled square
7	Synchronized maximum generating capacity	Area of outer grey filled square
8	Current generation	Area of black filled square
9	Synchronized minimum generation	Area of inner grey filled square. This is zero for hydro power.
10	Voltage	Arrow in the half circle to the right, the arc represents 1 p.u. and deviations are amplified five times. When all parts of a substation are not connected there are more than one arrow. The relative voltage phase angles are directly given by the angle of the arrows.
11	Maximum reactive support, capacitors and generators.	Length of upper part of the vertical grey line to the right in each substation.
12	Current reactive support or absorption, capacitors, reactors and generators.	Length and direction of the vertical black line to the right in each substation.
13	Maximum reactive absorption, reactors and generators.	Length of lower part of the vertical grey line to the right in each substation.

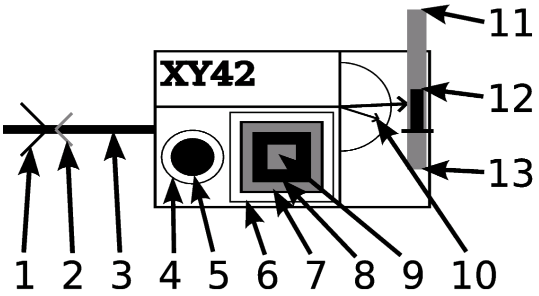


Figure 4.2: Power flow visualisation elements. For interpretation see table 4.1.

erating unit in the equivalent of Northern Finland (CT72-G1) is black started. This is an extreme case of the traditional restoration strategy in the Swedish system with restoration from the North. It is intended for testing of full restoration from one point.

- Restoration from the North and the South after a complete blackout. The extra black starting generating unit in the South (FT63-G11) is started in addition to the generating unit used in the previous example. This is intended for testing of the algorithm in a situation with multiple islands.
- Restoration after a partial blackout. This case is intended to be somewhat more realistic in that it is only a partial blackout. This case was created by turning off the Southern part of the system and three generating units in CT72, otherwise it is identical to the normal state in Figure 4.3.

4.3.1 Model parameters

The test system has a number of global parameters that affect the behaviour of the system. Changing these will be used to test the sensitivity to changes in different parameters. All the parameters have a default value that will be used when not specified otherwise. These parameters are listed in table 4.2.

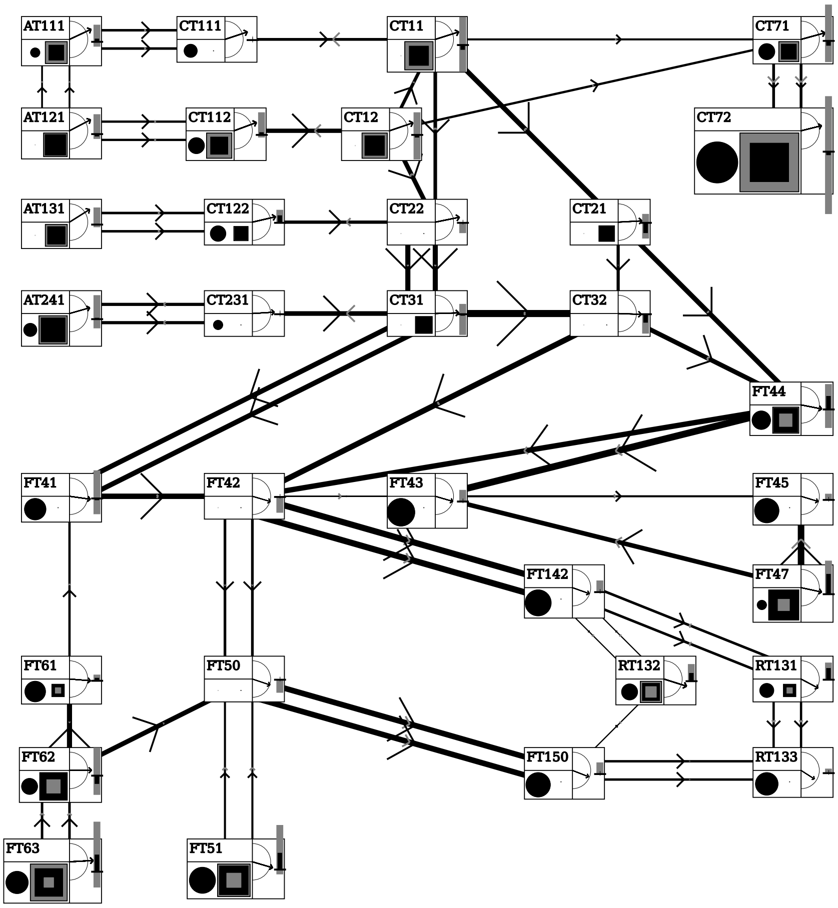


Figure 4.3: Power flow solution for the base case in Nordic 32. For interpretation see table 4.1. 400 kV substations have names with 2 digits. 130 kV substations have names with three digits and the first are 1. 220 kV substations have names with three digits and the first are 2. Lines between substations represent power lines if the substations have the same voltage, otherwise they represent transformers.

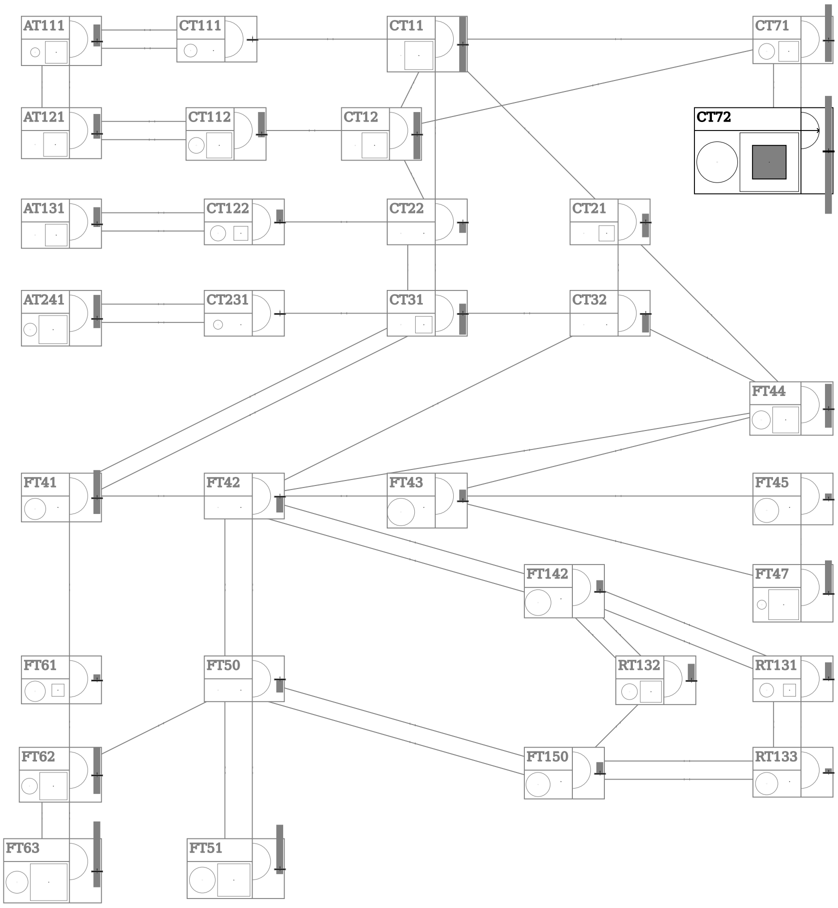


Figure 4.4: Initial state with one island in the North consisting of only the substation CT72.

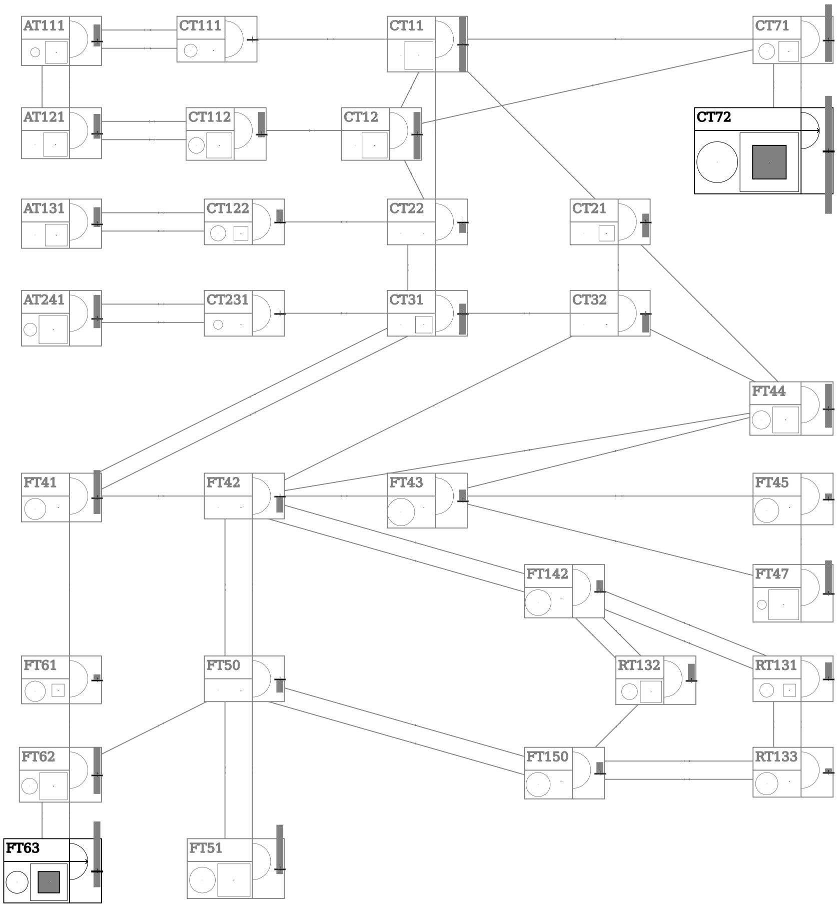


Figure 4.5: Initial state with one island in the North and one in the South with the substations CT72 and FT63.

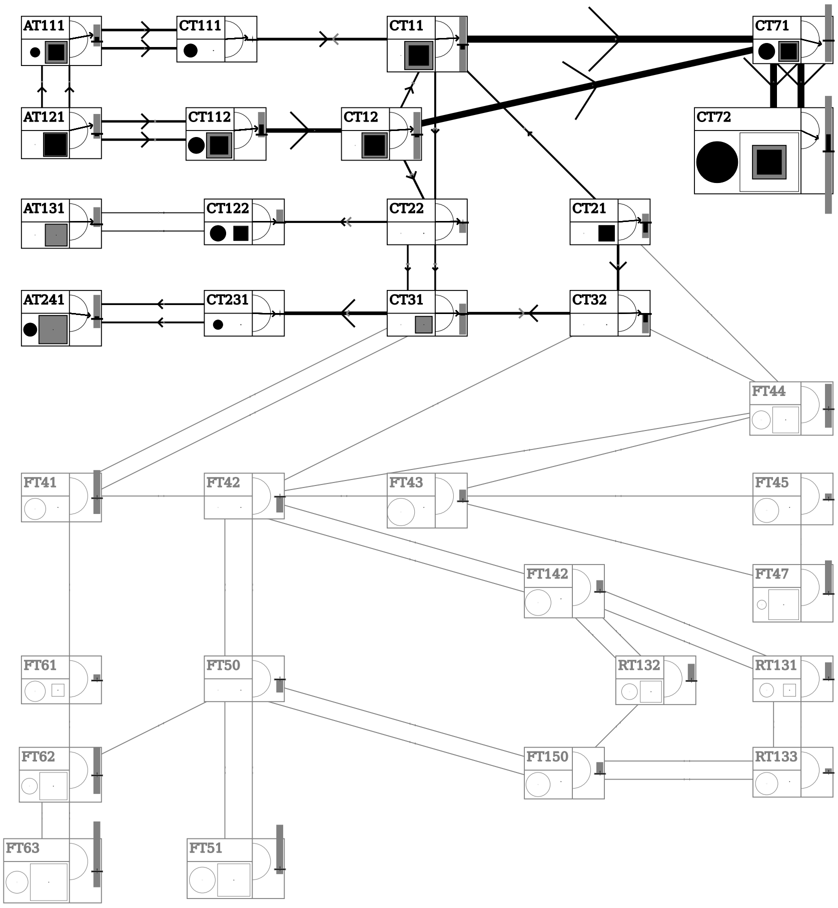


Figure 4.6: Initial state with the Northern half of the system running, somewhat similar to the situation after the collapse 1983.

Table 4.2: Parameters of the test system

Parameter	**Description**
Allowed frequency deviation	If the frequency in any part of the system deviates more than this amount from nominal frequency (50 Hz) the state is not considered feasible. Default value is 0.5 Hz.
Highest allowed voltage	If the voltage in any part of the system exceeds this value the state is not considered feasible. Default value is 1.1 p.u.
Lowest allowed voltage	If the voltage in any part of the system is lower than this value the state is not considered feasible. Default value is 0.9 p.u.
Generation factor P	Maximum and minimum active generation is scaled by this value for all generators. Default value is 1.
Generation factor Q	Maximum and minimum reactive generation is scaled by this value for all generators. Default value is 1.
Generation P_{init}	The initial active power set point of the generating units as fraction of $P_{max} - P_{min}$, $P_{init} = 0 \Rightarrow P_{set} = P_{min}$, $P_{init} = 1 \Rightarrow P_{set} = P_{max}$. If this does not match the load there will be a frequency deviation. Default value is 0
Generation V_{set}	Default initial voltage set point in per unit for the generating units. Default value is 1
Load factor P	All active load is multiplied by this factor. Default value is 1
Load factor Q	All reactive load is multiplied by this factor. Default value is 1
Load step size factor	All loads are divided into load steps of at maximum this value multiplied by the default step size for the load. Default value is 1
I limit factor	All current limits on transmission lines and transformers are multiplied by this value. Default value is 5. This almost means that current limits are disregarded by default.
Load on factor	The percentage of the loads on each busbar that will be connected to that busbar from the start. This can be regarded as loads without zero voltage protection. Default value is 0
Random lines missing	This number of lines will be chosen randomly for removal. This can be used for simulating random known line faults in the system. Default value is 0.

Chapter 5

Characterization of the problem

The restoration problem is a planning problem[1]. What characterizes such problems is that, given an initial state, a sequence of actions is sought such that a more desirable state is reached.

In the restoration problem there is no specific target state. The aim is to arrive at any state that satisfies certain requirements, for instance that every customer's electricity supply has been restored and that the state must fulfil certain requirements on the voltages and frequency.

The main difference between this problem and the planning problems, normally studied in the field of artificial intelligence, is that the conditions that govern which actions are permitted cannot be expressed as explicit logical expressions. Instead a simulation (static or dynamic) needs to be done, after which it is possible to check whether variables, such as voltages and frequency, are within permitted limits.

Consequently, traditional planning algorithms such as "partial order planning" [35, pages 387-395] cannot be used. It is also difficult to use the technology to search backwards from the target state to the initial state [35, pages 384-386], since there is no fixed target state. For these reasons, a more direct method is used, namely to search for a sequence of actions using the initial state.

---

[1] Planning problems are here used in a computer science context. In a power system context it is common to differ between operational planning and expansion planning. The difference is that the operational planning deals with how to run the system the coming hours and days, while the expansion planning deals with how to add new components to meet future demands over the coming years and decades.

5.1 NP-hardness

The power system restoration problem can be shown to be NP-hard. This has been asserted in papers without proof or citations [21]. That the power system restoration problem is NP-hard means that it belongs to a category of problems for which it is believed that no *efficient solution methods* exists. A well known example of a NP-hard problem is the travelling salesman problem.[2] However, it has also not been shown that *efficient solution methods* do not exist.

The expression *efficient solution methods* means methods whose calculation time is guaranteed not to increase asymptotically faster than a polynomial in the size of the problem, for example expressed in number of components n in the electric power system, as the size of the problem increases. For example, if the computation time for a solution method grows with n^{1000} it is considered an *efficient solution method* while if the computation time grows with 2^n is not considered to be an *efficient solution method*.

One consequence of this is that it is unrealistic to have as a goal an algorithm that is guaranteed to solve the problem in reasonable time, especially when searching for a solution that is in any sense optimal. It is therefore necessary to focus on finding algorithms which, in as large a proportion of realistic cases as possible, find solutions that are as good as possible in acceptable time.

Since the power system restoration problem is so complex, only a simplified version is used in the proofs below. This does in fact make the proofs stronger since any reasonable power system restoration model can model this simplified version. It is possible to define subsets of the power system restoration problem that can be guaranteed to be solvable in polynomial time, but it is hard to see how such a subset could include all or most of the non trivial practical cases in any useful way[3].

---

[2] The travelling salesman problem is the problem facing a travelling salesman that need to visit each city in a set of cities and want to find the order he/she should visit them in order to minimize the travelled distance. The distance between each pair of cities is assumed to be known.

[3] It is always possible to state that all practical power system restoration problems have less than some large number of components and then the computation time can be shown to be less than some very large constant but this gives no useful limit.

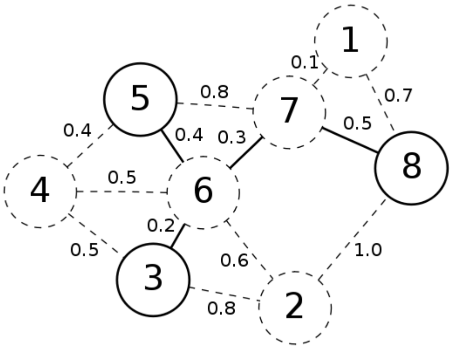


Figure 5.1: Example of a Steiner tree, the nodes 3, 5, 8 is required to be connected and the minimal Steiner tree are shown in bold.

5.1.1 Steiner tree in graphs

The normal way of proving that a problem X is NP-hard is by proving that any instance of a known NP-hard problem could be solved in polynomial time if X could be solved in polynomial time.

Since this would contradict the assumption that NP-hard problems can not be solved in guaranteed polynomial time it is assumed that X can not be solved in polynomial time.

Minimum Steiner tree in graphs is an NP-hard problem [10, page 395]. Given an arbitrary graph[4] where each edge has an associated cost, the problem is to decide if there exists a subset of the edges and vertices that form a tree[5] that connects all vertices in a given subset of the vertices with a total cost of the edges in the tree below a given value c. For an example see Figure 5.1.

Given an arbitrary instance of the Minimum Steiner tree in graphs problem there are different ways to create an instance of the power system restoration problem such that the solution of this problem gives the answer to the Minimum Steiner tree in graphs problem.

Two such transformations will be given. In both cases the graph is directly

---

[4] A graph in a computer science is a number of vertices connected by edges.

[5] A tree in a computer science is a number of vertices connected by edges in such way that there are no loops.

mapped to the power system, nodes and edges correspond directly to busbars and power lines. There is only one generator in the system. The set of nodes that need to be connected, correspond to the busbars with loads and the generator busbar.

Reactive limits

In this case the associated cost of an edge is translated into the reactive power generation of the corresponding power line and the maximum cost c corresponds to the maximum reactive absorption of the generator. All line impedances are assumed to be zero and all loads are assumed to be purely active power. If all limits and conditions other than the reactive limit on the generator are assumed to be set with large margins, then this system can be restored if and only if there exists a Steiner tree with lower cost than c. This proves that the power system restoration problem is NP-hard.

Since real power systems are built to be able to absorb the reactive power from the lines this can seem to be unrealistic but in a real situation there can be component failures. This could also be thought of as whether the load can be restored before all generating units have been started.

Restoration time

In this case it is assumed that there are no stability problems in the system so only the topology is interesting. Each power line has an associated time needed to connect it[6]. This time corresponds to the cost of the corresponding edge.

If an optimal[7] power system restoration sequence can be found, the corresponding minimum Steiner tree in graphs problem can be solved by comparing the restoration time with the maximum Steiner tree cost c. This again proves that the power system restoration problem is NP-hard.

---

[6] In the model used in this thesis where the cost of all switching operations is constant this could be modelled by replacing the power line by a series of power lines that have switches between them.

[7] Optimal here means shortest possible restoration time.

Discussion

There exist polynomial time algorithms that can calculate an approximation of the minimum Steiner tree in graphs so that it is guaranteed to find a Steiner tree that has a cost that is no more than 1.55 times the minimum cost [34].

Since the full power system restoration problem is much more complex it is not likely that such approximation algorithms can be found for power system restoration. The reason the full power system restoration problem is more complex is that it need to take into account factors such as voltage drop and the interaction between active and reactive power flows.

Chapter 6

Search algorithms

In order to find a sequence of actions that restores the power supply, a search algorithm is used. The algorithm is designed so that, given an arbitrary state, it attempts to find a sequence of actions that leads to a better state. Search algorithms are an important part of computer science and artificial intelligence.

A search algorithm is used to test different solution candidates to a problem in a systematic way. This is useful when it is easy to test if a proposed solution is valid but it is hard to generate a valid solution directly. Since the goal is to find a sequence of actions or in other words a path, a path finding algorithm[1] is needed. The environment in which the path exists is the state space graph, where every possible state in the power system is represented by a vertex in the graph and all actions in each state are represented by a directed edge to the resulting state. This graph is enormously large; fortunately it need not to be stored, only the relevant parts are generated when needed. Every action has an associated cost; the algorithm tries to minimize the sum of the cost of the actions in the action sequence. The search proceeds as a tree where every choice in each node generates a new node; see Figure 6.1. In this chapter the following terminology is used:

State A state in the state space graph, in the power system restoration this means a state of the power system. Denoted by S.

---

[1] The same algorithms can be used both when finding a physical path such as planning a tour using public transportation and in more abstract state spaces such as in power system restoration.

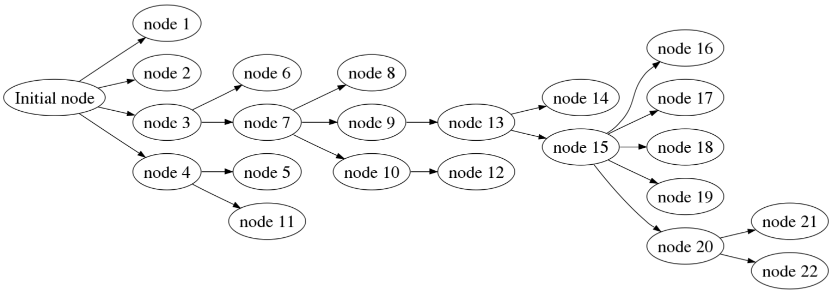


Figure 6.1: Example of a tree structure of nodes and actions, the arcs represents the actions. Examples of actions: closing a switch or increasing the set point for active power production by 10 %.

Node A node in the search tree, each node has a unique path or sequence of actions from the initial state to the state corresponding to this node. Several nodes can represent the same state if there are several paths to that state. Denoted by N. When referring to a node in the power system this will be stated explicitly to avoid confusion.

Action An action can be applied to a state and this will lead to a new state. Denoted by A.

Path A sequence of states and actions that goes from one state to another state.

The search algorithm works much like a chess computer; the difference is that in this case there is no opponent and because of this different search algorithms are used.

Search algorithms can be categorized as uninformed and informed. Uninformed search algorithms do not use any application specific information; they only need to know what actions are available in each state and the cost of the actions. Therefore, they need to test many paths until a path to a goal is found. For any reasonably large power system restoration problem this would be extremely slow. Informed search algorithms use some sort of heuristics to find the most promising of the tested paths and extend them. The heuristics are often in the form of an evaluation function that generates a value for each path. This value indicates which path that is most promising. With a good evaluation function the search time can be reduced by several orders of magnitude. Search algorithms can also be divided in online algorithms and offline algorithms. Offline algorithms calculate the whole sequence of actions from the start while online algorithms use information collected during the execution of the actions to decide what to do next. Online algorithms are useful for handling uncertainty. Since no uncertainty is modelled here an offline algorithm is used. For real power system restoration an online algorithm would be needed. An online algorithm can be created by rerunning an offline algorithm with the current state as the initial state during the execution of the action sequence.

6.1 The A^* algorithm

The algorithm used for the power system restoration in this project is based on an informed search algorithm called A^* [17]. In order to solve a path finding problem with A^* an initial state and four application specific functions are needed. These are:

Available actions Generates a list of available actions in a given state.

Resulting State Calculates the resulting state after a given action is applied in a given state. It also calculates the cost of the action.

Estimated cost to reach the goal from the current state, $h(S)$ Returns an estimate of the cost for the shortest path from the state S to a goal state. Can also be referred to as $h(N)$ where N is a node in the search tree that corresponds to the state S.

Is goal Test if a given state is a goal state, in some cases this is equivalent to $h(S) = 0$.

The sum of costs for all actions in the path to the node N is denoted $g(N)$. A^* uses $f(N) = g(N) + h(N)$ to guide the search, $f(N)$ is an estimate of the cost of the shortest path from the initial state to a goal state such that the path begins with the path to N.

A^* begins with one set of nodes called OPEN that only contains the node for the initial state and one empty set called CLOSED. The OPEN set will contain nodes that have untested actions or actions that need to be re-evaluated. The CLOSED set will contain nodes where all actions have been evaluated. The following steps are repeated until a path to a goal is found or the OPEN set is empty which indicates that no path from the initial state to a goal exists:

1. Select the node with the lowest value of $f(N)$ from the OPEN set, this node will be called N_n and the corresponding state S_n. If the OPEN set is empty then there does not exists any path from the initial state to any goal.
2. If S_n is a goal state then exit and return the path to N_n.
3. For each available action in node N_n do:
    (a) Calculate the resulting state after applying the action. This new state is represented by a new node in the search tree.[2]
    (b) If a node with the same state as the new node is not already in the OPEN or CLOSED set then add it to the OPEN set.
    (c) If a node with the same state as the new node is found in the OPEN or CLOSED set and the new path gives a lower cost than the previous then remove the old node and add the new node to the OPEN set.
4. Move the state S_n from the OPEN set to the CLOSED set.
5. Go to step 1.

Generally an $h(S)$ function that gives larger values will result in a narrower faster search. If $h(S)$ is exactly the cost of reaching a goal given any state S and all values of $f(N)$ are unique only states on the optimal path will be expanded.

6.2 Optimality

It can be shown that A^* will find the shortest path to a goal if the heuristic function $h(S)$ never overestimates the cost of the cheapest path to the goal

---

[2] If a node with the same state has previously been expanded then this could be replaced by looking up the resulting state in a hash table in order to save calculations.

[17]. This is called an admissible heuristic. It can also be shown that if $h(S)$ is consistent[3], then no algorithm that is guaranteed to find the shortest path can expand fewer nodes if it is not given any extra application specific information [17]. One problem with A^* is that the whole search tree needs to be stored; in many applications this requires too much memory. In order to handle this there are modified variants of A^* such as SMA^* that works like A^* until the memory is full and then starts deleting the worst leaf nodes, if needed these can be regenerated later so the algorithm will always[4] return the same path as A^* but can save memory at the cost of more computation.

6.3 Example

The small state space graph shown in Figure 6.2 will be used to illustrate how A^* works. The heuristic in Figure 6.2 is an example of an admissible heuristic. The search tree generated by A^* for this graph can be seen in Figure 6.3. The arrows represent actions and the numbers on the arrows represent the cost of the action. Nodes that represent the same states but are on different paths are connected by a dotted line. The name of the state is given at the upper row in each node. On the lower row are three fields separated by /, these are:

1. Node number or number of nodes found before this node during the search.
2. The value of $f(N) = g(N) + h(N)$.
3. The status of the node when the search is finished. The abbreviations are:

   O The node is in the OPEN set.

   C The node is in the CLOSED set.

   W The node was worse than an earlier found node in the OPEN or CLOSED sets for the same state.

   RO The node was removed from the OPEN set when a better path to the same state was found. (Step 3c in the algorithm)

---

[3] This means that $f(N)$ never decreases along any path.

[4] Given of course that the path to be found fits in the given memory.

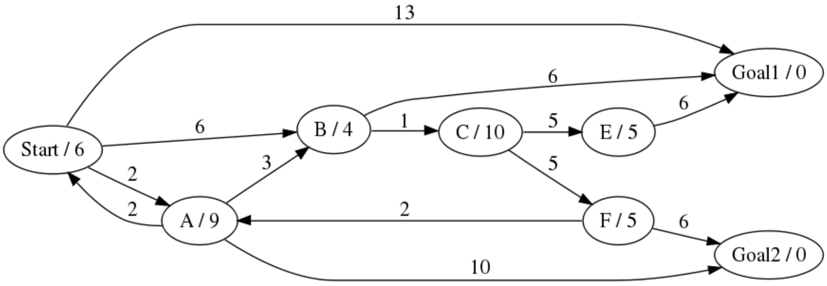


Figure 6.2: Example of a state space graph, the arrows represent actions and the numbers on the arrows represent the cost of the action. The name of the state and the value of $h(S)$, the estimated cost of reaching a goal from that node, are given in each state. The shortest path from start to a goal is Start – A – B – Goal1 and has a cost of 11.

RC The node was removed from the CLOSED set when a better path to the same state was found. (Step 3c in the algorithm)

The table 6.1 shows the progress of the search for each iteration. The algorithm does not need to visit the states E and F since they cannot be on an optimal path unless $h(C)$ is an overestimate[5]. Three paths to Goal1 are found and one to Goal2, the search continues until the optimal path is found. It finds a path back to the start state (Start -> A -> Start) but correctly detects that it has already a better path (The empty path) and stops searching that path. All the directly following actions are always tested in sequence. The cost of going from A to B is 3, this is less than $h(A) - h(B) = 5$, this means that $h(S)$ is not consistent and this is the reason that the suboptimal path to B is expanded before A on the optimal path. When the new path to the state B is found in node 7 the “children” Goal1 and C get updated so $g(N)$ and $f(N)$ take the new path in to account.

---

[5] The cost of reaching C is 6 and the estimated cost of reaching a goal from C is 10 so the estimated cost of reaching a goal through any path that includes C is 16 while the cheapest found path costs 11.

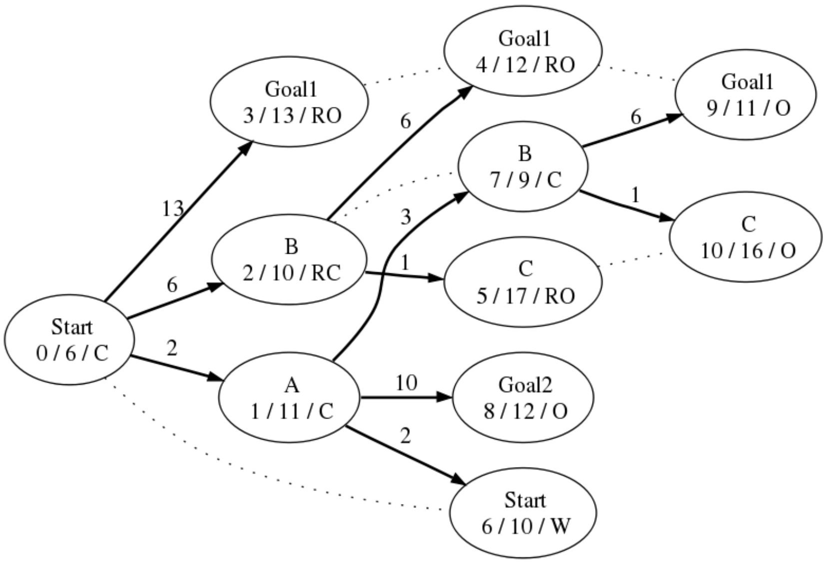


Figure 6.3: Example of a search tree generated by A^* for the graph in Figure 6.2. The arrows represent actions and the numbers on the arrows represent the cost of the action. Nodes that represent the same states but are on different paths are connected by a doted line.

Table 6.1: The content of the OPEN and CLOSED set in each step in the iteration. Each node is represented by the name of the state, the value of $f(N)$ and the value of $g(N)$

OPEN set	**CLOSED set**
(Start, 6, 0)	
(B, 10, 6) (A, 11, 2) (Goal1, 13, 13)	(Start, 6, 0)
(A, 11, 2) (Goal1, 12, 12) (C, 17, 7)	(Start, 6, 0) (B, 10, 6)
(B, 9, 5) (Goal2, 12, 12) (Goal1, 12, 12) (C, 17, 7)	(Start, 6, 0) (A, 11, 2)
(Goal1, 11, 11) (Goal2, 12, 12) (C, 16, 6)	(Start, 6, 0) (A, 11, 2) (B, 9, 5)

6.4 Power system restoration

Since the power system restoration problem requires so much computation in order to evaluate each node, the limiting factor is the computation time, not memory so in this application SMA^* is not needed.

No heuristic function that give estimates high enough to limit the number of expanded nodes to a reasonable number while still being admissible is known. Therefore a non admissible heuristic is used and it can not be guaranteed that an optimal path will be found.

An example of an admissible heuristic function is the total cost of closing a switch to every load that is not connected to anything. This could be improved by adding the cost of connecting a minimum set of generating units and the cost of increasing the active power set point. But such heuristics will not give any guidance in how to avoid voltage collapses and limits in e.g. voltage. Therefore such problems will be handled only by trial and error which will make the progress very slow.

In most states there are several hundred actions, this makes it too expensive to do a full power flow calculation in order to evaluate the result of all actions in each expanded node. Therefore the evaluation is divided into two steps. When a new node is reached, the node in itself is evaluated by an evaluation function $f_2(N)$ and all the actions of that node are evaluated by a very fast and simple function $f1(N, A)$, where N is the node and A is the action. When selecting a node to expand, the node with the lowest value of $f_2(N)$ is selected. Rather than calculating the result of each action, only one action is tested. The action to test is chosen randomly[6] from the best actions according to $f_1(N, A)$.

By testing it was found that choosing the action from the best 3 $\%$ of the actions gives good results but the exact value is not critical.

The resulting node of this action is evaluated together with its actions as described above. The evaluation function $f_2(N)$ includes calculating a power flow for the state. This means that for each iteration in the algorithm only one action is tested so the parent node must remain in the open set until it has no untested actions.

If the search continues until the node with the lowest value of $f_2(N)$ cor-

---

[6] The random choices are of course not truly random but are the result from a pseudo random number generator. The Mersenne Twister algorithm is used as pseudo random number generator.

responds to a goal state it will take too long time so instead the search stops as soon as a goal state is found.

Since there is no guarantee that there exists a path from the initial state to a goal or that it will be found within reasonable time even if it exists, a limit on the number of nodes to evaluate is needed. Since little improvement is seen in the test runs after about 2000 nodes we set the maximum number of nodes to 3000 after which the search will be aborted. The maximum number of nodes to evaluate depends of course on the power system that shall be restored.

The reason for choosing the action to test randomly from the most promising is that if no path to a goal is found in a search then a new search can be run with a new and hopefully better result.

The high number of states and available actions makes the probability of finding multiple paths to identical states low. Therefore the check for identical states in step 3 in the algorithm description has not been implemented. In step 3 the resulting state is just added to the OPEN set. This saves some computation but could on the other hand lead to recalculation of already calculated states in rare cases. If the power system model is extended with time dependent models of loads and power plants, the probability of finding multiple paths to identical states will become even lower since the timing of actions will affect the state.

Figure 6.4 and table 6.2 shows the search for a path in the example graph in Figure 6.2 without checking for duplicate states.

The evaluation functions have a number of parameters that must be given a value. The parameters are denoted p_n where n is an integer.

In order to have some starting point when studying variations in these parameters, in chapter 8, all parameters are given a default value. These values have been found by estimating the size of each term and the relative importance of the term. The chosen values were then refined by testing variations around these values. The default parameters are chosen to contain only one digit out of 1, 2, 3 or 5 in different decimal position or to be zero in order to show that the exact value are irrelevant.

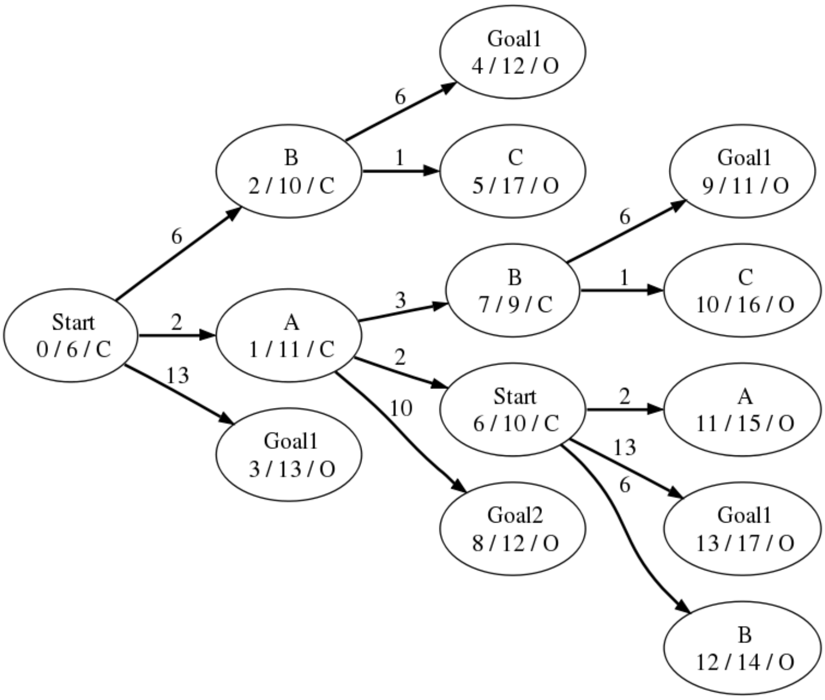


Figure 6.4: Identical to Figure 6.3 except that the check for duplicated states has been removed. When the start state is reached in node 6 parts of the search tree are duplicated.

Table 6.2: The content of the OPEN and CLOSED set in each step in the iteration of A^* with the check for duplicated states removed. Each node is represented by the name of the state, the value of $f(N)$ and the value of $g(N)$. Some states are represented by several paths but only the cheapest path is of interest.

OPEN set	**CLOSED set**
(Start, 6, 0)	
(B, 10, 6) (A, 11, 2) (Goal1, 13, 13)	(Start, 6, 0)
(A, 11, 2) (Goal1, 12, 12) (Goal1, 13, 13) (C, 17, 7)	(Start, 6, 0) (B, 10, 6)
(B, 9, 5) (Start, 10, 4) (Goal2, 12, 12) (Goal1, 12, 12) (Goal1, 13, 13) (C, 17, 7)	(Start, 6, 0) (B, 10, 6) (A, 11, 2)
(Start, 10, 4) (Goal1, 11, 11) (Goal1, 12, 12) (Goal2, 12, 12) (Goal1, 13, 13) (C, 16, 6) (C, 17, 7)	(Start, 6, 0) (B, 10, 6) (A, 11, 2) (B, 9, 5)
(Goal1, 11, 11) (Goal1, 12, 12) (Goal2, 12, 12) (Goal1, 13, 13) (B, 14, 10) (A, 15, 6) (C, 16, 6) (C, 17, 7) (Goal1, 17, 17)	(Start, 6, 0) (B, 10, 6) (A, 11, 2) (B, 9, 5) (Start, 10, 4)

6.5 The fast and simple evaluation function, $f_1(N, A)$

$f_1(N, A)$ is mainly used to avoid repeatedly testing bad actions in all nodes. The function $f_1(N, A)$ is shown in equation 6.1, lower values mean better action.

$$f_1(S, A) = p_1 n_b - p_2 n_g + p_3 r \tag{6.1}$$

The function has three terms:

$p_1 n_b$ n_b is the number of times the action has been tested on other nodes in the search tree before reaching the current node and the resulting node had worse evaluation than the parent node. This statistic is collected for each individual action e.g. closing a specific switch. This is intended to avoid actions that have been **bad** in other nodes. The default value of p_1 is 1.

$p_2 n_g$ n_g is the number of times the action has been tested on other nodes in the search tree before reaching the current node and the resulting node had better evaluation than the parent node. This is intended to prioritize actions that have been **good** in other nodes, this is most relevant for actions that can be applied several times e.g. increasing or decreasing set points. The default value of p_2 is 0.

$p_3 r$ r is a random number between 0 and 1. This is intended to mix actions that otherwise should have had the same evaluation. Since the other terms are small integers it is likely that many actions otherwise would have received the same evaluation. The default value of p_3 is 0.01. Theoretically any value strictly between 0 and 1 would give exactly the same result.

If the action opens a switch and does not only disconnect a reactor or capacitor then an additional value p_4 is added to the evaluation function. The default value of p_4 is 10. The intention with this parameter is to give lower priority to actions that open switches since the number of switches that needs to be opened during a restoration is much lower than the number of other actions.

This evaluation function could be improved in many ways, for example the prioritization when connecting and disconnecting capacitors and reactors could depend on the voltage in the power system node in the parent state.

The use of statistics from use of actions in other states gives the algorithm a limited form of learning. Observe that the function is not recalculated for actions in a given node when n_g and n_b changes.

Since actions that have been available in many nodes generally collect a number of bad usages, newly available actions such as closing switches on newly energized busbars will be given higher priority.

6.6 The full evaluation function, $f_2(N)$

The full evaluation function is calculated after the topology and power flow of the new state is calculated. Therefore there is much more information available in order to give a better evaluation. This function is used to select the node to expand next. The function can be seen in eq. 6.2.

$$f_2(N) = g(N) + p_5 h(S) \tag{6.2}$$

S is the state corresponding to the node N. $f_2(N)$ has two terms:

$g(N)$ is the number of steps in the path from the initial state to the node N.

$p_5 h(S)$ is an estimate of the remaining cost to a goal (a state were all loads are supplied). p_5 is used to adjust how optimistic the prediction will be, the default value is 2.

The function $h(S)$ is shown in eq. 6.3.

$$\begin{aligned} h(S) = p_6 n_{open\ switches} - p_7 \frac{P_{load}}{P_{total\ load}} - p_8 \frac{P_{gen\ max} - P_{gen\ min}}{P_{total\ load}} + \\ + p_9 \frac{\sum_{i=1}^{n_{live\ nodes}} (f_i - f_{base})^2}{n_{live\ nodes}^{p_{11}}} + p_{10} \frac{\sum_{i=1}^{n_{live\ nodes}} (|V_i| - 1)^2}{n_{live\ nodes}^{p_{11}}} \end{aligned} \tag{6.3}$$

$h(S)$ has five terms:

$p_6 n_{open\ switches}$: This term gives priority to states where many switches are closed. $n_{open\ switches}$ is the number of open switches in the system. Since only normally closed switches and switches to shunt capacitors and reactors are modelled most switches will be closed in a goal state. The default value of p_6 is 1.

$-p_7 \frac{P_{load}}{P_{total\,load}}$: This term gives priority to states that already supply a large fraction of the load. P_{load} is the active power of the currently connected loads. $P_{total\,load}$ is the total amount of load in the system. The default value of p_7 is 500.

$-p_8 \frac{P_{gen\,max} - P_{gen\,min}}{P_{total\,load}}$: This term gives priority to states that have much regulating power. $P_{gen\,max}$ is the sum of the maximum active power of all the energized generating units. $P_{gen\,min}$ is the sum of the minimum active power of all the energized generating units. The default value of p_8 is 50.

$p_9 \frac{\sum_{i=1}^{n_{live\,nodes}} (f_i - f_{base})^2}{n_{live\,nodes}^{p_{11}}}$: This term gives priority to states where the frequency in the power system nodes are near the base frequency [7] This is the sum of the squared deviation in frequency normalized according to the total number of energized power system nodes. The normalization is done by dividing by $n_{live\,nodes}^{p_{11}}$, the default value of p_{11} is 0.5. This is a compromise between the sum of squared deviations ($p_{11} = 0$) and the mean of squared deviations ($p_{11} = 1$). All power system nodes in the same island will have the same frequency so the frequency of a large island will have greater impact than the frequency of a small island. The default value of p_9 is 5.

$p_{10} \frac{\sum_{i=1}^{n_{live\,nodes}} (|V_i| - 1)^2}{n_{live\,nodes}^{p_{11}}}$: This term is similar to the previous but gives priority to states with voltages near 1 p.u. instead of states with good frequency. The same normalization exponent p_{11} is used. The default value of p_{10} is 500. This value needs to be larger since voltages are calculated in p.u. while frequencies are calculated in Hz and therefore can have larger numerical deviations.

For states with violated limits in voltage, frequency or current a large value is added to $h(S)$ in order to guarantee that it has worse evaluation than any state without violated limits.

---

[7] $f_{base} = 50Hz$ in the test system.

6.7 Summary

The power system restoration problem is a path finding problem in the state space graph. A power system restoration algorithm based on the A^*-algorithm has been described. A^* has a number of good theoretical properties if the evaluation function fulfils some constraints.

No good evaluation function that fulfils these constraints has been found for the power system restoration problem so the algorithm can not be guaranteed to find a path if one exists or to find the shortest path. According to section 6.1 it is very likely that every evaluation function that fulfils these constraints will give exponential computation time in some cases and thus no useful guarantees could be made anyhow. If this is not true an efficient algorithm to solve all NP-complete problems would exist.

The algorithm will hopefully find a good path in most of the cases.

The proposed algorithm has two evaluation functions. $f_1(N, A)$ is a very simple and fast evaluation function to compare different actions in the same search tree node. It is based on statistics for the same action in other search tree nodes. $f_2(N)$ calculates a power flow for the state corresponding to search tree node N in order to compare it with other nodes, therefore it takes longer time to calculate.

The algorithm is randomized so if it fails to find a good solution a new attempt at running the algorithm can be made with a new result.

Chapter 7

Simulation results

The algorithm has been tested with the three cases presented for the test system. Each of these has been tested with six different initiations of the pseudo random generator. For each of the test cases one of these six test runs has been selected for closer examination[1].

7.1 Resulting sequence

The state of the system at different stages during the restoration from both the North and the South for the selected test run can be seen in the Figures 7.1, 7.2, 7.3, 7.4 and 7.5.

The Southern island grows much faster than the Northern island. The lines from CT71 to CT11 and CT12 are energized very late which prevents the Northern island from growing. Larger islands have larger number of available actions and therefore the algorithm will test more actions on large islands than on smaller. This will result in that large islands grow faster than small. In reality the longer starting times for thermal power plants would have slowed down the Southern island so the Northern island had grown more before synchronization. The reason that the synchronization happens so late is probably that the algorithm does no special handling of synchronization and it is hard to find net states in which synchronization is immedi-

---

[1] The random seed for the selected test run for start in the North and for start after a partial blackout are 0. This initiation gave atypical behaviour for start from both the North and the South at the same time so 1 was chosen instead. The initiation for the other 5 test runs was 10,11,12,13 and 14 for all three cases.

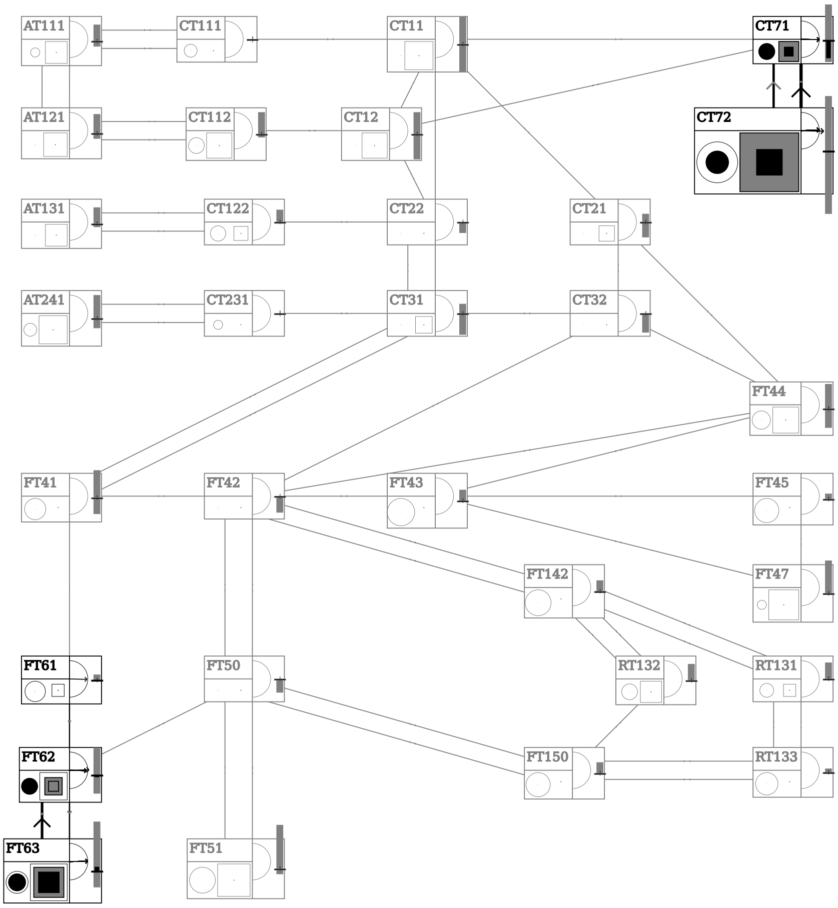


Figure 7.1: The state of the power system after 10 % of the actions in the restoration sequence when the restoration is started both from the North and the South. Five substations are energized and some load and generation has been connected.

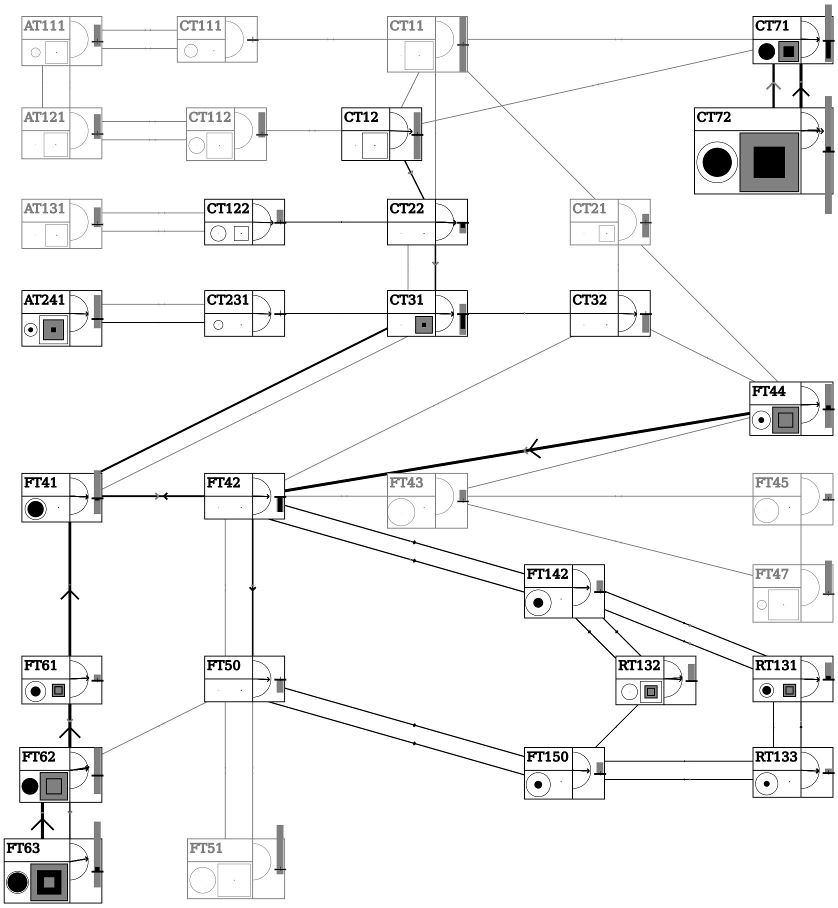


Figure 7.2: The state of the power system after 25 % of the actions in the restoration sequence when the restoration is started both from the North and the South. The Southern island has grown to 19 substations while the Northern island is just two substations. Large parts of the system are energized but very lightly loaded.

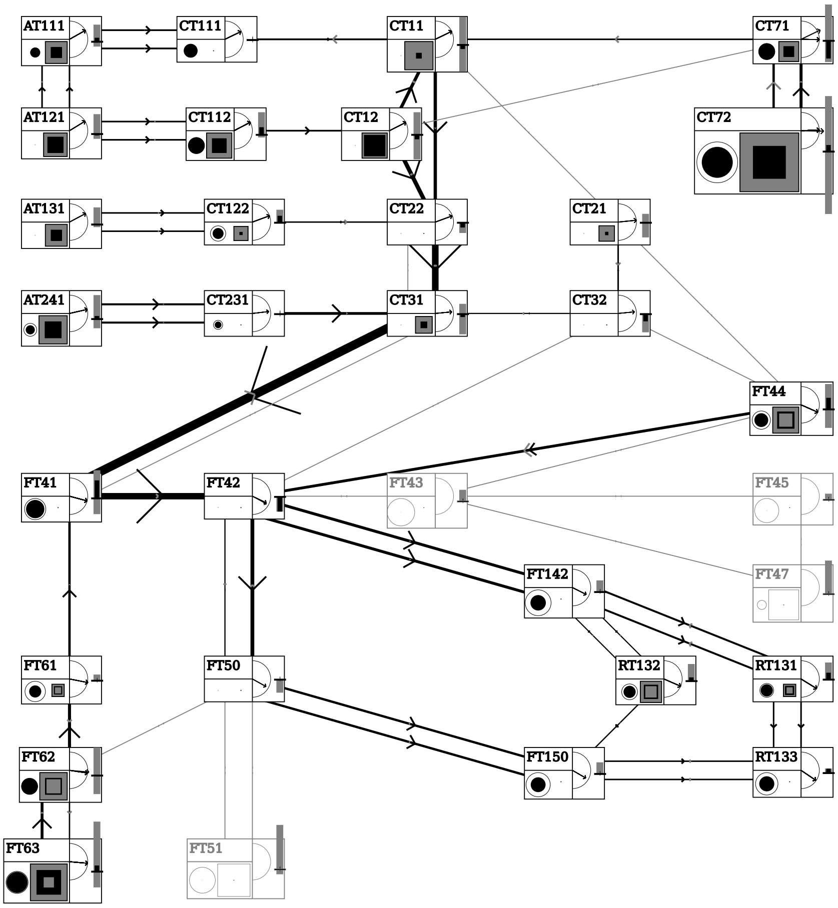


Figure 7.3: The state of the power system after 50 % of the actions in the restoration sequence when the restoration is started both from the North and the South. 28 substations are energized. The voltage angle differences in the system have increased. The islands have met in substation CT71 but are not yet synchronized.

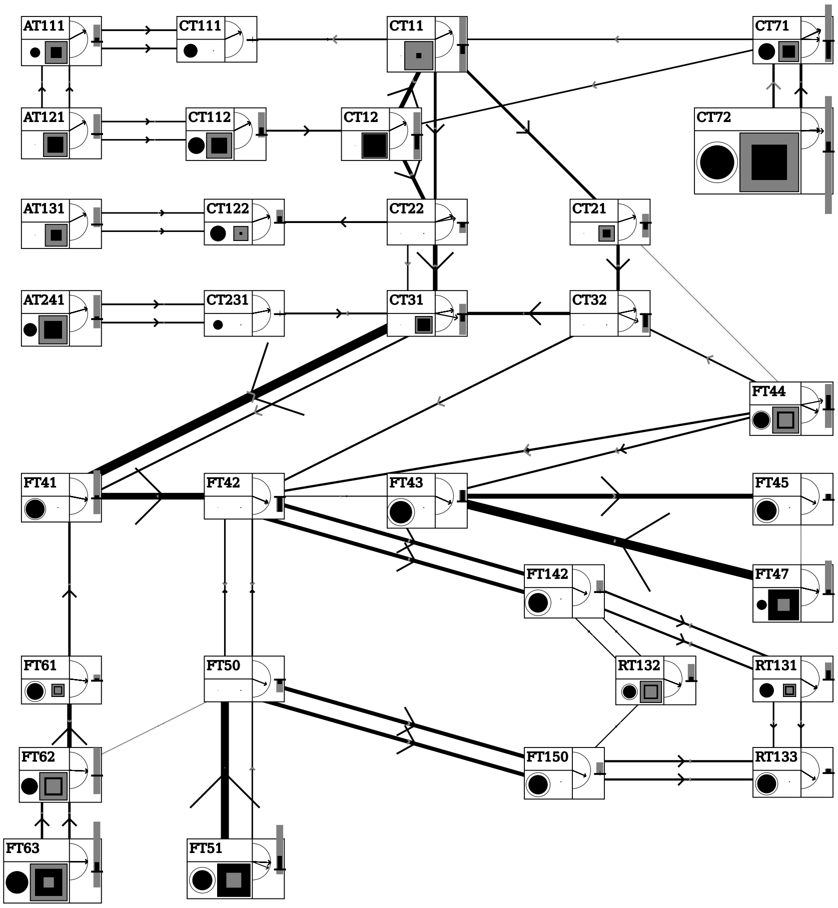


Figure 7.4: The state of the power system after 75 % of the actions in the restoration sequence when the restoration is started both from the North and the South. All substations and generating units are energized. Some power lines are not energized or only connected in one end. The islands are still separated in substation CT71.

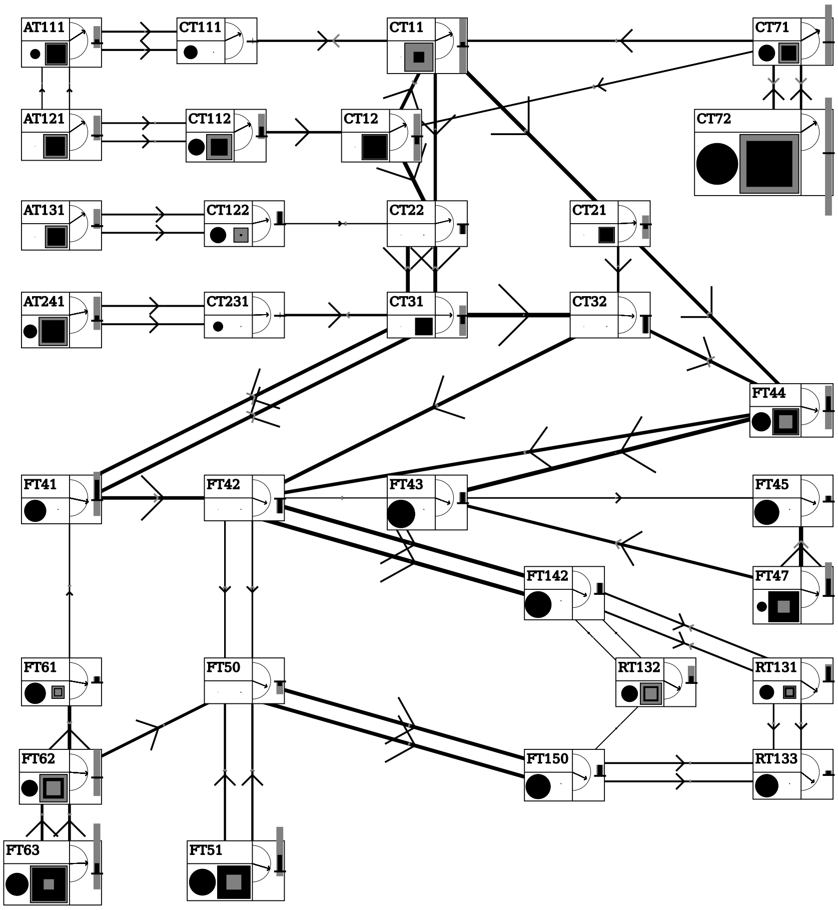


Figure 7.5: The state of the power system after the complete restoration sequence when the restoration is started both from the North and the South, compare with the normal state in Figure 4.3.

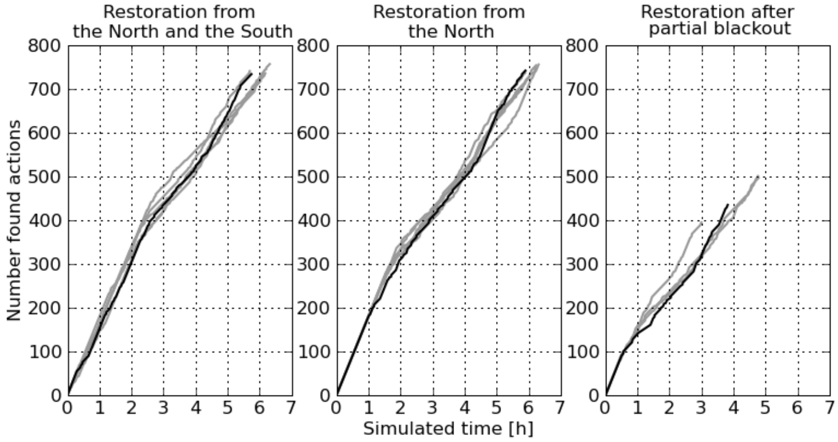


Figure 7.6: Number of actions taken as function of the simulated restoration time. The simulated restoration time are calculated according to section 3.4. The black lines represent the selected test run for each test case and the grey lines represent the other test runs.

ately advantageous.

The total number of actions taken during the restoration sequence can be seen in Figure 7.6. The graphs for restoration from the North and the South and from the North are very similar and the restoration from a partial blackout is similar to the later half of the other graphs. Two test runs for the partial blackout that finds very few actions and then get stuck (Not visible in Figure 7.6). The sequences starts with almost the maximum 180 actions per hour, during the latest 3 hour of the sequence the number of actions falls down to about 100 actions per hour due to more adjustments of active power set points is needed and takes longer time. This time estimate is of course very rough and does not take into account for example that the number of actions that can be taken in parallel increases with the size of the energized system.

The total available generation never limits the restoration in this case as can be seen in Figure 7.7. In a real restoration it is possible that the rate of load restoration would increase with the amount of synchronized generation since larger load steps can be connected and the active power can be ramped up faster. Since these aspects are not in the model it cannot be seen here.

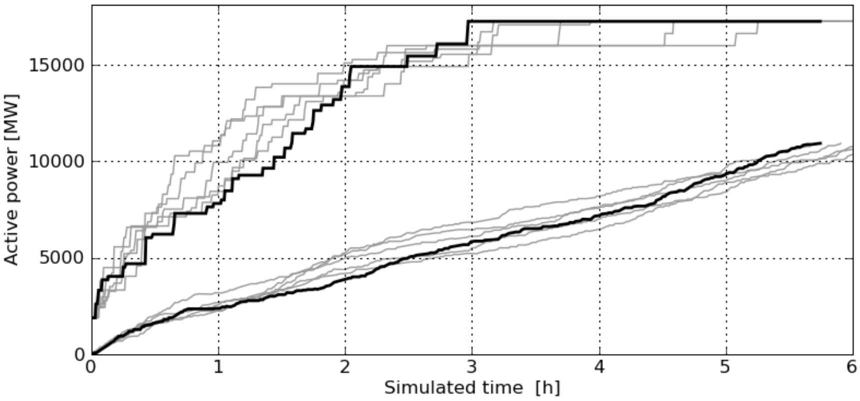


Figure 7.7: Upper line, connected active generating capacity. Lower line, connected active load. Both as function of the simulated time for start from both the North and the South at the same time. The black lines are for the selected test run and the grey are for the other test runs.

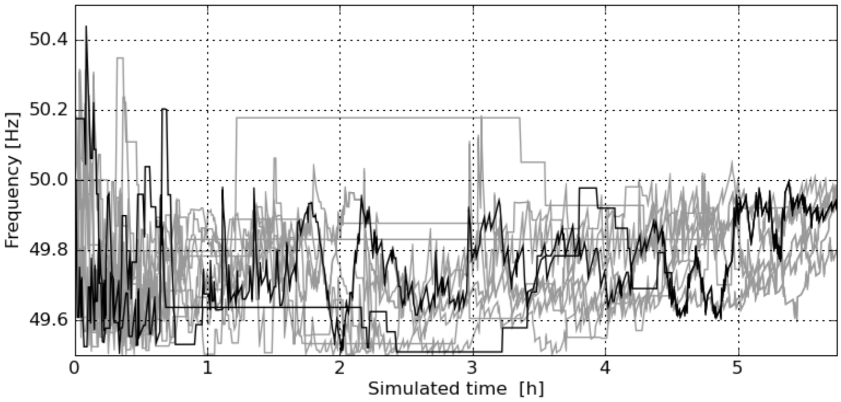


Figure 7.8: The frequency's in the system as function of the simulated time for start from both the North and the South. The black lines show the selected test run and the grey lines show the other five test runs.

The frequencies in the system during the restoration from both the North and the South can be seen in Figure 7.8. In the selected test run the synchronization of the islands can be seen after ca. 4.5 hours. Frequencies between 49.5 and 50.5 are allowed, the frequencies are mostly in the lower part of that band. One reason for this is that there are many more actions that lower the frequency than raise it so the frequency will tend to be adjusted down when too high faster than it is adjusted up when too low. One other reason is that the evaluation function has terms that prioritize states with much load connected and many switches closed.

The highest and lowest voltage in the system during the restoration from both the North and the South can be seen in Figure 7.9. It also shows the voltage after actions that have been tested but not used in the selected test run. Voltages between 0.90 and 1.10 p.u. are allowed. There are some tested actions that give too low voltages after about one hour, otherwise most voltage related difficulties seem to be with too high voltages, probably due to attempts to connect long lines.

The maximum voltage angle difference for the island with the largest angle difference can be seen in Figure 7.10. After about one hour there are

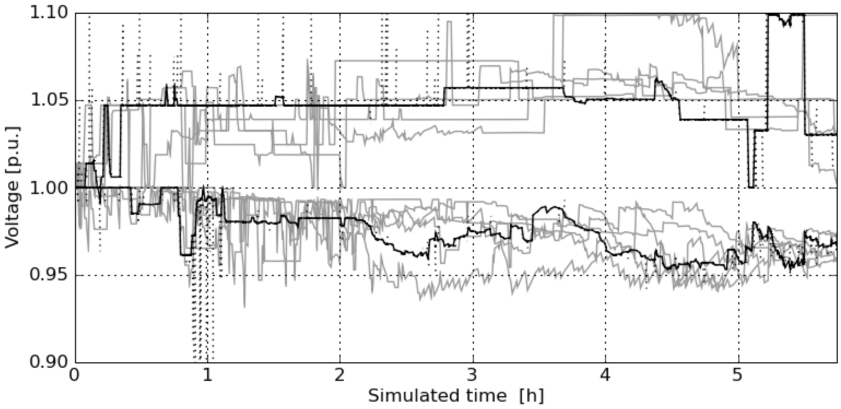


Figure 7.9: Highest and lowest voltage as function of the simulated time for start from both the North and the South. The black lines show the selected test run and the grey lines show the other five test runs. Doted lines represent actions tested in the selected test run but not used in the final sequence.

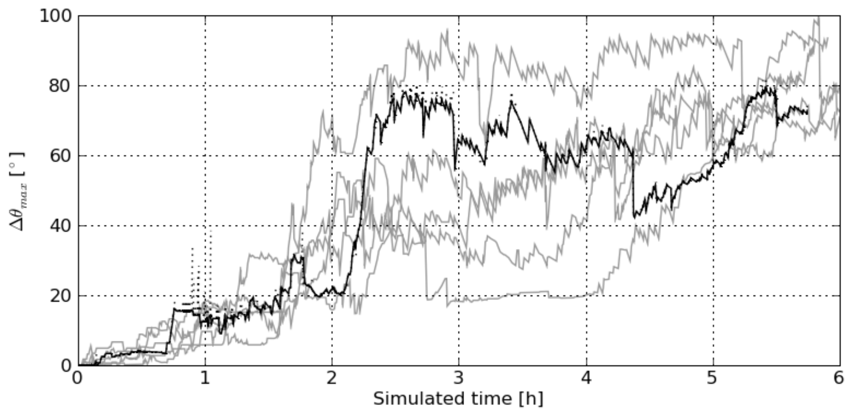


Figure 7.10: Maximum voltage angle difference in any of the islands as function of the simulated time for start from both the North and the South. The black lines show the selected test run and the grey lines show the other five test runs. Doted lines represent actions tested in the first test run but not used in the final sequence.

some actions that result in large increases in the maximum voltage angle difference, this indicates some stability problems and is probably related to the low voltages at the same time in Figure 7.9. The synchronization of the island cannot be seen since none of the nodes in the Northern island neither get the highest or the lowest angles. The large differences in maximum voltage angle difference between the different test runs are probable because the voltage angles are not directly used in the evaluation function so it only effects the evaluation indirectly. At the end of the restoration all the test runs converge between 65 and 95 degrees.

7.2 The search

The percentage of the test runs that succeeds in restoring the entire load and the average not restored load can be seen in table 7.1. The average restoration time for the different test cases can be seen in table 7.2. This is calculated as energy not delivered divided by the total load in the system. This

Table 7.1: Restoration results for test cases with base parameters. Average over 50 test runs.

Test case	Complete restorations [%]	Average load not restored [%]
Restoration from the North and the South	86	0.13
Restoration from the North	92	1.7
Restoration after partial blackout	32	22.0

Table 7.2: Average restoration times for test cases with base parameters. Average over 50 test runs.

Test case	During sequence [h]	Compensated for not restored load [h]
Restoration from the North and the South	2.83	2.85
Restoration from the North	2.91	3.07
Restoration after partial blackout	1.25	3.45

is similar to the reliable index ASIDI [5]. If not all load is restored in the found restoration sequence it is assumed that the remaining load will be reconnected manually 10 hours after the end of the sequence when calculating the energy not delivered. The best found complete restoration sequences for the three test cases with respect to average restoration time have an average restoration time of 2.53 hours for start from both the North and the South, 2.58 hours for start from the North only and 1.45 hours after a partial blackout.

In Figure 7.11 the number of found actions that will be part of the final sequence are plotted against the number of tested actions. For the total black-

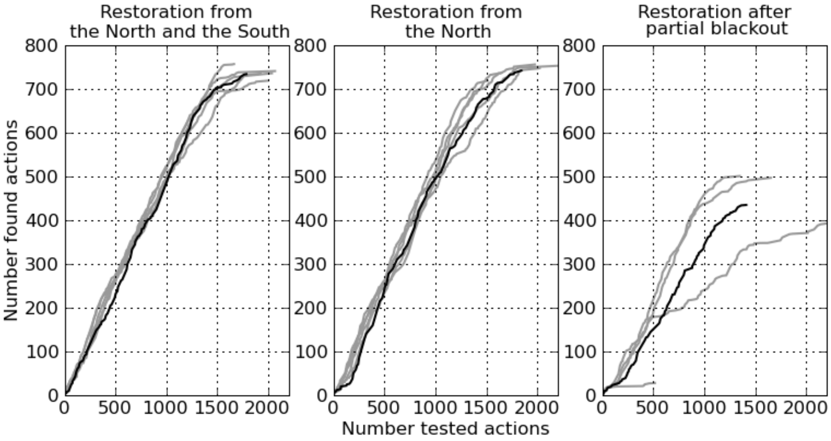


Figure 7.11: Length of sequence as function of number of investigated states. The black lines represent the selected test run for each test case and the grey lines represent the other tests.

outs about 50 % of the first 1200 actions tested are used in the final sequence, then the search becomes more difficult. The test runs for restoration from a partial blackout are much more diverse and only about 30 % of the tested actions is used in the final sequence. In Figure 7.12 a small part of a search tree can be seen in detail. The algorithm very seldom backs up and tests actions on other branches of the search tree but it has been observed in some test runs. The reason for this is that there are so many actions to test on each state.

The value of the evaluation function $f_2(N)$ for the search tree nodes along the found sequence (path) can be seen in Figure 7.13. This value should represent an estimate of the number of steps in the shortest sequence that restores all loads and that begins with the current action sequence. In this context it can seems strange to have negative values but it does not mater since these values are only compared to each other. As stated in the algorithm chapter this heuristic function is not admissible, this can be seen by the fact that the value of $f_2(N)$ is higher for states along the path than for the found goal state. The algorithm has very hard to find sequences that pass states with higher evaluation than earlier states due to the high number of available

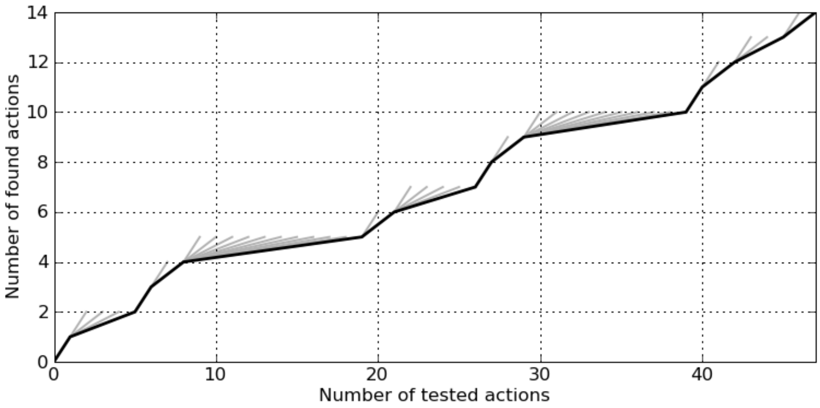


Figure 7.12: Length of sequence as function of number of investigated states for start in both the North and the South. Zoomed in at the beginning of the sequence. Grey lines represent actions that are not used in the final sequence.

actions that need to be tested.

7.3 Summary

The algorithm has been tested with the base parameter set presented in the algorithm chapter and found to work well. The algorithm has some difficulties with the partial blackout; this is probably because the algorithm has difficulties redistributing the power production. Another reason for this can be that the search parameters were mainly selected by trial and error on the test case with start from both the North and the South. The algorithm does very little backtracking. If no better state is found all the actions in the search tree node with best evaluation will be tested before actions is tested in other states. This takes long time so if the algorithm reaches a dead end the search will be very slow.

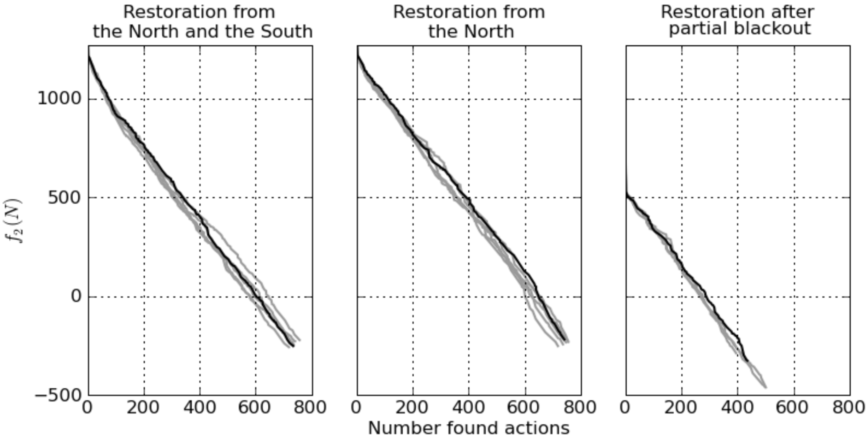


Figure 7.13: The value of the evaluation function $f_2(N)$ for the search tree nodes along the found sequence (path). The black lines represent the selected test run for each test case and the grey lines represent the other tests.

Chapter 8

Algorithm parameter sensitivity analysis

It is important that the need for tuning the search parameters is as low as possible and that one set of search parameters should work well in many different cases. Therefore a sensitivity analysis on the search parameters has been done.

One parameter at a time has been changed; the default values have been used for the other parameters. For each investigated value of each parameter 50 test runs have been performed. The average restoration time as defined in 7.2 has been used in order to compare how good different parameter values are.

8.1 The fast and simple evaluation function, $f_1(N, A)$

Equation 6.1:

$$f_1(S, A) = p_1 n_b - p_2 n_g + p_3 r$$

The most important parameter in $f_1(N, A)$ is p_1 that reduces the priority of actions that are tested in other states with bad result. If this is set to 0 then 14% and 25% of the load will not be restored for start from the North and the South and start from only the North respectively. This is probably because the search needs more than the maximum 3000 states due to a higher number of bad actions. As long as p_1 is greater than the randomization parameter p_3,

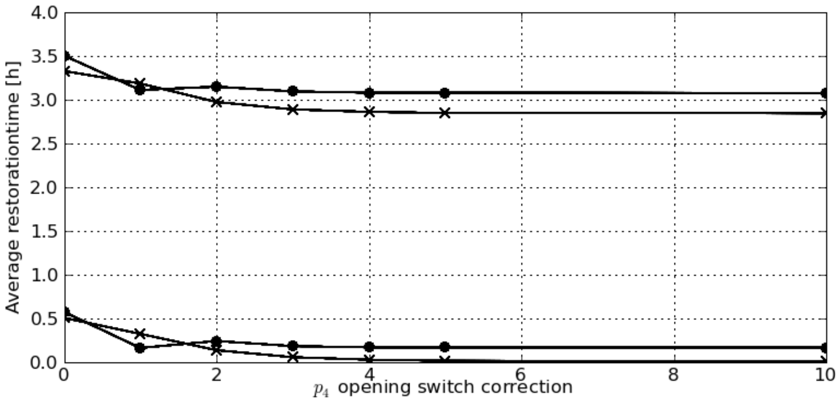


Figure 8.1: Average restoration time as a function of p_4 the correction added to the evaluation of actions that opens switches other than switches to shunt reactors and shunt capacitors. The default value is 10. The lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South.

the value of p_1 and p_3 do not to make any difference. The average restoration time becomes about 6 minutes longer if p_3 is 0.

p_2 is used to increase the priority of actions that have been tested in other states and given good results. p_2 makes very little difference, probably because most actions are only useful once in each sequence and the search tree is very dominated by one sequence.

The parameter p_4 reduces the priority of actions that open switches other than switches to shunt reactors and shunt capacitors. The impact of different values of the parameter p_4 can be seen in Figure 8.1.

All figures in this chapter will have the same lines but with different parameters on the horizontal axis.

The total restoration time including the assumed manual restoration of the remaining load in 10 h after the end of the automatic restoration sequence is shown. Below this line is a line that shows the part of the average restoration time that is due to the manual reconnection of the remaining load. The sensitivity to changes in the parameters has been tested both against the test case with start from the North and the test case with start from both the North and the South. As can be seen lower values than about 4 gives less successful restoration. Values above that seem to give no difference, probably could these actions be removed totally since they are not needed in most cases. The default value is 10.

The fraction of the available untested actions that the action to test is chosen from seems to make no difference as long as it is below 5% but it is important to allow for good randomization so the algorithm can be rerun on the same case until a good restoration sequence is found.

8.2 The full evaluation function, $f_2(N)$

Equation 6.2 and 6.3:

$$f_2(N) = g(N) + p_5 h(S)$$

$$h(S) = p_6 n_{open\ switches} - p_7 \frac{P_{load}}{P_{total\ load}} - p_8 \frac{P_{gen\ max} - P_{gen\ min}}{P_{total\ load}} + \\ + p_9 \frac{\sum_{i=1}^{n_{live\ nodes}} (f_i - f_{base})^2}{n_{live\ nodes}^{p_{11}}} + p_{10} \frac{\sum_{i=1}^{n_{live\ nodes}} (|V_i| - 1)^2}{n_{live\ nodes}^{p_{11}}}$$

The full evaluation function, $f_2(N)$ consists of two parts, the number of steps to the current state and the estimate of the remaining number of steps to a goal. The estimate can be scaled with p_5 to adjust how optimistic it is over all, this being equivalent with scaling all the parameters p_6 through p_{10}. The average restoration time for different scaling of the estimate can be seen in Figure 8.2. The test runs were done by scaling the real cost so far in the sequence, $g(N)$. From a conceptual perspective it makes more sense to scale the estimated cost since the real cost must be assumed to be known. Since the values of $f_2(N)$ are only compared with other values of the same function this makes no difference so the equivalent values for p_5 are shown on the horizontal axis. If the estimate is too optimistic about the remaining cost then most tested actions will look bad so a large number of actions are tested before one that improves the evaluation is found. If

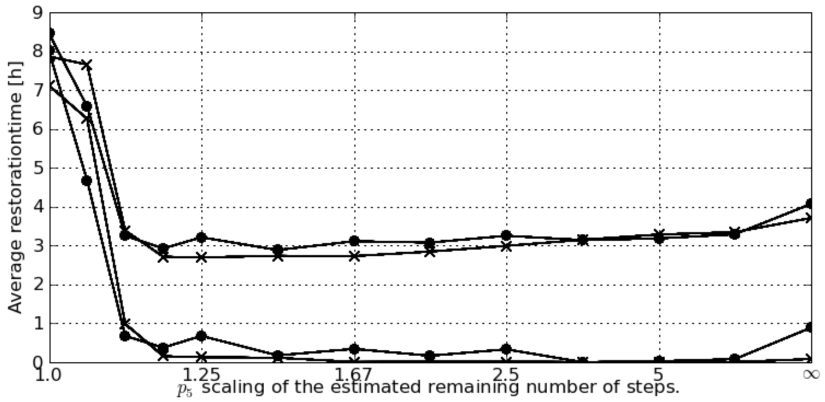


Figure 8.2: Average restoration time as a function of p_5, that scales the estimated remaining number of steps (cost). Observe the scale on the horizontal axis. The default value of p_5 is 2. The lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South.

the value of p_5 is lower than about 1.2 then the algorithm does not find a complete restoration sequence before the maximum number of states, 3000 is reached. If the estimate is too pessimistic more actions will look good so in reality the found actions will not go as straight towards the goal; this is seen as the faster restoration at 1.43 than at ∞. The default value of p_5 is 2.

The parameter p_6 is used to prioritize states with many closed switches, the effect of this parameter can be seen in Figure 8.3. If this is below 0.5 the action to close a switch with no other impact on the state such as energizing an unconnected bus bar will look bad to the algorithm. Since a number of such steps needs to be taken the algorithm will not find a complete restoration sequence before it reaches the maximum number of tested actions. The location of this transition is strongly dependent on the value of p_5, the transition in Figure 8.2 is to some extent a mirror image of this transition. A too high value of p_6 generates somewhat worse restoration sequences since there will be too much focus on closing switches.

The evaluation of connected load, p_7 is rather uncritical as can be seen

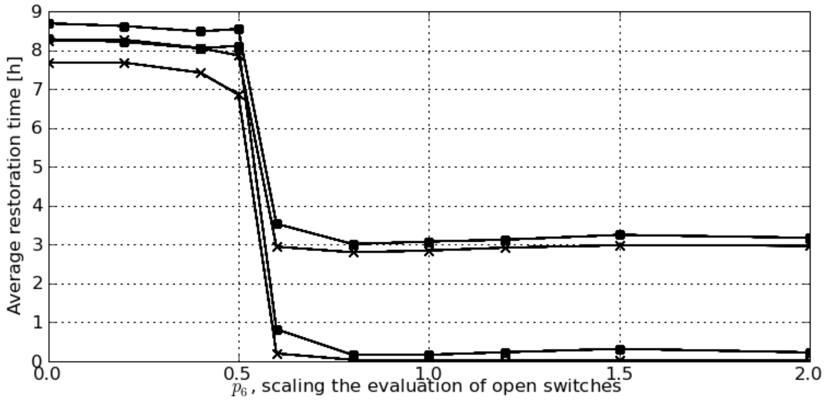


Figure 8.3: Average restoration time as a function of the parameter p_6 scaling the evaluation of open switches. The default value of p_6 is 1. The lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South.

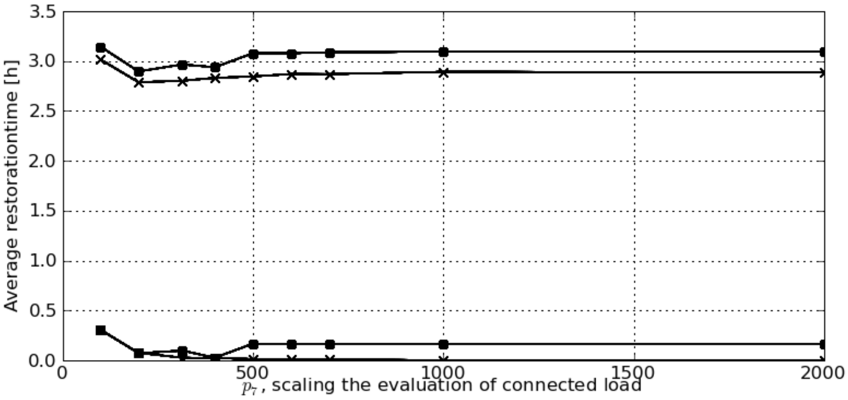


Figure 8.4: Average restoration time as a function of the parameter p_7 scaling the evaluation of connected load. The default value is 500. The lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South.

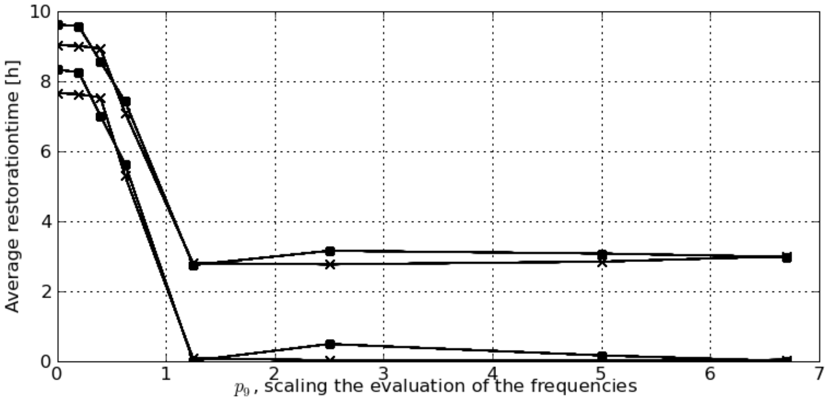


Figure 8.5: Average restoration time as a function of the parameter p_9 scaling the evaluation of frequency deviations. The default value is 5. The lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South.

in Figure 8.4. This is probably because p_6 has a similar function. A value between 200 and 500 seems to give the best results. The default value is 500.

The evaluation of regulating power range p_8 makes no difference and can be omitted, this is probably because the regulating range of the generating units is never an issue and it is almost always good to start all available generating units as soon as possible anyhow due to the voltage and frequency regulation.

As can be seen in Figure 8.5 the amount of automatically restored load decreases drastically if the prioritization of frequency deviation in the evaluation are below about 1, this is probably because then it does not seem advantageous to increase the active power set point of generating units even if the frequency is low. The default value is 5.

The impact of different prioritizing of the deviations in voltages can be seen in Figure 8.6. The default value is 500. The impact of lower values is surprisingly low.

The normalization exponent p_{11} was tested with the values 0.3, 0.4, 0.5, 0.6 and 0.7 The differences between the restoration times for the different

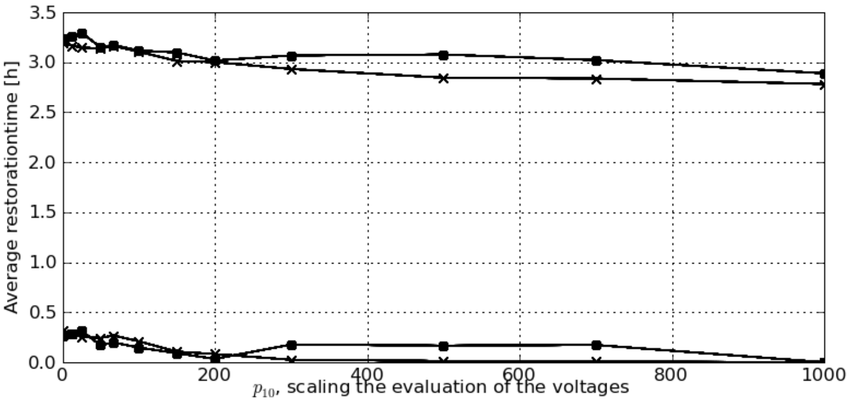


Figure 8.6: Average restoration time as a function of the parameter p_{10} scaling the evaluation of voltage deviations. The default value is 500. The lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South.

values are less than 20 minutes. The best result was fond for the values 0.5 and 0.6. The default value of p_{11} is 0.5.

8.3 Other parameters

The maximum number of tested actions must of course be large enough in order to find a complete restoration sequence, on the other hand a too large number will mean that when the search gets stuck it will use much computation time without doing significant progress. As can be seen in Figure 8.7 the restoration of the complete blackouts needs to test about 2 000 actions and more actions makes no further improvement while the restoration of a partial blackout continues to improve at least until 15 000 tested actions. The default is to test maximum 3000 actions.

If the maximum number of iterations in the power flow algorithm is less than 30 then at average more than 5% of the load will not be reconnected due to more states with non-converging power flows. A small improvement can be seen if the number of iterations is increased to 40. Larger values give no

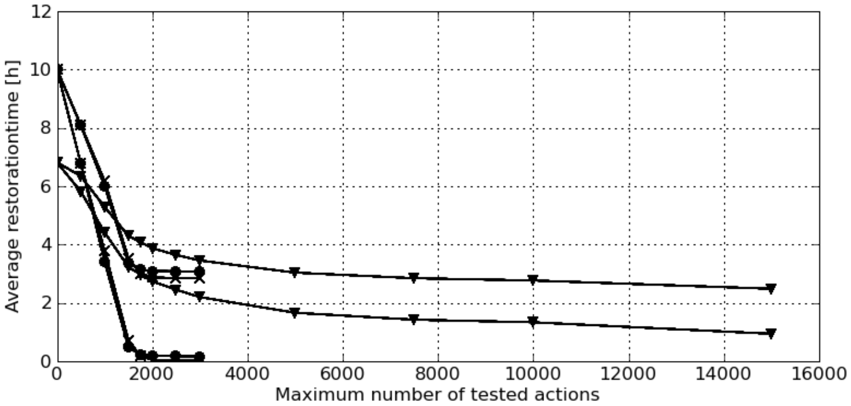


Figure 8.7: Average restoration time as a function of the maximum number of tested actions. The lines with triangles represent restoration from a partial blackout; the lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South. The default is to test maximum 3000 actions.

improvement but increases the computation time for non-converging states. The default value is 50.

In addition to the parameters above a number of additional parameters have been tested but found to be of little or no value. For example has it been tested to use $\sum_{i=1}^{n_{live\ nodes}} |f_i - f_{base}|$ in steed of $\sum_{i=1}^{n_{live\ nodes}} (f_i - f_{base})^2$.

8.4 Summary

Most parameters have a wide range where the restoration algorithm gives acceptable performance. There exists some critical transitions when the different parts of the evaluation function interact with each other, for example p_5 and p_6.

Chapter 9

Model parameter sensitivity analysis

Variations in the model parameters in table 4.2 have been tested in the same way that variations in the search parameters were tested in the previous chapter. This shows which conditions in the system that makes it hard to restore the system. The ability to compare the difficulties of restoration in different restoration situations was one of the main goals with automating the procedure. The original goal was to be able to compare the restoration difficulties in system with different amount of distributed generation of different types but more development is needed before the algorithm can be used to compare restoration with different production mixes in a useful way.

9.1 Limits

In order to make sure that all equipment works as intended the voltages, currents and frequencies need to be kept within certain limits. The narrower these limits are the cheaper equipment can be used and the probability that the equipment works as intended increases. On the other hand too narrow limits can slow down the restoration as more actions will be rejected and more set point adjustments need to be done. Figure 9.1 shows how the average restoration time depends on the allowed frequency deviation. If the allowed frequency deviation is below ca 0.2 Hz then the restoration algorithm gets difficulties. No improvement can be seen if the allowed frequency deviation is above 0.8 Hz.

The default allowed voltage range is 0.9 to 1.1 p.u. the effect of changing these limits can be seen in Figure 9.2. The voltage limits seem to have almost

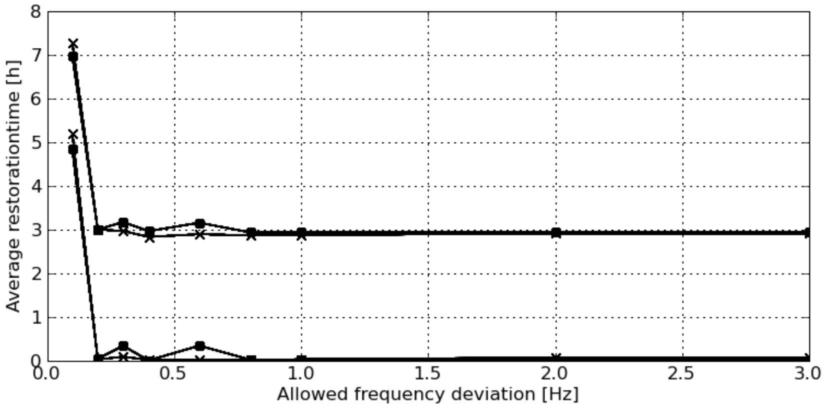


Figure 9.1: Average restoration time as a function of the maximum allowed deviation from nominal frequency. The default value is 0.5 Hz. The lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South.

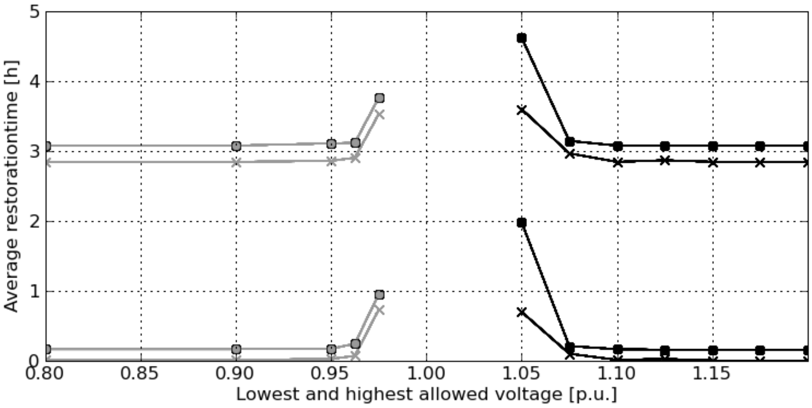


Figure 9.2: Average restoration time as a function of the highest and lowest allowed voltage. The default values are 0.9 and 1.1 p.u. Grey lines represent the lower voltage limit and black lines represent the higher voltage limit. The lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South. The voltage limits have been tested independent of each other with the other limit fixed to its default value.

no impact if they are outside 0.95 and 1.1 p.u. respectively. The restoration seems to need to be allowed more over voltage than under voltage, this is probably in order to be able to energize long lines. The algorithm will try to keep the voltage as close to 1 p.u. as possible and will in most cases not find a way to lower the voltage before connecting the line. These difficulties are bigger when energizing from the North only.

As can be seen in Figure 9.3 the maximum allowed currents in the transmission lines in this model are rather high compared to the currents in the normal power flow in Figure 4.3.

The effect of scaling the maximum allowed currents can be seen in Figure 9.4. The restorations are only marginally affected as long as the current limits are at least 50 % of their nominal values. By default the scale factor is set to 5 which are almost equivalent to turning off the current limits.

9.2 Random faults

Often during a restoration some components are out of service due to known faults and maintenances work. Unknown faulty components are an extra complication that is not handled here. In order to test the ability to restore the system in such situations, restorations with different number of random transmission lines removed have been tested[1]. The result can be seen in Figure 9.5. As expected restoration from the North are more sensitive to missing lines than restoration from both the North and the South. Restoration from the North is almost unaffected for up to 3 missing lines. Restoration from both the North and the South is almost unaffected up to 6 missing lines.

9.3 Scaling of the load

The effect of uniformly scale the active and reactive load can be seen in Figure 9.6. For scaling factors below 0.5 the restoration gives difficulties, probably due to high reactive generation in lightly loaded lines, this mostly affects restoration from the North only. For scaling factors between 0.5 and

---

[1] In order to get as consistent result as possible 50 random permutations of the 52 transmission lines where created. For each of the permutations and each value of n from 0 to 52 a restoration was attempted with the n first transmission lines in the sequence removed from the model.

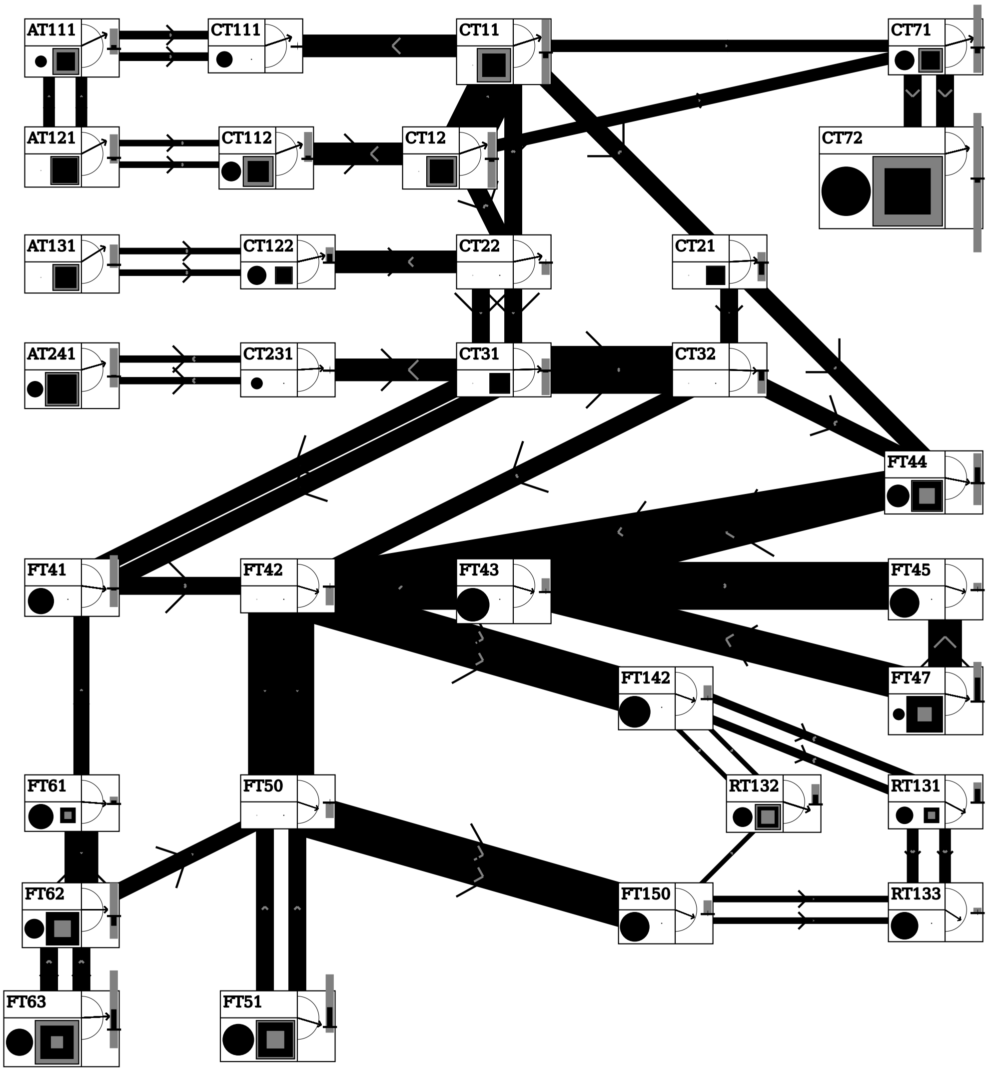


Figure 9.3: Schematic showing the current limits as the width of the transmission lines in the same scale as the earlier power flow visualizations.

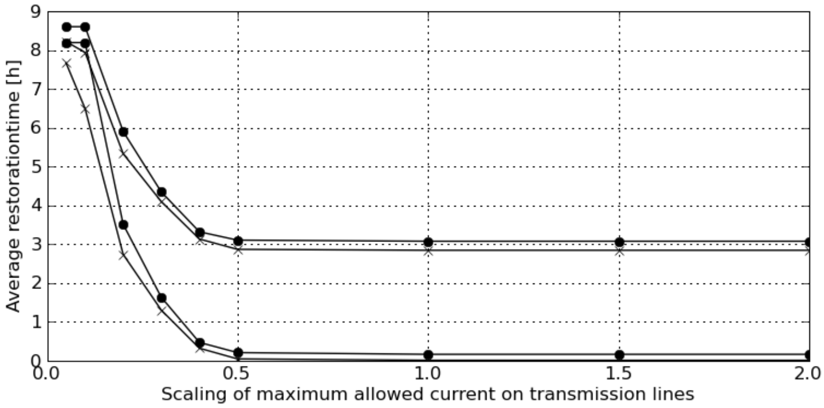


Figure 9.4: Average restoration time as a function of the scale factor for the highest allowed current on transmission lines. The default values are 5. The lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South.

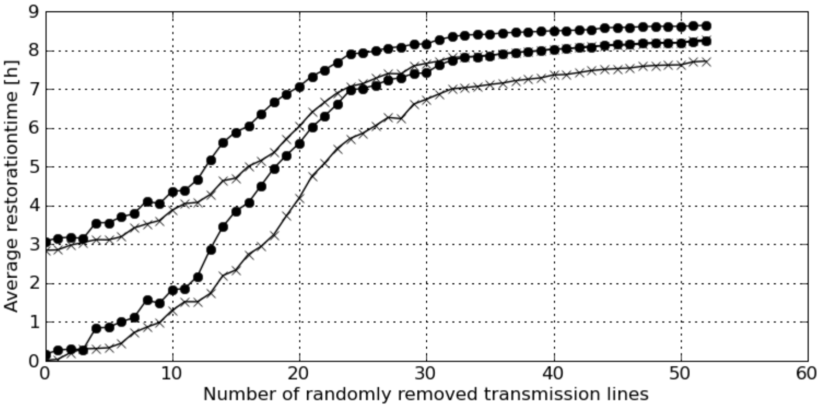


Figure 9.5: Average restoration time as a function of the number of removed transmission lines from the model. The lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South.

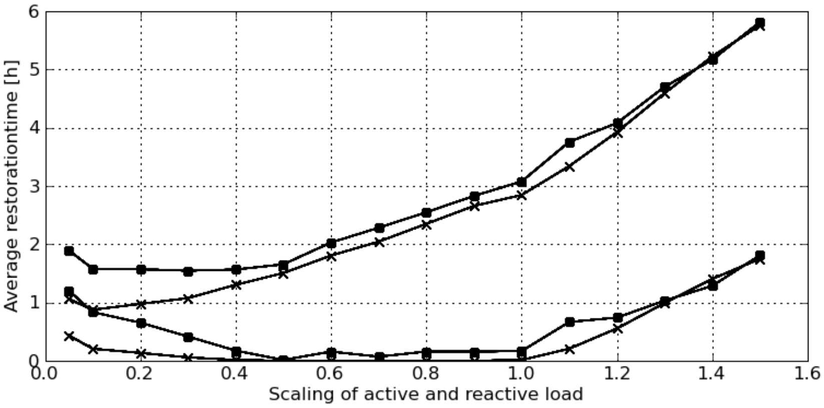


Figure 9.6: Average restoration time as a function of the scaling of active and reactive load. The lines with circles represent restoration from the North and the lines with x:s represents restoration from both the North and the South.

1.0 the average restoration time is almost proportional to the load. This can be explained by the proportionality between the number of load steps and the amount of load. For scaling factors above 1.0 it is difficult to energize the entire load even if the generating capacity is large enough due to difficulties transmitting the power to some loads.

9.4 Summary

The restoration algorithm has been tested in different power system restoration situations. Much of the features in the obtained result can be explained from a basic understanding of power system restoration.

Chapter 10

Conclusions

In this work, the search algorithm A^* has been adapted to the power system restoration problem.

10.1 Algorithm and test system

A power system restoration algorithm can be utilized in a number of different ways for example:

- To help operators in real time by suggesting relevant actions.
- To completely automate the power system restoration. Because of limited personnel on duty this is especially interesting in small power systems or independent restoration of many small islands in a larger power system.
- To make comparative off line studies of power system restoration in different situations such as:
    - Addition of new components such as power lines or generating units.
    - Restoration with some non operational components.
    - Variations in load patterns.

The proposed algorithm makes few direct assumptions about the restoration sequence which makes it suitable to compare different restoration situations and priorities during the restoration.

The proposed algorithm is relatively simple and can easily be extended with new rules and priorities.

The algorithm checks each state with a power flow calculation. The calculated power flow solution includes a frequency for each island, calculated according to the generating units frequency regulation in steady state and their limits in active power. The power system model contains no dynamic components so all changes are the direct result of an action.

The algorithm is randomized; the advantage of this is that the algorithm can be rerun several times with a new resulting restoration sequence each time until a satisfying sequence is found.

The algorithm has been tested on a modified variant of the NORDIC32 test system. NORDIC32 is standardised by CIGRE for studying long-term dynamics. It has 32 substations, 36 generating units and 545 switches. Most of the switches are due to the fact that all loads have been divided in to load steps of between 8 and 60 MW, each with its own switch. The total load in the system is 10 900 MW.

Three different initial states have been tested for the NORDIC32 test system:

- Total blackout with start in the North and the South of the test system.
- Total blackout with start in the North.
- Partial blackout somewhat similar to the blackout in year 1983.

The first two had high success rates, 92% and 86% of the test runs restores all loads. The start after a partial black out had a lower success rate, 32% of the test runs restores all load. One reason to why the partial blackout seems harder than the other cases is probably that the algorithm has difficulties to re-dispatch the production from the North to the newly energized Southern part.

Restoration with random power lines removed is tested with good results up to a point where about 20 % of the lines are removed.

The algorithm has been shown to be robust against variations in both search parameters and model parameters. This has been shown by changing one parameter at a time, run 50 test runs for each value of each parameter with different random number sequences and calculate the average restoration time for each value. In total more than 20 000 restorations have been tested with a total number of power flow calculations over 40 000 000.

The most important approximations made in the model are that it does not contain:

- Cold load pickup.
- Generating units starting times and ramping limits.
- Uncertainty and unpredicted events in the power system model.

When used in a real situation each action needs to be checked against a dynamic simulator. The proposed algorithm does not attempt to reach a state that fulfils the N-1 criterion after all loads are reconnected.

10.2 Computation time

This thesis also shows that any reasonable formulation of the power system restoration problem is NP-hard. Hence it is unlikely that an algorithm with guaranteed success in a reasonable time exists.

The algorithm implementation needs less than 3000 power flows and 30 seconds on a 2.2 GHz AMD Opteron$^{\text{TM}}$in order to find a sequence of about 750 actions that restores the power after an almost complete blackout in NORDIC32.

No direct tests are done on how the computation time increases with the complexity of the power system. Since the problem is NP-hard the computation time increases exponentially with the complexity of the power system in the worst case. The current results suggest that the algorithm scales almost linearly in terms of number of power flow calculations except when it reaches a state where the evaluation function does not give enough guidance, then the algorithm gets exponential running time.

Chapter 11

Future work

The presented algorithm shows promising results but the modelling needs to be complemented in order to reflect more important aspects of the power system restorations. The most obvious features to add are uncertainty in how the power system behaves and time dependent and dynamic models for load and generation.

11.1 Time dependent and dynamic models

The present algorithm only handles static systems, i.e. systems in which every change is due to an action. The algorithm needs to be upgraded to handle dynamic phenomena such as generating unit start up times, limited rate of change of the active power of generating units, reconnection of thermostatically-controlled loads, and so on.

The evaluation of the result of an action in such a model should not be based only at the state of the system immediately after the action but should take in consideration the future development of the power system state. This is easiest done if the evaluation assumes that every evaluated action is the last in the sequence.

A dynamic electromechanical simulation of the result of each action probably takes too much computation time. Instead repeated power flow calculations, for example at the times 1, 2, 4, 8, 16, 32, 64, 128, 256 and 512 minutes from each action is taken can be used.

Other forms of time dependence that need to be considered are the limited rate at which actions can be executed and the delay from decision to execution. This is especially important for actions that are executed manually.

11.2 Other model improvements

There are a lot of details in the modelling of the power system components that could be added such as:

- Tap changers.
- Protective relays.
- Switches with different ability to make and break current, this is the difference between circuit breakers and disconnectors.
- Distribution nets that shall not be run in a meshed configuration due to protection issues.
- More detailed power plant models.

11.3 Algorithmic improvements

The introduction of uncertainty and time dependent models increases the computation time needed. This combined with the general goal of being able to handle as many power system restoration situations as possible motivates improvements in the algorithm.

The algorithm can be improved at several levels:

- Improvements in the general algorithm that are not directly related to the details of power system restoration. For example:
    - Change the order in which actions are tested so that not all actions in the state with the best evaluation needs to be tested before actions in other states are tested.
    - Divide the evaluation of states into more steps where faster calculations are used to determine which states to evaluate further. The last step when it is almost decided that the action will be used could be a full dynamic simulation of the action.

- Improve the evaluation functions for example by adding terms for new variables such as voltage angle differences, short circuit power and so on.

- Speed improvements such as using incremental power flow calculation were appropriate and improving details in the implementation of the algorithm.

11.4 Distributed and hierarchical restoration

In order to plan restoration in large power systems or power systems modelled in great detail with reasonable computation times it is necessary to divide the model into loosely coupled sub systems.

This can be done with systems at the same level such as transmission systems with few tie lines between each other.

It can also be done in a hierarchical manner that separates the planning of the transmission system restoration from the restoration planning for each of the subtransmission systems at lower levels. The planning of the transmission system restoration then sets the limits for the exchange of power for each of the subtransmission systems. In this manner the complete power flow for the entire system need not to be recalculated each time a small action such as connecting a small load at low voltage level is considered.

This is similar to the restoration planning in the planned cell concept of Energinet.dk [28]. The cell concept is an attempt to redefine the distribution of responsibility between local grid companies and the transmission grid company responsible for the system.

11.5 Uncertainty

Restoration under uncertainty can be simulated by using two power system models. One to represent the real system, actions taken on this model can not be undone just as in reality. The second one is used to represent the algorithm knowledge about the real system, the algorithm can try different actions in a search tree on this model as described earlier. The algorithm searches for an action sequence in the second model and then applies one action at a time on the first model. The algorithm reads values from the first model corresponding to the measurements done in a real system. These values are compared against what is expected from the second model. If the difference is larger than some threshold then the second model is updated by some estimation algorithm so that it predicts values more similar to the one from the first model and a new action sequence is computed from the current state.

By introducing errors in the second model and in the measurements the effect of such errors on the restoration can be studied

When doing a real restoration online the first model is of course replaced by the real system.

This extension of the algorithm will increase the computation time significantly by requiring a recalculation of the action sequence each time the second model needs to be updated. By not computing a complete restoration sequence each time the second model is updated this increase in restoration time can be limited.

References

[1] Elavbrottet den 27 december 1983 – rapport från Vattenfalls störningskommission. Technical report, Vattenfall, 1984. In Swedish.

[2] Tekniska riktlinjer för kontrollanläggningar, automatiksystem, driftuppbyggnadsautomatik duba. kats 6.1 rev. 96-11-15 b. Technical report, Svenska Kraftnät, 1996. In Swedish.

[3] Tekniska riktlinjer för kontrollanläggningar, automatiksystem, nollspänningsautomatik (uo-aut.). kats 6.4 rev. 96-11-15 b. Technical report, Svenska Kraftnät, 1996. In Swedish.

[4] Elavbrottet 23 september 2003 – händelser och åtgärder. Technical report, Svenska Kraftnät, 2003. In Swedish.

[5] Ieee guide for electric power distribution reliabilty indices. *IEEE Std 1366-2003 (Revision of IEEE Std 1366-1998)*, 2004.

[6] Tekniska riktlinjer för kontrollanläggningar, hjälpkraftsystem, likströmssystem. tr2-09-1 rev. 06-02-02 d. Technical report, Svenska Kraftnät, 2006. In Swedish.

[7] CIGRE TF 38.02.08. Long term dynamics. phase ii. final report. Technical report, CE/SC 38 GT/WG 02, Ref. No. 102, 1995.

[8] M.M. Adibi. *Power System Restoration, methodologies & implementation strategies*. IEEE Press, 2000.

[9] Evert Agneholm. *Cold Load Pick-up*. PhD thesis, Chalmers university of Technology, 1999.

[10] Giorgio Ausiello, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, and V. Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.

[11] S. Barsali, R. Salvati, and R. Zaottini. Use of hvdc links for power system restoration. *Electric Power Systems Research*, 79(6):973–983, 2009.

[12] Liu C.-C., Liou K.-L., Chu R.F., and Holen A.T. Generation capability dispatch for bulk power system restoration: a knowledge-based approach. *IEEE Transactions on Power Systems*, 8(1):316–325, 1993.

[13] Teo C.Y. and Wei Shen. Development of an interactive rule-based system for bulk power system restoration. *IEEE Transactions on Power Systems*, 15(2):646–653, 2000.

[14] A. Edström and K. Walve. The training simulator aristo - design and experiences. In *IEEE Power Engineering Society Winter Meeting*, New York, NY, USA, 1999.

[15] J. Figueiredo, J. Sauve, P. Nicolletti, E. Rocha, S. Araujo, F. Amorim, A. Feitosa, R. Agra, and W. Ribeiro. Smart action: A tool to help power system restoration. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pages 167–167, 2008.

[16] L. H. Flink and K. Carlsen. Operating under stress and strain. In *IEEE Spectrum 15*, 1978.

[17] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

[18] Xiangzhen He, Zhiwei Liao, Wenxin Guo, Fushuan Wen, Junhui Liang, and Jian Fu. A multi-objective model for transmission network restoration based on multi-agent and tabu search. In *2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, pages 2392–2397, Nanjing, China, 2008.

[19] Keisuke Iwasaki and Hidenori Aoki. Service restoration problem in distribution power system using improved ga. *Electrical Engineering in Japan*, 166(1):10–19, 2009.

[20] Roger Kearsley. Restoration in sweden and experience gained from the blackout of 1983. *IEEE Transactions on Power Systems*, 2(2):422–428, May 1987.

[21] S. Khushalani, J.M. Solanki, and N.N. Schulz. Optimized restoration of unbalanced distribution systems. *IEEE Transactions on Power Systems*, 22(2):624–630, 2007.

[22] T. Kostic, R. Cherkaoui, A. Germond, and P. Pruvot. Decision aid function for restoration of transmission powersystems: conceptual design and real time considerations. *IEEE Transactions on Power Systems*, 13(3):923–929, 1998. Lab. de Reseaux Electr., Ecole Polytech. Federale de Lausanne.

[23] P. Kundur. *Power system stability and control*. McGraw-Hill, Inc, 1993. ISBN 0-07-035958-X.

[24] Dong Liu, Yunping Chen, Guang Shen, and Youping Fan. A multi-agent based approach for modeling and simulation of bulk power system restoration. *2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific*, 2005.

[25] Dong Liu, Yunping Chen, Rongxiang Yuan, and Guang Shen. Backtracking based algorithm in hierarchical time-extended petri net model for power system restoration. In *2005 IPEC The 7th International Power Engineering Conference*, Singapore, 2005.

[26] Qiang Liu, Libao Shi, Ming Zhou, Gengyin Li, and Yixin Ni. A new solution to generators start-up sequence during power system restoration. In *2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, pages 2845–2849, Nanjing, China, 2008.

[27] Yan Liu and Xueping Gu. Reconfiguration of network skeleton based on discrete particle-swarm optimization for black-start restoration. In

2006 IEEE Power Engineering Society General Meeting, Montreal, Quebec, Canada, 2006.

[28] Per Lund. The danish cell project - part 1: Background and general approach. In *2007 IEEE Power Engineering Society General Meeting*, Tampa, Florida, USA, 2007.

[29] K. Matsumoto, T. Sakaguchi, R.J. Kafka, and M.M. Adibi. Knowledge-based systems as operational aids in power systemrestoration. *Proceedings of the IEEE*, 80(5):689–697, 1992.

[30] A.A. Mota, L. Toledo, M. Mota, and A. Morelato. Restoration building blocks identification using a heuristic search approach. In *2006 IEEE Power Engineering Society General Meeting*, Montreal, Quebec, Canada, 2006.

[31] T. Nagata, Y. Tahara, and H. Fujita. An agent approach to bulk power system restoration. *Power Engineering Society General Meeting, 2005. IEEE*, 3:599–604 Vol. 1, 2005.

[32] T.J. Overbye and J.D. Weber. Visualization of power system data. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000.*, volume vol.1, 2000.

[33] Thi Thu Ha Pham, Y. Besanger, and N. Hadjsaid. New challenges in power system restoration with large scale of dispersed generation insertion. *IEEE Transactions on Power Systems*, 24(1):398–406, 2009.

[34] Gabriel Robins and Alexander Zelikovsky. Tighter bounds for graph steiner tree approximation. *SIAM Journal on Discrete Mathematics*, 19:122–134, 2005.

[35] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, 2003.

[36] K. Salek, U. Spanel, and G. Krost. Flexible support for operators in restoring bulk power systems. *2003 CIGRE/IEEE PES International Symposium - Quality and Security of Electric Power Delivery Systems*, pages 187–192, 2003.

[37] S. Srivastava and K. L. Butler-Burry. Expert-system method for automatic reconfiguration for restoration of shipboard power systems. *"IEE Proceedings – Generation, Transmission & Distribution"*, 153(3):253–261, 2006.

[38] System Restoration Subcommittee. An approach to standing phase angle reduction. *IEEE Transactions on Power Systems*, 9(1):470–478, 1994.

[39] K. Walve, A. Edstrom, and L. Karlstrom. Advanced dynamic simulations as an important aid to restoration studies and training. In *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, Pittsburgh, PA, USA, 2008.

[40] Kojima Y., Warashina S., Nakamura S., and Matsumoto K. Development of a guidance method for power system restoration. *IEEE Transactions on Power Systems*, 4(3):1219–1227, 1989.

Appendix A

Implementation

The first prototype of the system was developed in Matlab[1] and used PSAT[2] for power flow calculations. This was to slow and limited, it needed ca 30 minutes to find a restoration sequence.

Therefore the system was rewritten in C++[3]. It worked well but lacked a proper user interface and it was inconvenient to write simple scripting tasks in C++.

This was solved by writing a Python[4] program that encapsulates the computation core that remains in C++.

The connection between the parts in C++ and Python is created by SWIG. SWIG is an acronym for Simplified Wrapper Interface Generator and is a program for generating interface code between code written in C/C++ and a dynamic language such as Python[5]. An overview of the software implementation can be seen in Figure A.1

---

[1] See http://www.mathworks.com, 20 May 2009, the website of Mathworks, the company that sells Matlab.

[2] See F. Milano An Open Source Power System Analysis Toolbox, IEEE Trans. on Power Systems, Vol. 20, No. 3, August 2005, pp. 1199-1206. and http://www.power.uwaterloo.ca/ fmilano/psat.htm, 20 May 2009. The PSAT website of PSAT's creator Dr. Federico Milano.

[3] GCC is used to compile the C++ code see http://gcc.gnu.org/, 20 May 2009, the GNU Compiler Collection (GCC) website.

[4] Python is a high-level, general purpose, interactive and object oriented language with a large standard library. More info on: http://www.python.org/, 20 May 2009

[5] See http://www.swig.org/, 20 May 2009, The SWIG website.

The Python code contains the following parts:

- The overall search algorithm.
- Functions to load and save power system models including the state of the system.
- The web interface and the scripting interface.
- Functions for generating plots and other figures.

The C++ code contains these parts:

- The topology calculation that deduces the islands and nodes to calculate power flow for from the system topology and the status of the switches.
- The power flow solver. This is a Newton Raphson solver that uses the library Sparse 1.4[6] to solve the sparse linear equation systems of each iteration, see section 3.1.
- The evaluation functions for both states and actions.
- The function for choosing an action to apply on a given state. This uses the library RandomLib in order to randomize the algorithm. [7]

The power system model file, the model parameter file and the search parameter file are python programs that are run by the main program in order to set up the power system model and the parameters. Appendix B contains the power system model file for the variant of NORDIC32 that is used in this thesis.

The program can be used in two modes, by the web interface and by the scripting interface. The web interface is used for testing restorations interactively. A screenshot of the web interface can be seen in Figure A.2.

The scripting interface is used by the example script that generates the data for the figures in chapter 7.

The batch system also uses the scripting interface.

---

[6] See http://sparse.sourceforge.net, 20 May 2009. The website of the Sparse linear algebra library.

[7] See http://charles.karney.info/random/, 20 May 2009. The website for RandomLib

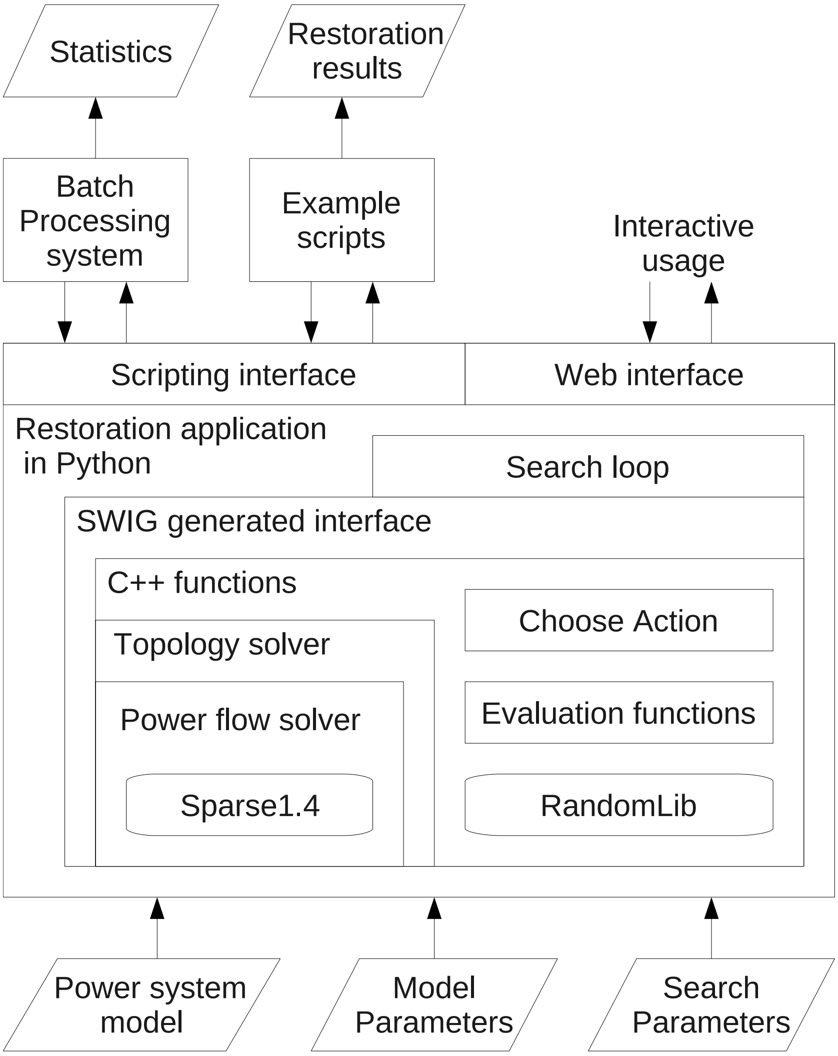
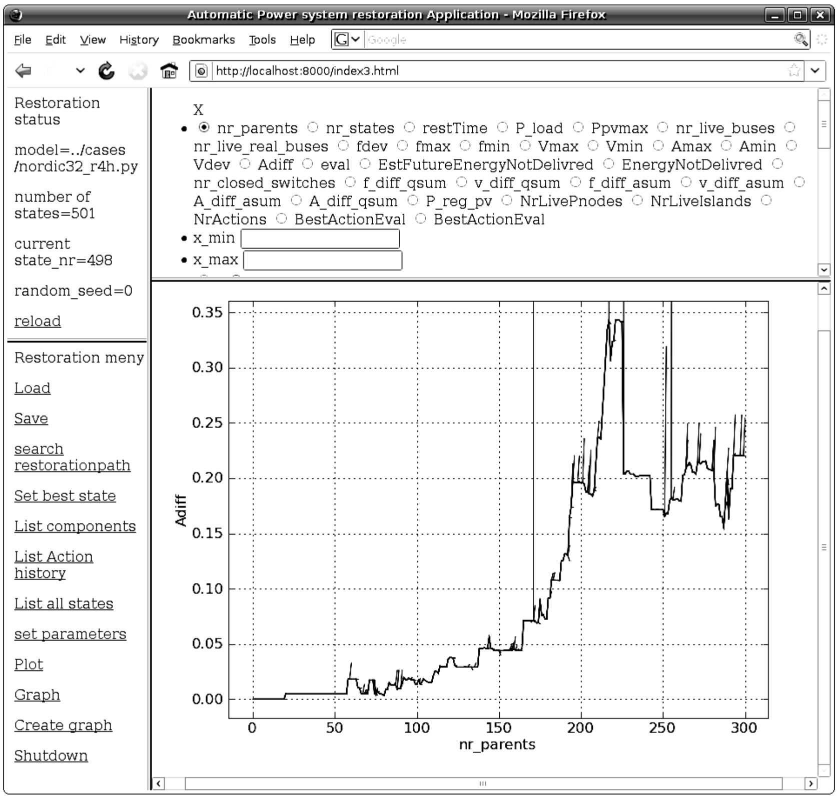


Figure A.1: Overview of the software implementation.



Restoration status

model=../cases/nordic32_r4h.py

number of states=501

current state_nr=498

random_seed=0

reload

X

- ◉ nr_parents ○ nr_states ○ restTime ○ P_load ○ Ppvmax ○ nr_live_buses ○ nr_live_real_buses ○ fdev ○ fmax ○ fmin ○ Vmax ○ Vmin ○ Amax ○ Amin ○ Vdev ○ Adiff ○ eval ○ EstFutureEnergyNotDelivred ○ EnergyNotDelivred ○ nr_closed_switches ○ f_diff_qsum ○ v_diff_qsum ○ f_diff_asum ○ v_diff_asum ○ A_diff_asum ○ A_diff_qsum ○ P_reg_pv ○ NrLivePnodes ○ NrLiveIslands ○ NrActions ○ BestActionEval ○ BestActionEval
- x_min
- x_max

Figure A.2: A screenshot of the web interface. The plot shows the maximum voltage angle difference during a restoration sequence.

The batch system is a set of Python scripts that are used to run large numbers of restoration simulations with variations in the parameters on a computer cluster. It is used to generate the results in chapter 8 and 9.

Appendix B

Net data

```
#Import some modules
import restore # The API for the C++ −power flow and restoration code
from math import ∗ #Elementary math functions
from model_helper import ∗ #Helper functions used to build the net model
import copy
r = restor #Just a shorter name

#Define defaults for net components.
#This makes the declaration of each component shorter and easier to understand.
#It is also necessary in order to be able to easy change global properties of the model.
#These defaults can be overridden by defining rw. defaults before loading the model.
#These defaults can also be overridden for each component by
#modifying the values before sending them to the component declaration . See FT63−G11
 defaults =Base_struct ()
 defaults . Initial_state ="NS" #other states are "N", "half" and "full "
 defaults . f_base=50.0 # Hz
 defaults . S_base=100.0 # MVA
 defaults . Vmax=1.1      #Maximum allowed voltage on any node
 defaults . Vmin=0.9      #Minimum allowed voltage on any node
 defaults . fmax=50.5        #Maximum allowed frequency on any node
 defaults . fmin=49.5        #Minimum allowed frequency on any node
 defaults . Load_factor_P=1          #provides a method of scaling all active load at once.
```

```
defaults.Load_factor_Q=1                #provides a method of scaling all
        reactive loads at once.
defaults.Load_Step_Size_factor=1        #provides a method of scaling the
        number of load steps in all loads.
defaults.Load_on_factor=0               #fraction of all load switches
        that are closed.
defaults.Generation_P_factor=1          #provides a method of scaling all
        active generation at once
defaults.Generation_Q_factor=1          #provides a method of scaling all
        reactive generation at once
defaults.Generation_Pinit=0             #The initial active power set
        point of the generating units as fraction of Pmax-Pmin, Pinit=0 ->
        pset=Pmin, Pinit=1 -> Pset=Pmax
defaults.Generation_On=False            #The default is that the
        generation is off in the initial state.
defaults.Generation_Vset=1.000          #Default voltage set point in per
        unit for generating units.
defaults.Generation_dQ_per_dV=10000.0 #Multiplied by (Qmax-Qmin),
        delta V in p.u.
defaults.Generation_dP_per_df_hydro=-0.1512 #Multiplied by (Pmax-Pmin),
        delta f in Hz
defaults.Generation_dP_per_df_thermal=0.0

#Has the defaults been overridden?
if rw.__dict__.has_key("defaults"):
        d=copy.deepcopy(rw.defaults) #Just a shorter name
        defaults=d
else:
        d=defaults      #Just a shorter name
rw.defaults_used=d #For use when printing key figures etcetera.

#Create an empty power net.
n=r.Net(d.f_base, d.S_base)
#Add components
#Switches to connect the objects to the busbars are automatically
        generated by the helper functions.
#The nodes between the switches and the objects are also automatically
        created.
#All data are given in per unit.

#Declare nodes that should be considered busbars and the nominal voltages
```

```
for these.
Busbars(n,d,Vn=400,
Names=["CT11B","CT71","CT11A","CT21","CT22B","CT12A","CT12B","CT72A","CT72B","CT22A","CT31","CT32A","FT44B","CT32B","FT41A","FT41B","FT42","FT44A","FT43B","FT43A","FT45","FT47A","FT61","FT50A","FT50B","FT47B","FT62","FT63A","FT63B","FT51"])

Busbars(n,d,Vn=220, Names=["CT231","AT241"])

Busbars(n,d,Vn=130,Names=["AT111","CT111","AT121","CT112","AT131","CT122","RT132B","FT142A","RT132A","FT142B","RT131B","RT131A","RT133","FT150A","FT150B"])

#Declare the substation names and one node in each.
#The program finds the rest of the nodes by following all
#switches.
 Substations (n, d, [
 ("AT111_swy","AT111" ),
 ("CT111_swy","CT111" ),
 ("AT121_swy","AT121" ),
 ("CT112_swy","CT112" ),
 ("AT131_swy","AT131" ),
 ("CT122_swy","CT122" ),
 ("CT231_swy","CT231" ),
 ("AT241_swy","AT241" ),
 ("CT71_swy" ,"CT71" ),
 ("CT11_swy" ,"CT11A" ),
 ("CT21_swy" ,"CT21" ),
 ("CT12_swy" ,"CT12A" ),
 ("CT72_swy" ,"CT72A" ),
 ("CT22_swy" ,"CT22A" ),
 ("CT31_swy" ,"CT31" ),
 ("CT32_swy" ,"CT32A" ),
 ("FT41_swy" ,"FT41A" ),
 ("FT42_swy" ,"FT42"  ),
 ("FT44_swy" ,"FT44A" ),
 ("FT43_swy" ,"FT43A" ),
 ("FT45_swy" ,"FT45"  ),
 ("FT47_swy" ,"FT47A" ),
 ("FT61_swy" ,"FT61"  ),
 ("FT50_swy" ,"FT50A" ),
```

```
("FT62_swy" ,"FT62" ),
("FT63_swy" ,"FT63A" ),
("FT51_swy" ,"FT51" ),
("FT142_swy","FT142A"),
("RT132_swy","RT132A"),
("RT131_swy","RT131A"),
("RT133_swy","RT133" ),
("FT150_swy","FT150A")])

#Loads(net, defaults , load_point ,Busbar,P , Q , step_size_p );
#Each row generates as many loads as necessary so
#that P of each load will not be bigger than step_size_p .
#P and Q are distributed even between the loads generated on one row.
Loads(n, d, "AT111−T" , "AT111" , 1.0  , 0.4 , 0.2 );
Loads(n, d, "CT111−T" , "CT111" , 2.0  , 0.8 , 0.2 );
Loads(n, d, "CT112−T" , "CT112" , 3.0  , 1.0 , 0.2 );
Loads(n, d, "CT122−T" , "CT122" , 2.8  , 0.95, 0.4 );
Loads(n, d, "FT142−T1", "FT142A", 4.0 , 1.5 , 0.4 );
Loads(n, d, "FT142−T2", "FT142B", 4.0 , 1.5 , 0.3 );
Loads(n, d, "FT150−T1", "FT150A", 1.5 , 0.5 , 0.5 );
Loads(n, d, "FT150−T2", "FT150B", 5.5 , 2.0 , 0.3 );
Loads(n, d, "RT131−T1", "RT131A", 1.725, 0.03, 0.2 );
Loads(n, d, "RT131−T2", "RT131B", 0.575, 0.01, 0.2 );
Loads(n, d, "RT132−T1", "RT132A", 1.5 , 0.4 , 0.3 );
Loads(n, d, "RT132−T2", "RT132B", 1.5 , 0.4 , 0.3 );
Loads(n, d, "RT133−T" , "RT133" , 6.0  , 0.2 , 0.3 );
Loads(n, d, "CT231−T" , "CT231" , 1.0  , 0.3 , 0.2 );
Loads(n, d, "AT241−T" , "AT241" , 2.0  , 0.5 , 0.2 );
Loads(n, d, "CT71−T" , "CT71" , 3.0  , 1.0 , 0.6 );
Loads(n, d, "CT72−T1" , "CT72A" , 10.0 , 2.5 , 0.4 );
Loads(n, d, "CT72−T2" , "CT72B" , 10.0 , 2.5 , 0.4 );
Loads(n, d, "FT41A−T1", "FT41A" , 2.85 , 0.9 , 0.5 );
Loads(n, d, "FT41−T2" , "FT41B" , 2.55 , 0.7 , 0.3 );
Loads(n, d, "FT43−T1" , "FT43A" , 4.75 , 1.11, 0.3 );
Loads(n, d, "FT43−T2" , "FT43B" , 4.25 , 1.92, 0.3 );
Loads(n, d, "FT44−T1" , "FT44A" , 0.75 , 0.4 , 0.3 );
Loads(n, d, "FT44−T2" , "FT44B" , 3.25 , 1.1 , 0.3 );
Loads(n, d, "FT45−T" , "FT45"  , 7.0  , 2.5 , 0.4 );
Loads(n, d, "FT47−T1" , "FT47A" , 0.75 , 0.375,0.3 );
Loads(n, d, "FT47−T2" , "FT47B" , 0.25 , 0.125,0.1 );
Loads(n, d, "FT51−T" , "FT51"  , 8.0  , 3.0 , 0.4 );
```

```
Loads(n, d, "FT61−T" , "FT61"  , 5.0   , 1.5  , 0.5  );
Loads(n, d, "FT62−T" , "FT62"  , 3.0   , 0.85, 0.5  );
Loads(n, d, "FT63−T1" , "FT63A" , 2.92 , 1.5  , 0.5  );
Loads(n, d, "FT63−T2" , "FT63B" , 2.92 , 1.5  , 0.5  );

#Generation_unit(net, defaults , Type, Name, Busbar, Pmax, Pmin, Qmax, Qmin)

 Generation_unit(n, d, "hydro"   , "AT111−G" , "AT111" , 5.7 ,  0.0,  3.0 ,−1.0 )
 Generation_unit(n, d, "hydro"   , "AT121−G" , "AT121" , 6.65,  0.0,  3.5 ,−1.0 )
 Generation_unit(n, d, "hydro"   , "AT131−G" , "AT131" , 5.7 ,  0.0,  3.0 ,−0.6 )
 Generation_unit(n, d, "hydro"   , "CT112−G1", "CT112" , 2.85, 0.0,  1.5 ,−0.2 )
 Generation_unit(n, d, "hydro"   , "CT112−G2", "CT112" , 4.75, 0.0,  2.5 ,−0.4 )
 Generation_unit(n, d, "hydro"   , "CT122−G" , "CT122" , 2.38,  0.0, 1.75,−0.25)
 Generation_unit(n, d, "hydro"   , "RT131−G" , "RT131A" , 1.8 ,  0.6,  2.0 , 0.0 )
 Generation_unit(n, d, "thermal", "RT132−G1", "RT132A", 3.6 , 1.2,  2.0 , 0.0 )
 Generation_unit(n, d, "thermal", "RT132−G2", "RT132B", 1.8 , 0.6,  1.0 , 0.0 )
 Generation_unit(n, d, "hydro"   , "AT241−G1", "AT241" , 4.75, 0.0,  2.5 ,−1.0 )
 Generation_unit(n, d, "hydro"   , "AT241−G2", "AT241" , 4.75, 0.0, 1.75,−0.8 )
 Generation_unit(n, d, "hydro"   , "CT11−G1" , "CT11A" , 3.8 ,  0.0,  2.0 ,−0.5 )
 Generation_unit(n, d, "hydro"   , "CT11−G2" , "CT11B" , 5.7 ,  0.0,  3.0 ,−0.5 )
 Generation_unit(n, d, "hydro"   , "CT12−G1" , "CT12A" , 4.75, 0.0,  2.5 ,−0.3 )
 Generation_unit(n, d, "hydro"   , "CT12−G2" , "CT12B" , 2.85, 0.0,  1.5 ,−0.3 )
 Generation_unit(n, d, "hydro"   , "CT21−G" , "CT21"  ,  2.85,  0.0,  1.5 ,−0.3 )
 Generation_unit(n, d, "hydro"   , "CT31−G" , "CT31"  ,  3.33,  0.0,
```

```
1.75,−0.4 )
Generation_unit(n, d, "hydro"  , "CT71−G" , "CT71"  , 4.75, 0.0, 2.5 ,−1.0 )
Generation_unit(n, d, "hydro"  , "CT72−G1" , "CT72A" , 13.5, 0.0, 3.0 ,−3.0 )
Generation_unit(n, d, "hydro"  , "CT72−G2" , "CT72B" , 13.5, 0.0, 3.0 ,−3.0 )
Generation_unit(n, d, "hydro"  , "CT72−G3" , "CT72A" , 7.15, 0.0, 2.0 ,−1.5 )
Generation_unit(n, d, "hydro"  , "CT72−G4" , "CT72B" , 7.15, 0.0, 2.0 ,−1.5 )
Generation_unit(n, d, "thermal", "FT41−G" , "FT41A" , 0.0 , 0.0, 3.0 ,−2.0 )
Generation_unit(n, d, "thermal", "FT44−G1" , "FT44B" , 6.3 , 1.8, 3.5 , 0.0 )
Generation_unit(n, d, "thermal", "FT44−G2" , "FT44A" , 1.8 , 0.6, 1.0 , 0.0 )
Generation_unit(n, d, "thermal", "FT47−G1" , "FT47A" , 5.4 , 1.0, 3.0 , 0.0 )
Generation_unit(n, d, "thermal", "FT47−G2" , "FT47B" , 5.4 , 1.0, 3.0 , 0.0 )
Generation_unit(n, d, "thermal", "FT51−G1" , "FT51"  , 6.3 , 1.5, 3.5 , 0.0 )
Generation_unit(n, d, "thermal", "FT51−G2" , "FT51"  , 6.3 , 1.5, 3.5 , 0.0 )
Generation_unit(n, d, "thermal", "FT61−G" , "FT61"  , 1.8 , 0.6, 1.0 , 0.0 )
Generation_unit(n, d, "thermal", "FT62−G1" , "FT62"  , 5.4 , 1.5, 3.0 , 0.0 )
Generation_unit(n, d, "thermal", "FT62−G2" , "FT62"  , 1.8 , 0.6, 1.0 , 0.0 )
Generation_unit(n, d, "thermal", "FT62−G3" , "FT62"  , 1.8 , 0.6, 1.0 , 0.0 )
Generation_unit(n, d, "thermal", "FT63−G1" , "FT63A" , 5.4 , 1.5, 3.0 , 0.0 )
Generation_unit(n, d, "thermal", "FT63−G2" , "FT63A" , 5.4 , 1.5, 3.0 , 0.0 )
#Define a black start unit in the South
#represents an aggregation of Gas turbines and hydro power in Scania and Zealand
#that has black start capability , not part of the original Nordic 32
```

```
        model.
#Give some modified defaults.
d_black_start = copy.deepcopy(d)
d_black_start.Generation_Pset =1.5/(5.4+1.5) #To get 50 Hz at 0 MW
d_black_start.Generation_dP_per_df_thermal=-4.0/7.0
Generation_unit(n, d_black_start, "thermal", "FT63-G11", "FT63A",
        5.4,-1.5, 3.0,-3.0 )

#Shunt_reactance(net, defaults, Name, bus, Qnom)
Shunt_reactance(n, d, "CT122-EK", "CT122", 0.5)
Shunt_reactance(n, d, "FT142-EK", "FT142A", 2.0)
Shunt_reactance(n, d, "FT150-EK", "FT150A", 2.0)
Shunt_reactance(n, d, "RT131-EK", "RT131B", 1.0)
Shunt_reactance(n, d, "RT133-EK", "RT133", 0.7)
Shunt_reactance(n, d, "CT11-X1", "CT11A",-1.0)
Shunt_reactance(n, d, "CT11-X2", "CT11B",-1.0)
Shunt_reactance(n, d, "CT11-X3", "CT11A",-1.0)
Shunt_reactance(n, d, "CT11-X4", "CT11B",-1.0)
Shunt_reactance(n, d, "CT12-X1", "CT12A",-1.0)
Shunt_reactance(n, d, "CT12-X2", "CT12B",-1.0)
Shunt_reactance(n, d, "CT12-X3", "CT12A",-1.0)
Shunt_reactance(n, d, "CT12-X4", "CT12B",-1.0)
Shunt_reactance(n, d, "CT21-X1", "CT21",-1.5)
Shunt_reactance(n, d, "CT21-X2", "CT21",-1.0)
Shunt_reactance(n, d, "CT22-X1", "CT22A",-1.0)
Shunt_reactance(n, d, "CT22-X2", "CT22B",-1.0)
Shunt_reactance(n, d, "CT31-X1", "CT31",-1.5)
Shunt_reactance(n, d, "CT31-X2", "CT31",-1.0)
Shunt_reactance(n, d, "CT31-X3", "CT31",-1.0)
Shunt_reactance(n, d, "CT32-X1", "CT32A",-1.5)
Shunt_reactance(n, d, "CT32-X2", "CT32B",-1.0)
Shunt_reactance(n, d, "CT32-X3", "CT32A",-1.0)
Shunt_reactance(n, d, "CT71-EK", "CT71", 4.0)
Shunt_reactance(n, d, "CT71-X1", "CT71",-1.5)
Shunt_reactance(n, d, "CT71-X2", "CT71",-1.5)
Shunt_reactance(n, d, "CT72-X1", "CT72A",-1.5)
Shunt_reactance(n, d, "CT72-X2", "CT72A",-1.0)
Shunt_reactance(n, d, "FT41-EK", "FT41B", 2.0)
Shunt_reactance(n, d, "FT41-X", "FT41B",-1.0)
Shunt_reactance(n, d, "FT42-X1", "FT42",-1.0)
Shunt_reactance(n, d, "FT42-X2", "FT42",-1.0)
```

```
Shunt_reactance(n, d, "FT42-X3" , "FT42" ,-1.0)
Shunt_reactance(n, d, "FT43-EK" , "FT43A" , 2.0)
Shunt_reactance(n, d, "FT44-X1" , "FT44A" ,-1.5)
Shunt_reactance(n, d, "FT44-X2" , "FT44B" ,-1.0)
Shunt_reactance(n, d, "FT44-X3" , "FT44A" ,-1.0)
Shunt_reactance(n, d, "FT45-EK" , "FT45" ,  1.0)
Shunt_reactance(n, d, "FT50-X1" , "FT50A" ,-1.5)
Shunt_reactance(n, d, "FT50-X2" , "FT50B" ,-1.0)
Shunt_reactance(n, d, "FT51-EK" , "FT51" ,  1.0)
Shunt_reactance(n, d, "FT51-X" , "FT51"  ,-1.0)
Shunt_reactance(n, d, "FT62-X1" , "FT62" ,-1.5)
Shunt_reactance(n, d, "FT62-X2" , "FT62" ,-1.0)
Shunt_reactance(n, d, "FT62-X3" , "FT62" ,-1.0)

Ibase130=d.S_base /130.0/(3.0**0.5) #base current in kA for 130, 220 and 400 kV
Ibase220=d.S_base /220.0/(3.0**0.5)
Ibase400=d.S_base /400.0/(3.0**0.5)

#Line(n, d, Name, busbar1, busbar2, R, X, B, Imax)
Line(n, d, "AL1", "AT111", "CT111", 0.010, 0.070, 0.014, 3.12/Ibase130 )
Line(n, d, "AL2", "AT111", "CT111", 0.010, 0.070, 0.014, 3.12/Ibase130 )
Line(n, d, "AL3", "AT111", "AT121", 0.007, 0.050, 0.010, 4.37/Ibase130 )
Line(n, d, "AL4", "AT111", "AT121", 0.007, 0.050, 0.010, 4.37/Ibase130 )
Line(n, d, "AL5", "CT112", "AT121", 0.014, 0.090, 0.018, 2.25/Ibase130 )
Line(n, d, "AL6", "CT112", "AT121", 0.014, 0.090, 0.018, 2.25/Ibase130 )
Line(n, d, "AL7", "AT131", "CT122", 0.030, 0.200, 0.030, 2.25/Ibase130 )
Line(n, d, "AL8", "AT131", "CT122", 0.030, 0.200, 0.030, 2.25/Ibase130 )
Line(n, d, "AL9", "CT231", "AT241", 0.012, 0.090, 0.015, 1.50/Ibase220 )
Line(n, d, "AL10","CT231", "AT241", 0.012, 0.090, 0.015, 1.50/Ibase220 )
Line(n, d, "CL1", "CT11B", "CT71" , 0.005, 0.045, 1.400, 1.50/Ibase400 )
Line(n, d, "CL2", "CT11A", "CT21" , 0.006, 0.060, 1.800, 2.25/Ibase400 )
Line(n, d, "CL3", "CT11B", "CT22B", 0.004, 0.040, 1.200, 2.25/Ibase400 )
Line(n, d, "CL4", "CT11A", "CT12A", 0.001, 0.008, 0.200, 6.25/Ibase400 )
Line(n, d, "CL5", "CT12B", "CT71" , 0.005, 0.050, 1.500, 1.50/Ibase400 )
Line(n, d, "CL6", "CT71" , "CT72A", 0.003, 0.030, 3.000, 2.25/Ibase400 )
Line(n, d, "CL7", "CT71" , "CT72B", 0.003, 0.030, 3.000, 2.25/Ibase400 )
Line(n, d, "CL8", "CT12A", "CT22A", 0.004, 0.035, 1.050, 2.25/Ibase400 )
Line(n, d, "CL9", "CT22B", "CT31" , 0.004, 0.040, 1.200, 2.25/Ibase400 )
Line(n, d, "CL10","CT22A", "CT31" , 0.004, 0.040, 1.200, 2.25/Ibase400 )
Line(n, d, "CL11","CT21" , "CT32A", 0.004, 0.040, 1.200, 2.25/Ibase400 )
```

```
Line(n, d, "CL12","CT21" , "FT44B", 0.010, 0.060, 3.000, 2.25/Ibase400 )
Line(n, d, "CL13","CT31" , "CT32B", 0.001, 0.010, 0.300, 6.25/Ibase400 )
Line(n, d, "CL14","CT31" , "FT41A", 0.006, 0.040, 2.400, 2.25/Ibase400 )
Line(n, d, "CL15","CT31" , "FT41B", 0.006, 0.040, 2.400, 2.25/Ibase400 )
Line(n, d, "CL16","CT32A", "FT42" , 0.006, 0.050, 2.400, 2.25/Ibase400 )
Line(n, d, "CL17","CT32B", "FT44A", 0.010, 0.040, 2.000, 2.25/Ibase400 )
Line(n, d, "FL1" ,"FT43B", "FT44A", 0.002, 0.015, 0.500, 4.37/Ibase400 )
Line(n, d, "FL2" ,"FT41A", "FT42" , 0.003, 0.030, 0.900, 2.25/Ibase400 )
Line(n, d, "FL3" ,"FT42" , "FT43A", 0.001, 0.010, 0.300, 6.25/Ibase400 )
Line(n, d, "FL4" ,"FT43A", "FT45" , 0.001, 0.010, 0.300, 6.25/Ibase400 )
Line(n, d, "FL5" ,"FT45" , "FT47A", 0.001, 0.015, 0.500, 4.37/Ibase400 )
Line(n, d, "FL6" ,"FT41A", "FT61" , 0.006, 0.045, 1.300, 2.00/Ibase400 )
Line(n, d, "FL7" ,"FT42" , "FT50A", 0.002, 0.020, 0.600, 4.37/Ibase400 )
Line(n, d, "FL8" ,"FT42" , "FT50B", 0.002, 0.020, 0.600, 4.37/Ibase400 )
Line(n, d, "FL9" ,"FT43B", "FT47B", 0.002, 0.020, 0.600, 4.37/Ibase400 )
Line(n, d, "FL10","FT61" , "FT62" , 0.002, 0.020, 0.600, 4.37/Ibase400 )
Line(n, d, "FL11","FT50A", "FT62" , 0.011, 0.080, 2.400, 2.00/Ibase400 )
Line(n, d, "FL12","FT62" , "FT63A", 0.003, 0.030, 0.900, 2.25/Ibase400 )
Line(n, d, "FL13","FT62" , "FT63B", 0.003, 0.030, 0.900, 2.25/Ibase400 )
Line(n, d, "FL14","FT50A", "FT51" , 0.004, 0.040, 1.200, 2.25/Ibase400 )
Line(n, d, "FL15","FT50B", "FT51" , 0.004, 0.040, 1.200, 2.25/Ibase400 )
Line(n, d, "FL16","FT42" , "FT44B", 0.002, 0.020, 0.600, 4.37/Ibase400 )
Line(n, d, "RL1" ,"RT132B","FT142A",0.038, 0.280, 0.060, 1.50/Ibase130 )
Line(n, d, "RL2" ,"RT132A","FT142B",0.038, 0.280, 0.060, 1.50/Ibase130 )
Line(n, d, "RL3" ,"RT131B","FT142A",0.010, 0.080, 0.016, 3.12/Ibase130 )
Line(n, d, "RL4" ,"RT131A","FT142B",0.010, 0.080, 0.016, 3.12/Ibase130 )
Line(n, d, "RL5" ,"RT131A","RT133", 0.010, 0.060, 0.012, 4.37/Ibase130 )
Line(n, d, "RL6" ,"RT131B","RT133", 0.010, 0.060, 0.012, 4.37/Ibase130 )
Line(n, d, "RL7" ,"RT132B","FT150A",0.050, 0.300, 0.060, 1.50/Ibase130 )
Line(n, d, "RL8" ,"RT133", "FT150A",0.015, 0.120, 0.025, 2.25/Ibase130 )
Line(n, d, "RL9" ,"RT133", "FT150B",0.015, 0.120, 0.025, 2.25/Ibase130 )

#Transformer(net, defaults, Name, busbar1, busbar2, R, X, B, Imax, tap_changer_steps, current_TC_step)

Transformer(n, d, "TRAFO1", "CT11A", "CT111" , 0.0, 0.008, 0.0, 1e+200, (1.0,) , 1)
Transformer(n, d, "TRAFO2", "CT12B", "CT112" , 0.0, 0.008, 0.0, 1e+200, (1.0,) , 1)
Transformer(n, d, "TRAFO3", "CT22A", "CT122" , 0.0, 0.012, 0.0, 1e+200, (1.0,) , 1)
```

```
Transformer(n, d, "TRAFO4", "FT42" , "FT142A", 0.0, 0.010, 0.0, 1e+200,
      (1.0,) , 1)
Transformer(n, d, "TRAFO5", "FT42" , "FT142B", 0.0, 0.010, 0.0, 1e+200,
      (1.0,) , 1)
Transformer(n, d, "TRAFO6", "FT50A", "FT150A", 0.0, 0.010, 0.0, 1e+200,
      (1.0,) , 1)
Transformer(n, d, "TRAFO7", "FT50B", "FT150B", 0.0, 0.010, 0.0, 1e+200,
      (1.0,) , 1)
Transformer(n, d, "TRAFO8", "CT31" , "CT231" , 0.0, 0.010, 0.0, 1e+200,
      (1.0,) , 1)

#switches between busbars need to be defined manually.
#Switch(n, d, Name, node1, node2, state )

Switch(n, d, "CB_CT11" , "CT11A" , "CT11B" , False)
Switch(n, d, "CB_CT12" , "CT12A" , "CT12B" , False)
Switch(n, d, "CB_CT22" , "CT22A" , "CT22B" , False)
Switch(n, d, "CB_CT32" , "CT32A" , "CT32B" , False)
Switch(n, d, "CB_CT72" , "CT72A" , "CT72B" , False)
Switch(n, d, "CB_FT41" , "FT41A" , "FT41B" , False)
Switch(n, d, "CB_FT43" , "FT43A" , "FT43B" , False)
Switch(n, d, "CB_FT44" , "FT44A" , "FT44B" , False)
Switch(n, d, "CB_FT47" , "FT47A" , "FT47B" , False)
Switch(n, d, "CB_FT50" , "FT50A" , "FT50B" , False)
Switch(n, d, "CB_FT63" , "FT63A" , "FT63B" , False)
Switch(n, d, "CB_FT142", "FT142A", "FT142B", False)
Switch(n, d, "CB_FT142", "FT142A", "FT142B", False)
Switch(n, d, "CB_FT150", "FT150A", "FT150B", False)
Switch(n, d, "CB_RT131", "RT131A", "RT131B", False)
Switch(n, d, "CB_RT132", "RT132A", "RT132B", False)
Switch(n, d, "CB_RT132", "RT132A", "RT132B", False)

if d. Initial_state =="NS":
        #Generation_state(net,Name,on,Vset,Pset)
        Generation_state (n,"FT63−G11pv",True,1.0,0.0)
        Generation_state (n,"CT72−G1pv",True,1.0,0.0)

elif d. Initial_state =="N":
        #Generation_state(net,Name,on,Vset,Pset)
        Generation_state (n,"CT72−G1pv",True,1.0,0.0)

```

```
elif d.Initial_state =="half":
        #Generation_state(net,Name,on,Vset,Pset)
        Generation_state(n,"AT111-Gpv",True,1.060000,3.000000)
        Generation_state(n,"AT121-Gpv",True,1.060000,5.500000)
        Generation_state(n,"AT131-Gpv",True,1.060000,0.000000)
        Generation_state(n,"CT112-G1pv",True,1.060000,1.000000)
        Generation_state(n,"CT112-G2pv",True,1.060000,3.000000)
        Generation_state(n,"CT122-Gpv",True,1.060000,2.000000)
        Generation_state(n,"AT241-G1pv",True,1.030000,0.000000)
        Generation_state(n,"AT241-G2pv",True,1.030000,0.000000)
        Generation_state(n,"CT11-G1pv",True,1.030000,2.000000)
        Generation_state(n,"CT11-G2pv",True,1.030000,3.000000)
        Generation_state(n,"CT12-G1pv",True,1.030000,3.000000)
        Generation_state(n,"CT12-G2pv",True,1.030000,2.000000)
        Generation_state(n,"CT21-Gpv",True,1.030000,2.500000)
        Generation_state(n,"CT31-Gpv",True,1.030000,0.000000)
        Generation_state(n,"CT71-Gpv",True,1.030000,3.000000)
        Generation_state(n,"CT72-G1pv",True,1.030000,7.500000)
        Generation_state(n,"CT72-G2pv",False,1.030000,7.500000)
        Generation_state(n,"CT72-G3pv",False,1.030000,2.500000)
        Generation_state(n,"CT72-G4pv",False,1.030000,2.500000)
        #close  switches
        Set_switch_state_re(n, True, [".*CT.*", ".*AT.*"])
        #open some switches
        Set_switch_state_re(n, False, [".*-EK.*", ".*-X.*"])
        Set_switch_state_re(n, False, [".*CL12.*", ".*CL14.*", ".*CL15.*"
            , ".*CL16.*", ".*CL17.*"])
        Set_switch_state_re(n, False, ["CB_CT72-G2","CB_CT72-G3","
            CB_CT72-G4"])
        #close  some switches
        Set_switch_state_re(n, True, ["CB_CT32-X1", "CB_CT21-X1"])


elif d.Initial_state =="full":
        #Generation_state(net,Name,on,Vset,Pset)
        Generation_state(n,"AT111-Gpv",True,1.060000,3.000000)
        Generation_state(n,"AT121-Gpv",True,1.060000,5.500000)
        Generation_state(n,"AT131-Gpv",True,1.060000,4.000000)
        Generation_state(n,"CT112-G1pv",True,1.060000,1.000000)
        Generation_state(n,"CT112-G2pv",True,1.060000,3.000000)
        Generation_state(n,"CT122-Gpv",True,1.060000,2.000000)
```

```
    Generation_state (n,"RT131−Gpv",True,1.060000,1.800000)
    Generation_state (n,"RT132−G1pv",True,1.060000,3.000000)
    Generation_state (n,"RT132−G2pv",True,1.060000,0.600000)
    Generation_state (n,"AT241−G1pv",True,1.030000,3.500000)
    Generation_state (n,"AT241−G2pv",True,1.030000,4.000000)
    Generation_state (n,"CT11−G1pv",True,1.030000,2.000000)
    Generation_state (n,"CT11−G2pv",True,1.030000,3.000000)
    Generation_state (n,"CT12−G1pv",True,1.030000,3.000000)
    Generation_state (n,"CT12−G2pv",True,1.030000,2.000000)
    Generation_state (n,"CT21−Gpv",True,1.030000,2.500000)
    Generation_state (n,"CT31−Gpv",True,1.030000,3.100000)
    Generation_state (n,"CT71−Gpv",True,1.030000,3.000000)
    Generation_state (n,"CT72−G1pv",True,1.030000,7.500000)
    Generation_state (n,"CT72−G2pv",False,1.030000,7.500000)
    Generation_state (n,"CT72−G3pv",False,1.030000,2.500000)
    Generation_state (n,"CT72−G4pv",False,1.030000,2.500000)
    Generation_state (n,"FT41−Gpv",True,1.030000,0.000000)
    Generation_state (n,"FT44−G1pv",True,1.030000,6.300000)
    Generation_state (n,"FT44−G2pv",True,1.030000,0.600000)
    Generation_state (n,"FT47−G1pv",True,1.030000,5.400000)
    Generation_state (n,"FT47−G2pv",True,1.030000,5.400000)
    Generation_state (n,"FT51−G1pv",True,1.030000,5.000000)
    Generation_state (n,"FT51−G2pv",True,1.030000,5.000000)
    Generation_state (n,"FT61−Gpv",True,1.030000,1.700000)
    Generation_state (n,"FT62−G1pv",True,1.030000,5.300000)
    Generation_state (n,"FT62−G2pv",True,1.030000,1.700000)
    Generation_state (n,"FT62−G3pv",True,1.030000,1.700000)
    Generation_state (n,"FT63−G1pv",True,1.030000,5.300000)
    Generation_state (n,"FT63−G2pv",True,1.030000,5.300000)

    #close all switches
    Set_switch_state_re (n, True, [".∗"])
    #open a few switches to reactive compensation
    Set_switch_state_re (n, False, [".∗−EK.∗",".∗−X.∗"])
    Set_switch_state_re (n, True, ["CB_CT32−X1", "CB_CT21−X1", "CB_FT62−X1"])

else :
    print "error␣unknown␣initial␣state"
```