Abstract

History has shown that power transmission networks disruptions can cause large blackouts that result in significant (both human and economic) losses. The list of notable large scale outages worldwide through history is large, and today a recognised necessity exists to prevent such blackouts by making power transmission networks more reliable. We respond to such necessity by studying how to find a low cost expansion plan such that, once applied, the expanded power transmission network operates normally not only on its (future) projected base state, but also on states that result from potential disruptions predicted by a given reliability criterion (thus becoming more reliable).

The ability to find this kind of expansion plans (unfortunately) requires us to solve extremely challenging combinatorial optimisation problems subject to very complex constraints which may no longer be relaxed in the future, as traditionally done, due to the nature of future applications. The underlying optimisation problem asks to find the expansion plan of minimum investment cost (that is, a cost that can easily reach the order of millions to billions of dollars) such that the expanded network satisfies two sets of constraints: the set of non-convex non-linear constraints (or ‘AC constraints’ for short in this abstract) that Ohm’s law imposes on any AC power transmission network, and a given set of operational constraints that are a necessity for the stable operation of the power system.

The complex AC constraints not only grow with the number of nodes in the network, but also involve, among a large number of model variables, both discrete and continuous decision variables that appear in terms that grow in turn with the number of edges in the network. With no surprise, the resulting program is NP-hard and its intractability is dealt in the literature, most of the time, by using systematic or stochastic, global or local search approaches based on the popular linearised DC approximation of the AC constraints (or ‘LDC constraints’ for short in this abstract), or by using an optimisation heuristic approach that tries to satisfy the proper AC model (that is, to find ‘AC feasible’ solutions).

The aim of this work is to try to work out an optimisation algorithm by using both conventional systematic search and unconventional local search techniques to find low cost AC feasible expansion plans that satisfy the $n - 1$ reliability criterion. In other words, we aim here to find an algorithm to expand the transmission network at the lowest possible cost such that, once expanded, the network operates normally in both its (future) projected base state and whenever one single circuit is disrupted.
We use to this end an approximation of the AC constraints called the ‘LPAC constraints’ that are linear, yet, in contrast to the widely used LDC constraints, take into account both reactive power and voltage root mean square magnitudes (as required by several applications). Reactive power and voltage magnitudes will become more important for the transmission network expansion problem (and other applications) in the future; for instance, due to the growing incorporation of renewable energy sources. Furthermore, solutions produced by incorporating reactive power and voltage magnitudes into the involved approximation are of better quality in terms of AC feasibility, and linear approximations lead to models of linear nature that are amenable to the existent state-of-the-art implementations of optimisation algorithms for both linear and mixed-integer linear programming (which enjoy decades of research in both academy and industry).

In a first series of experiments, we study the expansion problem in the context of transmission network expansion planning (‘TNEP’ for short in this abstract) for a projected base state. Then, in a second series of experiments, we attempt to incorporate the $n - 1$ reliability criterion by using an evolutionary algorithm. We observe in the results a significant gap between the investment cost of expansion plans obtained with LDC approximations and AC heuristics that can be bridged by using the LPAC power-flow constraints. We show that solutions obtained for TNEP using the LPAC power-flow constraints are AC feasible and have none or minimal operational constraint violations.

Finally, we demonstrate the LPAC-TNEP formulation strength by jointly optimising circuit expansion with reactive power compensation resulting in solutions with no violations at all. By exploring a genetic algorithm approach to incorporate the $n - 1$ reliability criterion (a much harder problem), we find suboptimal solutions with no violations and describe which parameters seem to work better. Our case study suggests that the underlying TNEP formulation has significant impacts on the proposed expansion plans and that the LPAC approximation is a good mechanism to finding expansion plans of high quality with respect to the AC model even considering the $n - 1$ reliability sub-networks.
To my brother, Alejandro,
for teaching me how to play football;
and to my sister, Paola,
for showing me how to be brave.
Declaration

This is to certify that:

i. the thesis comprises only my original work towards the MPhil except where indicated in the Preface,

ii. due acknowledgement has been made in the text to all other material used,

iii. the thesis is less than 50 thousand words in length (exclusive of tables, bibliographies, and appendices).
Preface

This thesis is the result of my first timid steps into the world of professional investigation where I have experienced for the first time a complete process of research. As a research student at The University of Melbourne holding a NICTA funded scholarship, I have carried out this MPhil thesis project within the broad interests of National ICT Australia.

The project was supervised by Pascal Van Hentenryck, and the findings in the context of the transmission network expansion planning problem (that appear in the second part of this document) were published in the Power Systems Computation Conference (PSCC) 2014 proceedings, pages 1 to 8. The paper was entitled ‘Transmission Network Expansion Planning: Bridging the gap between AC heuristics and DC approximations’ and co-authored between Russell Bent, Carleton Coffrin, Rodrigo Gumucio (that is, me), and Pascal Van Hentenryck.

The required clarification in this preface about authors’ contribution to any material in that paper (not always reflected in an alphabetically ordered list of authors) that simultaneously appears inside chapters of this thesis document follows: conception, execution, and interpretation of the results were mostly done by Rodrigo and Carleton, supervised by Pascal, all committed to ensure quality and originality, with Rodrigo undertaking most of the tasks.

Russell analysed our test cases, run some benchmark experiments to compare with their discrepancy bounded local search algorithm for expansion planning, and contributed positively in the discussion being him the most experienced among us in trying to solve the problem at hand. The part of the study on a three-bus test case, showing the benefits of the LPAC model for expansion planning over other competitor power-flow models, involving second order cone and non-linear constraints was mostly done by Carleton.

I also submitted an extended abstract about what I was planning to achieve while incorporating the $n-1$ reliability criterion to the University of Melbourne CIS 2014 Doctoral Colloquium, and it was published in its proceedings. This one-pager and (the more important) PSCC paper are attached to Appendix C for reference.

The programming tasks to run all of the experiments were extensive and non-trivial; since the source code is neither compact nor elegant enough to print (and attach it to an appendix), it was annexed in a DVD when submitting this document. The programming language of choice was Python with (not exclusively) the following libraries: NumPy, SciPy, GurobiPy, NetworkX, and PyPower.
In the first part of this document, the AC power-flow constraints are derived from Ohm’s law. The construction of the AC power definition from Ohm’s law (that is to say, roughly, equations 1 to 18) was crafted (by me) from bits and pieces I found in the World Wide Web by linking them and filling the gaps (a non-trivial task). The derivation of the AC power-flow constraints from the definition of AC power (that is to say, roughly, equations 19 to 41) is sourced from both Andersson’s ‘State Estimation in Electric Power Systems – A generalized approach’ [1] and Monticelli’s ‘Modelling and Analysis of Electric Power System’ [24] books. To assist the discussion in this context, I found necessary to include a diagram of the unified π-model for a circuit. Drawing such a diagram too creatively would destroy the nice π shape of the model. Therefore, Figure 2.2 shows similarities with the analogous figures in both sources (namely, Figure 4.7 and Figure 2.9 respectively) due to the standard symbols.

This thesis document was typed using LATEX.
Acknowledgments

I would like to thank both The University of Melbourne and National ICT Australia: to the former for letting me experience a bit of graduate school through a Melbourne International Fee Remission Scholarship and to the latter for letting me both use their infrastructure and experience professional research as a funded MPhil student. I thank Pascal Van Hentenryck and Carleton Coffrin for their guidance; thanks to the Optimisation Research Group for both the live conference streaming of research seminars and the optimisation summer school. I am also thankful to Peter Stuckey and to James Bailey for participating in the required formal meetings and commenting on my progress. In particular, thanks to Peter for skimming a draft of this document and providing feedback.
Contents

Preface 9
List of Tables 15
List of Figures 17
List of Models and Algorithms 19
Chapter 1. Introduction 21

Background 37
Chapter 2. Theory basics 39
Chapter 3. Literature review 53
Chapter 4. Description of the problem 65
Chapter 5. Derivation of the research question 71

Experiments 77
Chapter 6. Practice basics 79
Chapter 7. Design of the experiments 91
Chapter 8. Results, analysis, and interpretation 105
Chapter 9. Conclusion and future work 115
Bibliography 119

Appendices 123
Appendix A. Constraints derivations 125
Appendix B. Study and test cases 135
Appendix C. Papers and a few more tables 137
Afterword 151
Index 153
List of Tables

1.1 Summary of notation and terminology 35

3.1 Summary of TNEP papers using some accurate alternative form of the AC model 63

5.1 AC power-flow solution for the 3-bus network (preliminary study) 72
5.2 Results for the 3-bus network expansion planning problem with one low thermal limit circuit (preliminary study) 73
5.3 Results for the 3-bus network expansion planning problem with tight voltage bounds (preliminary study) 74
5.4 Best known/best found objective values comparison for DC-TNEP solutions for the g1, g2, g3 and g4 networks (preliminary study) 76

8.1 Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP 106
8.2 Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP incorporating the constraint tightening procedure 107
8.3 Solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP for perfect voltage profile networks 109
8.4 Active and reactive power injection comparison for solutions produced for the networks with perfect voltage profile 109
8.5 Evaluation of solutions produced by LPAC-TNEP along the Pareto frontier for the 24-bus network with perfect voltage profile 110
8.6 Evaluation of LPAC-TNEP solutions produced for the g1, g2, g3, g4, and gf networks by using constraint tightening 112
8.7 Solutions produced for TNEP by using the genetic algorithm 113
8.8 Solutions produced for TNEP with all n – 1 sub-networks by using the genetic algorithm 113

C.1 Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP with constraint tightening at 0%. 149
C.2 Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP with constraint tightening at 5%. 149
C.3 Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP with constraint tightening at 10%. 149
C.4 Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP with constraint tightening at 15%. 150
C.5 Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP with constraint tightening at 20%. 150
# List of Figures

1.1 Structure of the electric power system 22  
1.2 24-bus IEEE reliability test system 23  
1.3 A graph representation for the 24-bus IEEE reliability test system 24  
1.4 Nonintuitive behaviour of power-flows after network expansion 29  
    
2.1 Current, voltage, and power in an AC system 46  
2.2 Model for a circuit between two buses \( n \) and \( m \) 49  
    
4.1 Constrained network flow example 67  
    
5.1 3-bus network data and representation (preliminary study) 72  
5.2 Non-linear and LPAC (linear) power-flow constraints correlation plots for the 24-bus OPF benchmark (preliminary study) 75  
    
6.1 The knapsack problem and one problem instance data (example) 83  
6.2 Knapsack problem instance branch-and-bound solution tree (example) 84  
6.3 Crossover operation 88  
6.4 Mutation operation 88  
    
8.1 Comparison of number of circuits in the expansion plans produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP 108  
8.2 Reactive power injection needed to keep a perfect voltage profile 110  
8.3 Part of the Pareto frontier solutions for the 6 and 24-bus networks produced by LPAC-TNEP 111  
    
A.1 Model for a transmission line circuit between two buses \( n \) and \( m \) 128  
A.2 Model for a transformer circuit between two buses \( n \) and \( m \) 128  
A.3 Non-linear and LPAC (linear) power-flow constraints correlation plots for the 14-bus OPF benchmark 132  
A.4 Non-linear and LPAC (linear) power-flow constraints correlation plots for the 57-bus OPF benchmark 133
List of Models and Algorithms

4.1 AC-TNEP model 69
6.1 Branch-and-bound algorithm 82
6.2 Genetic algorithm 89
7.1 Thermal limit constraints piecewise linear approximation 95
7.1 LPAC-TNEP model 96
7.2 DC-TNEP model 97
7.2 HAC-TNEP algorithm 98
CHAPTER 1

Introduction

Combinatorial optimisation and power systems engineering are both, as many others, vast and complex disciplines with topics that usually need to be introduced gently. Combinatorial optimisation itself is a fascinating field where, from my point of view, maths and computer science meet each other to put in evidence what we end up doing most of the time in the problem solving process: capturing a problem in a model and applying algorithms to draw conclusions from the model and ultimately find the best possible solution to the problem.\footnote{With high level declarative programming languages in the front-end and clever algorithms (or ideally one master optimisation algorithm) implemented in the back-end, the holy grail of computer science may be found one day: a piece of software that lets us humans (easily) express any problem and then lets the hardware (efficiently) find the solution for us.} A wide range of optimisation problems arise in the power systems domain, another interesting field, where solving problems and applying their solutions to the physical world requires, from my point of view, a great deal of technical expertise together with a strong intuition grounded in the laws of physics (both probably acquired only after several years of exposure to both theory and practice). Fortunately, it is not necessary to be an expert in one or both of these fields to understand the research presented herein. In the first part of this thesis, we arm the reader with all the essentials required to understand the second part. That is to say, we include all the necessary background in the next chapters to proceed later with the core and synthesis chapters of the thesis.

This first introductory chapter gives only a few basics of both power transmission networks and combinatorial optimisation,\footnote{If the reader does not find this chapter straightforward to read, we suggest to skip it for now and come back after understanding the background material presented in the next chapter and reviewing the basic definitions of combinatorial optimisation somewhere else. See also Table 1.1.} and links them together as a way to put our study in context and to develop an informal understanding of the optimisation problem, the methods, and the overall purpose of our research.

First, we explain in simple terms the structure of a power system to give an idea of where the power transmission network stands within this system. The power transmission network itself is our object of study, so we abstract a graph from it and explain its components in detail. Second, we give a notion of how the power transmission network is constrained by both the power-flow and the operational constraints; we make concrete what we mean by saying that a transmission network ‘operates normally’ and explain how we determine if a network is in such a desirable condition by means of simulation. Third, we explain the expansion/reliability problem, put in evidence its combinatorial nature, formulate it...
informally as a combinatorial optimisation problem, and give a notion of how we try to solve it. Finally, we frame the resulting formulation within the known hierarchy of combinatorial optimisation problems, and, at last, we make a general statement of the research question, explain the aim and scope of our research, mention our contributions, and give an overview of how the next chapters are organised.

The electric power system, as shown in Figure 1.1 consists of four subsystems: generation, transmission, distribution, and consumption. The generation subsystem is mainly composed of generating units that convert primary sources of energy (like wind, water falls, or nuclear reactions to name a few) into electrical energy that is injected into the system. The transmission network, a mesh network which spans long distances and operates at high voltages carries the energy from generating units to points for distribution to the consumers. The distribution network, a network with usually radial topology that operates at lower voltages, is responsible for carrying the energy the last kilometers from transmission to consumption. Finally, the consumers, also called loads, receive this electrical energy to consume and get the benefits of light bulbs, heaters, motors, and any other appliance or electronic device. Electrical energy is consumed by converting into other forms of energy and power is the rate of such conversion. As elaborated later, any load consumes either active or reactive power, or a combination of both.

\footnote{If you think about the generating units as producers, you get a producers-consumers pair.}
The overall goal of the power transmission network within the power system is to transmit the energy or (less accurately) power generated by generating units to points where this energy can ultimately be consumed by loads. In this sense, for a given topology of a power transmission network, the flow of power should ensure that the generation of power meets the system’s demands. This constraint reflects well the network’s purpose, but generation should meet the system’s demands by satisfying, in addition, other more complex operational and power-flow constraints.

Before giving notions of the more complex constraints, it is convenient to zoom in on a power transmission network to reveal its basic components. Figure 1.2 shows a 24-bus network known in the literature as the 24-bus IEEE RTS [35].

**Figure 1.2.** Graphical representation of the 24-bus IEEE reliability test system (the original data for this network was first published by the Reliability Test System Task Force of the Application of Probability Methods Subcommittee [35]. This representation is adapted from the PM Subcommittee’s Figure 1 [35]). A power transmission network is composed by both buses (here nodes) and circuits (here edges). Both loads and generating units connect to buses, and circuits can be either transmission lines or transformers.
This network has 24 buses connected by 38 circuits at two different voltages: 138 and 230 [kV]. The peak demand is of 2850 [MW] active and 580 [MVAr] reactive power, and the total generation capacity is of 3405 [MW] active and from -535 to 1776 [MVAr] reactive power. The power generation capacity is therefore sufficient to satisfy the peak demand of power. Note in Figure 1.2 that buses are actually load/generation buses in the sense that either loads or generating units can connect to a bus. Loads (\(\rightarrow\)) are aggregated at each bus, but generation of power (\(\rightarrow\)), although aggregated as well in the graphical representation, is actually given by one or more generating units connecting to the bus. Different generating units may have different generation capacities or cost models. Devices that only generate reactive power are also sometimes connected to a bus; we call such devices synchronous condensers (see bus 14 in Figure 1.2), and their purpose is to act as voltage corrective devices. For the same purpose, sometimes shunt elements are connected to buses as well (for example, see the reactor (\(\rightarrow\)) in bus 6 of Figure 1.2). Note also that some circuits in Figure 1.2 (see buses 11, 12, and 24) are actually 230/138 [kV] transformers (\(\rightarrow\)) (observe that the top part of the network is at 230 [kV] while the low part is at 138 [kV] accordingly). A circuit is then either a transmission line or a transformer,\(^4\) and we also use the notion of a corridor.\(^5\) A corridor is a walk (of length \(\leq 1\)) between two buses and there can be zero, one, or more circuits in a corridor. A modern power transmission network may have other components.

\(^{4}\)The curious reader may find that in the original 24-bus IEEE reliability test system (see buses 6 and 10 in PM Subcommittee’s Figure 1 [33]), circuits can also be cables, but we ignore this distinction here and take cables as lines.

\(^{5}\)As commonly done in the literature. The curious reader may notice oval symbols (named from A to G) around some pairs of circuits in PM Subcommittee’s [33] Figure 1 that indicate a ‘common right of way’, but we take here the notion of a corridor instead (as usually done).
1. INTRODUCTION

as well (for instance, FACTS\textsuperscript{6} devices), but the description in this paragraph is a good representation of the network components present in our study cases.

Figure 1.2 already devises a graph abstraction \((\mathcal{B}, \mathcal{C})\) for the power transmission network that is made more explicit in Figure 1.3. The finite set \(\mathcal{B}\) of nodes or vertices is the set of buses, and the set \(\mathcal{C}\) of edges whose elements are subsets of \(\mathcal{B}\) of cardinality two is the set of circuits.\textsuperscript{7} Note that the simplified graph representation in Figure 1.3 shows edges as corridors instead of circuits.\textsuperscript{8} The circuits within the same corridor are usually identical which means that the arguments for the circuit parameters (which are model variables) are usually identical for the circuits within the same corridor. One important circuit parameter is the thermal limit of a circuit. Different amounts of power flow through different circuits\textsuperscript{9} and this is highly influenced by the topology of the network. However, thermal limits should be respected at all times. If the topology changes, the amount of power flowing in the edges will change, and sometimes with undesirable effects like violations in the thermal limits of the circuits (that are only part of the operational constraints).

The graph in Figure 1.3 roughly shows our abstraction for a power network \(\mathcal{PN}\). Although not shown in Figure 1.3, but explicit in Figure 1.2, there can be many sources and terminals in \(\mathcal{B}\). Sources are buses with generating units connected to them and terminals are buses with connected loads. We call them respectively generator and load buses. Thus, a bus is either a generator bus (for example, bus 22 in Figure 1.2), or a load bus (for example, bus 20 in Figure 1.2), or both (that is, a generator-load bus (for example, bus 16 in Figure 1.2), or none (that is, just a bus; for example, bus 17 in Figure 1.2). Similarly, a circuit \(n-m\) is either a transmission line circuit (for example, circuit 1-2 in Figure 1.2) or a transformer circuit (for example, circuit 3-24 in Figure 1.2) with ends at both buses \(n\) and \(m\). Among many other circuit parameters, any circuit \(n-m\) has a given thermal limit \(\bar{s}_{nm}\) and a resistance \(r_{nm}\). Due to the resistance, the power at the \(n\) end of the circuit is not the same as the power at the \(m\) end of the same circuit \(n-m\). That is, \(\overline{p}_{nm} \neq \overline{p}_{mn}\). In a network flow for a graph, as the one in Figure 1.3, we expect both that all the thermal limits are respected (that is to say, we expect that \(\max(|\overline{p}_{nm}| + |\overline{p}_{mn}|) \leq \bar{s}_{nm}\) for any circuit \(n-m\)\textsuperscript{10} and that all the power flowing into any bus \(\tilde{n}\) is equal to all the power flowing out from \(\tilde{n}\) (including the power injected by generating units to \(\tilde{n}\) and the power withdrawn from \(\tilde{n}\) by loads). This could look like a standard flow problem but there is more behind the scenes.

For a given topology, the way that power flows in the network (i.e., the amount and direction of power flowing in the circuits) is determined by Ohm’s law. As a result, the power-flows are a function of the bus voltages and several circuit

\textsuperscript{6}FACTS stands for flexible AC transmission system.

\textsuperscript{7}Since such a graph has repeated edges, we can say this is really a multigraph.

\textsuperscript{8}Only to assist the discussion. In our experimental models, edges are circuits not corridors.

\textsuperscript{9}Although not so easy to see, the diverse widths of the edges in the graph of Figure 1.3 show that the amount of power flowing through different circuits is different.

\textsuperscript{10}Thermal limit constraints can be seen as a sort of capacity constraints.
parameters excluding the thermal limits. This means that for a given topology, if
the configuration is not adequate, the enforced power-flow constraints can produce
overloaded circuits (that is, circuits that exceed their thermal limits) that may
eventually melt down. We say the power-flow is enforced because Ohm’s law is a
law of nature; thus the power-flow constraints resulting from this law are enforced
by nature. We can do very little about this, at most try to indirectly influence its
behaviour which is hard to do because these power-flow constraints are non-linear.

The non-linearity of the power-flow constrains, that in turn involves several
decision variables, is what makes this problem depart from a more traditional net-
work flow problem. An informal way of thinking about the problem we are trying
to describe is that of having a graph, like the one in Figure 1.3, which is a variable
in the sense that its topology can change (for example, due to circuit disruptions or
expansions in the network) and the two sets of constraints on this graph-variable
that need to be satisfied: the mentioned non-linear power-flow constraints\footnote{Note throughout the thesis, to express ourselves better, we refer to these constraints as ‘AC constraints’, ‘AC power-flow constraints’, ‘non-linear power-flow constraints’, or ‘non-linear non-convex power-flow constraints’, or other minor variations. All of them are equivalent.} and
the operational constraints (for example, those mentioned requiring that no circuit
exceeds its thermal limit). We try to find an optimal configuration, a sort of as-
signment to a graph-variable for a power transmission network such that once the
power-flow constraints are enforced by nature, all of the operational constraints are
satisfied (almost as a side effect). When the enforced flow of power makes the gen-
eration meet the demand with all of the operational constrains satisfied, we say the
network is operating normally. We make this more concrete in the next paragraph
by explaining how we find and measure the operational constraints violations.

Suppose for a moment that a network configuration is given only by the topol-
ogy of the network. There is a set of all possible topologies, and we choose one;
that is to say, we choose a configuration. We set up the network parameters with
the chosen configuration – under our assumption that means that we expand the
network but in a more general sense that properly means that we configure the
network. And finally, we run a simulation to enforce the power-flow constraints (or
informally, let the network run) so that we can see how the flow of power behaves
and whether the thermal limits are respected in all circuits (and more generally
whether all of the operational constraints are satisfied). If some circuits are over-
loaded, we measure to what degree they exceed their limits; i.e., we measure the
constraint violations in the thermal limit operational constraints (and more gener-
ally, we measure the quality of the configuration in terms of the violations in all of
the operational constraints). This sounds straightforward to do but the simulation
itself is an iterative process that may be computationally expensive and there are
other variables that need to be explained slightly more in detail.

Power is the product of voltage and current, and current can be obtained from
voltage. Therefore, the simulation is run to compute the bus voltages from where
circuit currents are determined and power-flows are made known. In an AC system, both voltage and current (modeled as complex numbers) have a time component. Therefore, the unknown bus voltages have two components as well: a $\tilde{e}$ magnitude and a $\theta$ phase angle. The latter is the time component, and the two of them need to be determined. Consistently, the AC power values also have two components: the $\bar{p}$ active (or real) power and the $q$ reactive (or imaginary) power. In order to run the simulation to solve for the voltages, we require exactly two of the four pieces of information\textsuperscript{12} per bus (in other words, to run a simulation we need to know two of these four \{$p_n, q_n, \tilde{e}_n, \theta_n$\} for any bus $n$).

What is typically known is the amount of power injected to or demanded from a bus, and voltages are usually fixed for generator buses which we also call voltage controlled buses for this reason. This means that we can classify buses based on what pieces of information are known for them, as $p$-$q$ and $p$-$\tilde{e}$ buses; that is to say, as non-generator and voltage-controlled buses. For voltage-controlled buses, we could probably also specify $\bar{p}$ and $q$ (hence use the same pieces of information for all buses). However, there are two problems that we only mention here. The first problem is related to the operational control of generator buses and the second problem is related to balancing power on the system. Due to the first problem, it is normal to specify $\tilde{e}$ instead of $q$ for all voltage controlled-buses, and, due to the second problem, we make an exception in one of the voltage-controlled buses for which we specify $\theta$ instead of $p$, and we call this single bus the slack bus or $\theta$-$\tilde{e}$ bus.

The simulation is then the process of taking all the known pieces of information of $p$-$q$ and $p$-$\tilde{e}$ buses, and of the slack bus (for which it is routine to set $\theta = 0$ as a time reference) and finding the unknown voltages (both magnitude and phase angle) for all buses. Unfortunately, this process is computationally expensive and sometimes the internal iterative methods do not converge causing the simulation to fail. Moreover, for running a simulation, we need to specify a full network configuration which is much more complex than specifying the topology alone as supposed for simplicity when starting the current discussion. To configure a network in order to run a simulation, we need: its topology, the active and reactive power at the $p$-$q$ buses, and the $p$ and $\tilde{e}$ values at the voltage controlled-buses. For the slack bus, we set free the value of $p$ (and use $\theta = 0$ instead). The slack bus is needed because the resistance of the circuits produces power that is dissipated (as heat) in the transmission network, therefore the amount of power generated by generating units does not match exactly the amount of power demanded by loads (it should exceed it). Since there is no way to tell in advance what the power transmission losses are going to be, the assumption made is that they are small enough such that a single generator bus (that is, the slack bus) can adjust its output (and take up the slack) to balance the $\bar{p}$ power in the system.\textsuperscript{13}

\textsuperscript{12}Less than two pieces of information makes the system unsolvable, more than two is redundant information.

\textsuperscript{13}The avid reader may ask about $q$ power balance. Indeed, both $\bar{p}$ and $q$ power should be balanced at all times. The reactive power generated by generating units needs to match the
The described simulation process is difficult because each value of real and reactive power can be consistent with many different possible combinations of voltages (and currents). To choose the correct ones, one needs to check each bus in relation to all adjacent buses and find values that are consistent throughout the system; this is already a combinatorial problem. In fact, if one thinks about problems this way, it becomes clearer that combinatorial problems are everywhere; combinatorial problems are ubiquitous in our society. This particular one is known as power-flow analysis and referred to as power-flow in the literature (or as load-flow in industry). Implementations of power-flow analysis algorithms that are ready to be used are available in existing packages like MatPower.\textsuperscript{14}

Finally, to check whether a power transmission network operates normally for a given configuration, we configure the network parameters as per the given configuration and let the network run. If the resulting voltage magnitude values fall within their bounds in all buses – yet another constraint – and the power-flow values of the circuits respect their thermal limit constraints, then we say the network operates normally. Otherwise, we can measure the constraint violations and have an idea about the quality of the given configuration. Faced with the event that the simulation fails, we do not say anything more than that the configuration is a bad configuration for our purposes.

Now consider what happens when a circuit is disrupted, or more generally, what happens when the status of a circuit changes (i.e., consider what happens when a circuit becomes enabled or disabled or, in other words, when a circuit changes its status from ‘off’ to ‘on’ or vice versa). This can happen, for example, due to lightning or repair work or more drastically, for example, due to natural disasters or adversarial attacks. The topology of the power transmission network changes. Therefore, the way that power flows through the network changes. In the new flow of power, if the network is not designed to be reliable, the operational constraints may become violated so that the network may no longer operate normally.

Faced with the event that an overloaded circuit turns off, the topology of the power transmission network changes; thus, a different flow of power is enforced by the non-linear power-flow constraints. In the new flow of power, for example, another circuit may become overloaded and turn off as well [4]. This may go on, and with a very small probability, overloaded circuits may turn off one after another resulting in a cascade effect. Such undesirable situation is terminated by disconnecting much of the demands thus producing blackouts. Blackout sizes may range from a building or a block to a city or the entire network.

Any blackout is a highly undesirable situation. In fact, the list of notable wide-scale power outages worldwide, through history, is large and it is known from there

\textsuperscript{14}MatPower is a MATLAB Power System Simulation Package, see http://www.pserc.cornell.edu/matpower/ – Accessed on October 2014.

reactive power demanded by loads plus the amount that is lost in the transmission circuits. In practice, since voltage magnitude is the explicit operational control variable in voltage controlled buses (in other words, since \( \tilde{e} \) is specified instead of \( q \)) all generators share the reactive slack.
1. INTRODUCTION

Circuit Capacity Overflow Usage

\[ (n, m, k) \quad \text{[MVA]} \quad \text{[MVA]} \quad \% \]

\begin{tabular}{cccc}
6 & 10 & 1 & 175.00 \quad 75.81 \quad 143.32 \\
6 & 10 & 2 & 175.00 \quad 75.81 \quad 143.32 \\
6 & 10 & 3 & 175.00 \quad 75.81 \quad 143.32 \\
6 & 10 & 4 & 175.00 \quad 75.81 \quad 143.32 \\
6 & 10 & 5 & 175.00 \quad 75.81 \quad 143.32 \\
\end{tabular}

\begin{tabular}{cccc}
2 & 6 & 1 & 175.00 \quad 0.09 \quad 100.05 \\
5 & 10 & 1 & 175.00 \quad 19.08 \quad 110.90 \\
6 & 10 & 1 & 175.00 \quad 118.31 \quad 167.61 \\
6 & 10 & 2 & 175.00 \quad 118.31 \quad 167.61 \\
6 & 10 & 3 & 175.00 \quad 118.31 \quad 167.61 \\
6 & 10 & 4 & 175.00 \quad 118.31 \quad 167.61 \\
\end{tabular}

(a) Before

(b) After

Figure 1.4. Thermal limit violations before (a) and after (b) adding one circuit to the most overloaded corridor 6-10 on an instance of the 24-bus IEEE RTS network. The way that the flow of power behaves upon adding one circuit can trick our intuition; for example, adding a redundant circuit may worsen the situation.

that blackouts can result in significant, both human and economic losses. Witness, for example, the blackouts produced by Hurricane Sandy\(^{15}\) which struck the cost of the United States of America (USA) in 2012. Estimates compiled from a variety of sources suggest that Sandy was responsible for at least 50 billion dollars in damage, and even worse, at least 87 deaths were indirectly associated with Sandy or its remnants from which about 50 were the result of extended power outages [6].\(^{16}\) A less severe example can be found in the blackouts produced by Ex Tropical Cyclone Oswald which caused, in 2013, the loss of power to over 250,000 customers in South East Queensland, Australia.

Natural disasters are not the only (potential) cause of power systems disruptions; for example, adversarial attacks are another acknowledged threat. In the context of Hurricane Sandy, a report released by the USA’s National Research Council [26] states that terrorist attacks could cause even more damage than natural disasters (like Sandy) producing blackouts of large regions for weeks or even months costing many billions of dollars in loss. A necessity exists therefore to prevent blackouts by making power transmission networks more reliable.

\(^{15}\)Category 3 at its peak (as measured using the 1-2-3-4-5 Saffir-Simpson Hurricane Wind Scale where 5 is the most severe); that is, a hurricane already considered major.

\(^{16}\)Power outages can lead to deaths, for example, from hypothermia during cold weather, falls in the dark by senior citizens, carbon monoxide poisoning, and car accidents [6].
We aim therefore to make a network more reliable by expanding it in such a way that the expanded network operates normally at a base configuration and satisfies a given reliability criterion. Expanding a power transmission network entails the addition of a set of new circuits to it which is typically expensive. We strive then to choose a set of circuits called an \textit{expansion plan}, for which the investment cost is minimum. The minimisation problem is difficult because, due to Ohm's law, it is subject to the non-convex non-linear power-flow constraints that not only grow with the number of buses in the network, but also involve, among a large number of model variables, both discrete and continuous decision variables in terms that grow with the number of circuits. Furthermore, since the power-flow constraints are enforced by nature, in the same way that we cannot predict what is going to happen when a circuit is disrupted, we cannot predict how the new flow of power behaves when adding one extra circuit to the network.

For example, Figure 1.4(a) shows a 24-bus network with a configuration that causes 4 circuits (in corridor 6-10) to be overloaded by 43.32% of their capacity. The intuitive action to take in order to avoid such overload is to add a new (extra) circuit to corridor 6-10. However, as shown in Figure 1.4(b), the resulting configuration causes the 5 circuits (in corridor 6-10) to be even more overloaded, now by 67.61% of their capacity. Moreover, since other two circuits (in different corridors) become overloaded as well, the resulting configuration is worse after expansion. One may give an intuitive explanation to this phenomena; for example, saying that power ‘likes’ to flow through corridor 6-10 so that adding another circuit to that corridor increases its conductance (in a sense) thus possibly increases the flow of power through it. However, such explanations, although useful in designing heuristics, are of limited use – adding circuits to overloaded corridors usually does work to mitigate the power-flow violations in that corridor. Expanding a power transmission network in order to get the desired effects is tricky to accomplish and harder to accomplish in an optimal way.

In a more broad perspective, a power transmission network may need to be expanded not only to make it more robust against potential threats, but also for other reasons; for instance, to satisfy a projected future energy demand. An overarching view of this problem is that of finding the lowest cost expansion plan such that the expanded network satisfies all of the constraints in any \textit{scenario} of a given set of scenarios:\footnote{Although the word ‘scenario’ may convey little information in some contexts here, we use it anyway to not lose the overarching view of the problem. Concretely, with this word we mean a network with a different demand or a network without some given components; in the latter case, we sometimes prefer to talk in terms of sub-networks.} for example, one scenario where the demand has increased or many scenarios defined by a given reliability criterion. A common reliability criterion is the one called \(n - k\) where each scenario is characterised by the failure of \(k\) (sometimes of different type) components of the network. However, the optimisation problem with a single scenario is already computationally difficult and therefore worth studying.
In order to experimentally explore this problem, we take the $n - 1$ reliability criterion where each scenario is given by the failure of one single circuit, and in a first series of experiments we study the problem with one scenario where a new projected future energy demand is given and the network needs to be expanded (so that the generation meets the projected demand). This base problem is known in the literature as Transmission Network Expansion Planning (TNEP), and coincidentally, it is also usually desired that a solution to the TNEP problem satisfies a reliability criterion, like the $n - 1$. Ideally, we would like to solve the problem of expanding the network at a minimum cost by using the non-linear power-flow constraints such that the expanded network is reliable with respect to an arbitrary set of scenarios. For example, we may be interested in those characterised by an $n - k$ reliability criterion with a small value for $k$.

The expansion problem is difficult and has an evident combinatorial nature. When selecting options for an expansion plan, we usually have a finite set of expansion-options to choose from.$^{18}$ In the expansion problem, we look for a subset of the expansion-options set such that the expanded network operates normally in all of the scenarios of a given set of scenarios. Since we want to do this at the minimum possible cost, the problem is a combinatorial optimisation problem asking to select the best subset of expansion-options (that is, the expansion plan of minimum cost) for which the expanded network operates normally in all of the scenarios (taken from a given scenarios set).

Suppose that we have a set $F$ of all expanded networks that operate normally in all the given scenarios. In our experimental study cases, we allow up to 6 parallel circuits per corridor. Suppose 1 of these 6 is a circuit-option; for a network with $n$ corridors, there would be $2^n$ possible expansions. Our 46-bus study case has 79 corridors, so even if we could check if an expanded network is in $F$ in one microsecond, it would take about thousands of millions of years to do this for all possible expanded networks and then select the optimum.

This last approach to solve the problem is naïve,$^{19}$ but it shows that the problem can in principle be solved with a computer; the best solution (given that one exists) can be found. However, since in practice it would take too long to find a solution, the approach is considered intractable. Here we aim to deal with its intractability by any means.$^{20}$ Moreover, for this problem, checking if a given network is in $F$, when possible, can be expensive; nevertheless, there are linear approximations and convex relaxations of $F$ proposed in the literature that can make the operation faster or even lead to mixed-integer linear or conic programs for which efficient implementations of global optimisation algorithms exist. Broadly speaking, our

---

$^{18}$In combinatorial problems we look for an object from a finite or possibly countably infinite set.

$^{19}$Or more than naïve it could seem ignorant; however, some small problems may be solved this way.

$^{20}$Some practical ways of deal with intractability include approximations, branch-and-bound, local search, dynamic programming, and others (and combinations of them).
experimental method consists of practical ways of trying to deal with intractability by using approximate power-flow constraints and measuring the quality of the solutions produced.

As in any combinatorial optimisation problem, besides $\mathcal{F}$ (the finite set of objects to choose from), an objective function $f$ is involved, say $f : \mathcal{F} \rightarrow \mathbb{R}$, so that an instance of the optimisation problem is actually a pair $(\mathcal{F}, f)$. Such an optimisation problem instance asks to find $\mathcal{PN}^* \in \mathcal{F}$ such that $f(\mathcal{PN}^*) \leq f(\mathcal{PN})$ for all $\mathcal{PN} \in \mathcal{F}$. The expanded network $\mathcal{PN}^*$ that we look for is the globally optimal solution, usually harder to find when there are many local optimal solutions (a situation that can happen in a non-convex set). Notice that what we have been calling an optimisation problem so far is actually an instance of an optimisation problem (as we just made precise). An optimisation problem is a set of instances of an optimisation problem. Therefore, the optimisation problem we try to solve here is by definition a collection of instances, yet we experiment with 6, 24, and 46-bus networks only (as commonly seen in the TNEP literature).

One type of optimisation problem that is particularly interesting is a linear program (LP). An instance of a LP is a pair $(\mathcal{G}, g)$ with $\mathcal{G} = \{x \in \mathbb{R}^n \mid A \cdot x = b \land x \geq 0\}$ and $g(x) = c' \cdot x$, where: $n$ and $m \in \mathbb{Z}^+$, $b \in \mathbb{Z}^m$, $c \in \mathbb{Z}^n$ and $c'$ is a row vector, and $A$ is an $n \times m$ matrix with elements in $\mathbb{Z}$. LPs are interesting because they have a combinatorial nature, and at the same time enjoy efficient implementations of algorithms that work remarkably well in large instances of such optimisation problems. Therefore, LPs can, in principle, solve some large purely combinatorial optimisation problems. Furthermore, when some of the $x$ decision variables of LPs are constrained to take integer values and problems become mixed-integer linear programs (or MILPs), industrial-strength implementations of algorithms exist as well that enjoy of significant research developed in the past decades. However, MIPs have a large complexity difference with respect to LPs; the difference due to the integrality constraints in LPs can be the difference between polynomial time (P) and NP complete.

Finally, notice that it is essentially the set of constraints of the problem which characterises the set of feasible solutions (and the applicable solution methods). For LP problems, all of the constraints in the set $\mathcal{G}$ are linear. However, in our problems, the set $\mathcal{F}$ is characterised, among others, by the non-linear non-convex power-flow constraints. This means that the problem we try to solve here is in practice much harder. That is, our expansion problem classifies as a general non-linear problem.
1. INTRODUCTION

The formulation of such an optimisation problem instance is:

\[
\text{Minimize } f(x) \\
\text{Subject to } g_i(x) \geq 0 \quad \text{for } i \in [1, \ldots, m] \\
h_j(x) = 0 \quad \text{for } j \in [1, \ldots, n]
\]

where \( f, g_i, \) and \( h_j \) are general functions of the decision variables \( x \) with domains in \( \mathbb{R}^n \). In some problems, \( f \) is convex, \( g_i \) are concave, and \( h_j \) are linear; thereby becoming convex programming optimisation problem instances. Such problems have the property that local optimality implies global optimality.\(^{26}\) In some problems, all \( f, g_i, \) and \( h_j \) are linear; thereby becoming linear programs. Every linear function is concave as well as convex, so every instance of LP is convex. Implementations of interior-point methods to solve linear and non-linear convex optimisation problems exist as well. When some of the decision variables are constrained to be integer, the model is said to be mixed; for example, mixed-integer linear programming or mixed-integer convex programming – these are much harder problems.

We aim to make a power transmission network more robust by expanding it such that it not only operates normally at a (future) base state but also satisfies a given reliability criterion. In other words, we look for the lowest possible cost expansion plan that produces an expanded network that satisfies all of the (non-linear non-convex) constraints inherent to the problem in all of the scenarios (in a given set of scenarios). To this end, we investigate what kind of solutions we can get by using the recently introduced LPAC (linear) power-flow constraints [9]. We run a first series of experiments to deal with the hardness of the problem with one scenario where a new projected future energy demand requires an expansion of the network (so that the generation can meet the projected demand). After that, we run a second series of experiments to try with the harder problem of expanding the scenarios set to include the sub-networks defined by the \( n - 1 \) reliability criterion.

The advantage of using the LPAC power-flow constraints as an alternative to the well known LDC constraints and the non-linear AC constraints is that they promise a good balance between the tractability of the former and the accuracy of the latter. In contrast with the non-linearity of the AC constraints, both DC and LPAC power-flow constraints are linear. Therefore, both of them enable us to formulate linear and mixed integer linear programs amenable to state-of-the-art solvers of industrial strength built upon decades of research in both academy and industry. However, the LDC constraints ignore reactive power and voltage magnitudes which can be unrealistic in some cases (for example, in cold start scenarios such as after a major blackout has occurred) leading to infeasible solutions. Recent work in the community started to consider the transmission network expansion planning problem with the full AC power-flow constraints (for example, Bent et

\(^{25}\)We prefer to use this more explicit form to formulate our optimisation problems than the pair.

\(^{26}\)Furthermore, they have the Kuhn-Tucker sufficient conditions for optimality.
In this direction, the LPAC constraints, as a more accurate approximation to the AC constraints (than the LDC constraints) can lead to more economically efficient expansion plans of much higher quality (in terms of AC feasibility).

The main contribution of this work is showing computationally that, for a case study composed of three standard benchmark power networks from the literature, the LPAC-based formulation for transmission network expansion planning provides solutions of much higher quality than those obtained for the DC-based formulation, sometimes obtaining solutions which are indeed AC feasible. We study the expansion problem with and without reliability constraints, including a variant with reactive compensation. The LPAC-based mixed-integer linear programs behave particularly well without the reliability constraints and on the problem variant with reactive power compensation. We try a genetic algorithm to handle the reliability constraints using the LPAC power-flow constraints, and observe that solutions found this way with and without the reliability constraints appear to require an investment of similar cost.

The thesis is organised in two parts as follows: the first part (that is, Chapters 2 to 5) presents most of the background material, and the second part (that is, Chapters 6 to 9) presents the core of the work. The bibliography and appendices follow the second part. The chapters are organised as follows:

- Chapter 2 develops a basic understanding of the electricity basics.
- Chapter 3 reviews the literature.
- Chapter 4 formalises the problem.
- Chapter 5 derives the research question.
- Chapter 6 gives basics of the chosen optimisation techniques.
- Chapter 7 designs the experiments.
- Chapter 8 displays, analyses, and interprets the results.
- Chapter 9 concludes and gives directions for future work.

The best way to read the material is from the beginning to end. The case study is composed of 3 networks: the 6-bus network from [29] with investment cost data thereof, the 24-bus network from MatPower’s distribution case files with investment cost data from [29], and the 46-bus network from [14] with investment cost data thereof. Appendix B reviews all of the networks we used in detail. Any material that appears simultaneously here and in our PSCC 2014 proceedings paper [2] is indicated whenever a new chapter starts using a special footnote marked with a dagger (†) symbol. We built an index at the end.

Terminology and notation are introduced when appearing for the first time and most of the time can also be deduced from the context. Nevertheless, for convenience, the basic notation and terminology is summarised in Table 1.1.
1. INTRODUCTION

Table 1.1. Summary of the basic notation and terminology.

<table>
<thead>
<tr>
<th>Symbol(s)</th>
<th>Description(s)</th>
<th>Example(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V = { v_1, v_2, ..., v_n } )</td>
<td>A set of elements ( v_1, v_2, ..., v_n )</td>
<td>( W = { 0, \frac{\pi}{2}, \pi, 3 \cdot \frac{\pi}{2}, 2 \cdot \pi } )</td>
</tr>
<tr>
<td>( \mathbb{R}, \mathbb{R}^n )</td>
<td>The set of real numbers, the set of ordered ( n )-tuples of real numbers</td>
<td>( {-1.5, 0, 1, \sqrt{2}, e, 3, \pi } \subset \mathbb{R}, \left( -\frac{\pi}{2}, +\frac{\pi}{2} \right) \in \mathbb{R}^2 )</td>
</tr>
<tr>
<td>( \mathbb{Z}, \mathbb{Z}^n, \mathbb{Z}^+ )</td>
<td>The set of integers, the set of ordered ( n )-tuples of integers, the set of nonnegative integers</td>
<td>( {-1, 0, 1, 2, 3} \subset \mathbb{Z}, (2, 3, 1) \in \mathbb{Z}^+, { i \mid i \in \mathbb{Z} \land i \geq 0 } = \mathbb{Z}^+ )</td>
</tr>
<tr>
<td>( E, I, S, Z, \cdots )</td>
<td>Capital letters usually denote complex numbers</td>
<td>( Z = r + j \cdot x = a \angle \theta )</td>
</tr>
<tr>
<td>( B )</td>
<td>Set of buses</td>
<td>( B = { 1, 2, 3 } )</td>
</tr>
<tr>
<td>( G )</td>
<td>Set of voltage-controlled buses</td>
<td>( G = { 1 } \subset B )</td>
</tr>
<tr>
<td>( C )</td>
<td>Set of circuits</td>
<td>( C = {(1, 2, 1), (1, 2, 2), (1, 2, 3), (2, 3, 1)} = {1-2-1, 1-2-2, 1-2-3, 2-3-1} )</td>
</tr>
<tr>
<td>( C_o )</td>
<td>Set of circuit options</td>
<td>( C_o = {1-2-2, 1-2-3} \subset C )</td>
</tr>
<tr>
<td>( e, E )</td>
<td>Voltage</td>
<td>( e = r \cdot i, E = Z \cdot I )</td>
</tr>
<tr>
<td>( i, I )</td>
<td>Current</td>
<td>( i(t) = i \cdot \cos(\omega \cdot t + \theta_i), I = \tilde{i} \angle \theta_i )</td>
</tr>
<tr>
<td>( p, S )</td>
<td>Power</td>
<td>( p(t) = e(t) \cdot i(t), S = s \angle \theta_{ei} )</td>
</tr>
<tr>
<td>( \overline{p}, q )</td>
<td>Average (or active or real or true) power, reactive (or imaginary) power</td>
<td>( \overline{p} = \frac{1}{T} \int p(t) , dt, S = \overline{p} + j \cdot q )</td>
</tr>
<tr>
<td>( s, \theta )</td>
<td>Apparent power, phase shift</td>
<td>( s = \tilde{i} \cdot \tilde{e}, \overline{p} = \tilde{i} \cdot \tilde{e} \cdot \cos \theta_{ei} )</td>
</tr>
<tr>
<td>( \overline{p}<em>{nm}, \overline{p}</em>{mn} )</td>
<td>Power ( \overline{p} ) at the ( n )-end, power ( \overline{p} ) at the ( m )-end</td>
<td>( \overline{p}<em>{nm} = -\overline{p}</em>{mn}, S_{nm} \neq -S_{mn} )</td>
</tr>
<tr>
<td>( r, x, Z )</td>
<td>Resistance, reactance, impedance</td>
<td>( Z = r + j \cdot x )</td>
</tr>
<tr>
<td>( g, b, Y )</td>
<td>Conductance, susceptance, admittance</td>
<td>( Y = \frac{1}{Z} = g + j \cdot b )</td>
</tr>
</tbody>
</table>

- Upper bound, lower bound | \( \tilde{e}_n \leq \tilde{e}_n \leq \tilde{e}_n \) |
- Root mean square | \( \tilde{e} = \sqrt{\frac{1}{2} \cdot \tilde{e}} \) |
- Best | \( f(P^{N^*}) \leq f(P^N) \forall PN \in F \) |
- Complex conjugate | \( Z^* = r - j \cdot x \) |
Background
The research presented in this thesis can be seen as an interdisciplinary research in the sense that the problem we try to solve lies in the intersection of electrical engineering and computer science. However, even though advanced knowledge is necessary both to formalize the problem and to try to find solutions to it, the solution methods used (as explained in the second part of the thesis) are restricted to the discipline of combinatorial optimisation. As a result, one may argue that (regardless of the appearance) this work is not interdisciplinary after all. In any case, since readers with backgrounds in different disciplines may find some interest in this work, this first part of the thesis tries to unify the background knowledge necessary to clearly understand the problem we investigate in the second part. Basics on the combinatorial optimisation techniques used in the experiments are postponed until the first chapter of the second part for practical purposes.

Most chapters, in both the first and the second parts, build on previous ones.27 The main goal of this first part is to provide a solid context for the work presented in the thesis, and it is organised as follows:

Chapter 2 develops an understanding of basic concepts of electricity. It starts by giving the intuitions behind both voltage and current, and explaining Ohm’s law. Then defines power in an AC power system context, and finalizes by deriving the non-linear power-flow constraints that are enforced in any large scale power transmission network.

Chapter 3 reviews the literature. It aims to locate current knowledge about the problem, and to see what are the researchers’ views. It informs about what has been done before, what seems to be currently taking place, and what are the methods that have been used to conduct research in the area. It starts by locating the problem in time, and, by the end, summarises in a table those studies that, to the best of my knowledge, give the state-of-the-art.

Chapter 4 formalises the problem. It gives an unambiguous description and a formal definition of the problem we try to solve. It starts by defining both power-flow and operational constraints in the context of an AC power system, then provides some minimal examples to illustrate how power-flow constraints make the problem harder to solve, and, by the end, formulates a constrained optimisation model that needs to be solved.

Chapter 5 concludes this part by deriving the research question on top of all of the previous chapters. It starts with a small preliminary study of the methods that could be employed to attempt to find solutions to the problem, and finalises with a statement of the research question. The research question is (of course) what drives our experimental design and gives us way to the next part.

Roughly speaking: Part 1 is more about the theory and the statement of problem, and Part 2 concerns about the practice and the solution to the problem. The last chapter of Part 2 both formulates conclusions and suggests future work.

27For example, the real starting point for this thesis is not the following background chapters but the previous introductory chapter which should thus not be skipped.
CHAPTER 2

Theory basics

When reading about power transmission networks, a basic understanding of electricity becomes necessary since electric power, by a simple definition, is the product of current and voltage. The intuitions behind both current and voltage however, are usually not easy to gain without the opportunity to experiment with electricity. In this chapter, we provide (or reinforce) a basic intuition behind these important two concepts by first defining them in terms of electric charge and then showing how Ohm’s law relates one to another. Furthermore, once this is done, we redefine both voltage and current in the context of AC power systems by integrating a time component on each of them, and then explain how Ohm’s law takes form in the AC context. On top of this, we define electric power and analyse how power is conceived for analysis in the context of AC power systems. The chapter finalises with an explanation of the non-linear non-convex power-flow constrains that we aim to satisfy in the experiments.

1. Voltage, current, and Ohm’s law

Voltage and current are two closely related important concepts that can be defined in terms of electric charge; that is, in terms of a property that appears in nature in either one of two opposite flavours (arbitrarily) named positive and negative. We observe that, by nature, objects with the same charge repel each other while opposite charged objects attract each other. The closer the charged objects are, the stronger the effect. A positive charged object is an object with a deficit of electrons. Conversely, a negative charged object is an object with an accumulation of them.

Electrons and protons are opposite charged subatomic units. Protons reside quietly in the nucleus of an atom, but electrons, with a natural tendency to escape, orbit the nucleus in a cloud (like planets around the sun, so to speak).

---

1George Ohm (Erlangen 1789 - 1854 Munich), physicist and mathematician, developed a complete theory of electricity that was published in his book ‘The Galvanic Circuit Investigated Mathematically’ in 1827. Ohm’s law appeared for the first time in that book, but it is said that it was earlier discovered (independently) by Henry Cavendish (Nice 1731 - 1810 London) who had also developed an unpublished theory of electricity that did not become known until 1879.

2Normally, in atoms (and group of atoms, molecules, and so on) electrons and protons exist in the same quantity (that is to say, the charge is balanced), so from the outside they appear electrically neutral in the sense that they cause no effect in the environment.

3In the context of electric power, charge is measured in Coulombs, where one Coulomb is equivalent to a charge of 6.25 · 10^8 protons (that is, a huge amount of charge).

4We are aware of the more recent quantum mechanical model of the atom, but Bohr’s model, as simplified here, should serve well for our purposes.
electrons do not succeed with their natural tendency to escape (due to their attraction to protons), but they can, and many times do escape from their atoms and travel elsewhere.

When electrons travel, they are actually being transferred from one material to another, and this ability of electrons to flow is what gives the intuition behind current. Electrons ‘like’ to travel so much in order to alleviate imbalances caused by them whenever they can. In this sense, a positive or negative accumulation of charge (that is, a deficit or accumulation of electrons) will cause tension (that is, discomfort) in the environment in such a way that, unless physically restricted, electrons will travel to balance the charge and release the tension. Charged objects that cause this type of tension are said to possess electrical potential energy, or, as commonly in the context of electric power, are said to be at a certain voltage.

This electrical potential energy (i.e., voltage) can be used to perform work since, by definition, energy is the capacity to perform work. For example, the water released from a dam creates energy (that is to say, creates capacity to perform work) that is converted at the hydroelectric power stations (that is, a type of generating unit chosen for this example) into electrical energy that is injected into the system and reaches us, the consumers or loads, through the power transmission and distribution networks. We use this energy to make useful work by using our electrical appliances; for example, to get the benefits of motion, heat, and light.

A little more formally, we can think about the (electric) potential energy held by a charged object as the work\footnote{Work is defined (in physics) as the product of force and distance; that is, the amount of energy transferred by a force acting through a distance.} required to move the object from a point $m$ to another point $n$. Hence, the voltage (that is, more accurately, the electric potential energy) held by a charged object is always relative to a reference location; therefore, it is measured between two points $n$ and $m$. The voltage held by an object is equivalent to the work required to move the object from $m$ to $n$, or, equivalently, the work the object will do by interacting with other objects in its way, to move from $n$ to $m$. For example, a charge moving through a wire from a location $n$ to a more comfortable location $m$ can make work by producing heat (such work is dissipated energy that is normally undesired). Work and energy are equivalent\footnote{Work and energy are measured in the same units; that is, in the context of electric power, in Watt-hours. A 100 Watt light bulb, for instance, consumes $24 \cdot 100$ Watt-hours or 2.4 [kW-h] of energy per day.}, and since electrical potential energy (that is to say, voltage) is measured between two points, a bit of caution is necessary when the reference location is not explicit.

When reading about potential at a single location, usually, the implicit reference location is a (real or abstract) ground location which is neutral in the sense that its negative and positive charges’ influence cancel each other.\footnote{Moreover, a ground location has the ability to remain neutral after absorbing an arbitrary amount of either positive or negative charge.} For example, when saying a transmission line is at 250 kiloVolts, we actually mean 250 [kV] relative to ground, and this is the line voltage. Conversely, when we talk about the voltage...
drop across a line we refer to the difference between the voltages at the two ends of the line, and this is the *line drop* (typically, a few percent only of the line voltage).

The unit for voltage is the Volt ([V]), and one Volt equals one Joule per Coulomb.\(^8\) Since energy is measured in Joules, voltage is defined as *potential energy per unit of charge*. Similarly, the unit for current is the Ampere ([A]), and one Ampere is equivalent to one Coulomb per Second. That is to say, current is defined as the *rate of flow of charge*. Therefore, current can be quantified in terms of the number of electrons moving past a point (say between two ends of a circuit) in a given period of time.

The experimentally established relation between voltage and current is given by *Ohm’s law* as follows:

\[ e \propto i \] (1)

Ohm’s law is a law of nature, and states that the current \( i \) between two points is proportional (\( \propto \)) to the difference of electrical potential energy \( e \) between them. That is to say, given a difference of potential \( e \) between two points, it is possible to create a proportional current \( i \). However, exactly how large the current will be depends on the circuit (or device) that connects these two points.\(^9\)

For any circuit (wire or other device) that offers resistance to the flow of electrons, an electrical *resistance* value is defined for it as the *proportionality constant* between voltage and current. Hence, for a given device with resistance \( r \):

\[ e = r \cdot i, \text{ or, equivalently, } i = \frac{e}{r}. \] (2)

The resistance \( r \) is constant for many materials, and, in the context of power systems, Ohm’s law should describe in sufficient detail what happens in transmission circuits. However, as elaborated in the next section, in the context of a power system where current and voltage are alternating in time with nearly sinusoidal profiles (i.e., in the context of an AC power system), Ohm’s law is expressed as:

\[ E = Z \cdot I \] (3)

In this form of Ohm’s law, both voltage \( E \) and current \( I \) are represented by using complex numbers,\(^{10}\) and they relate each other by a third quantity named *impedance* (denoted by \( Z \)). Although not explicit at this point, impedance not only represents the resistance to the flow of electrons of the connecting device, but also its influence on the relative timing of an alternating current and voltage.

What is an alternating current or voltage? Given two points \( n \) and \( m \), when either \( n \) or \( m \) is at all times at a higher voltage than the other, we say the voltage is unipolar. Unipolar voltage sources produce what we call a *direct current* (DC).

---

\(^8\)One Joule is equivalent to the work required to produce during one Second one Watt of power (that is, one Watt-Second).

\(^9\)Concretely, it will depend on the resistance of a device. Resistance is a good example of what we call a ‘model variable’.

\(^{10}\)Hence the capital letters.
However, polarity can periodically be reversed in time (resembling a sinusoidal waveform) and result in an *alternating voltage*. When a voltage source periodically alternates polarity, it produces what we call an *alternating current* (AC).

Consistently, an *AC power system* is a power system where both current and voltage are alternating in time. In an AC system, both voltage and current are modeled using sinusoidal curves where the time component is described in terms of an angle (and not using standard units of time like Seconds). Representing time with an angle makes sense because the oscillation of both current and voltage is a process that repeats itself (in other words, ‘where the oscillation is’ is more important than ‘when’).

The time given as an angle has no physical dimension; it is expressed in terms of degrees or radians and merely represents a fraction of a whole. The periodic increase and decrease of both voltage and current can then be expressed with either a sine or a cosine waveform. For convenience, we take here the cos function to define both current and voltage as functions of time as follows:

\[
I(t) = I_e \cos(w \cdot t + \theta_i) \quad E(t) = E_e \cos(w \cdot t + \theta_e)
\]

In the previous: \(I_e\) (\(E_e\)) is the *maximum value* of the current (voltage) (that is, the amplitude of the waveform), \(w\) denotes the *frequency* (i.e., the number of complete oscillations per unit of time that, when expressed in terms of radians per second, is called the *angular frequency*), and \(\theta_i\) (or \(\theta_e\)) denotes a *phase shift* (which if zero means that the curve starts at time zero and otherwise means that the curve starts shifted by an angle of \(\theta_i - \theta_e\)).

Note that \(\theta_i\) is not necessarily equal to \(\theta_e\). Therefore, the maximum values do not necessarily coincide in time. This means voltage and current are not necessarily in phase, and, as discussed next, the phase shift between voltage and current has a significant impact in the definition of power in the context of AC systems.

2. Electric power

Electric power is defined as the rate at which electric energy is converted into other forms of energy. Given a current \(i\) flowing across a difference of electric potential energy \(e\), the *electric power* \(p\) is formally defined as:

\[
p = e \cdot i
\]

Since voltage is a measure of energy per unit of charge and current is the flow rate of charge, power becomes a measure of energy per unit of time; that is, a *flow rate of energy* measured in Watts.\(^{11}\) For example, the flow rate of energy dissipated as heat in a device with resistance \(r\) is readily given by \(e \cdot i\). However, when the current is known, since \(e = i \cdot r\) by Ohm’s law, resistive heating\(^{12}\) can alternatively

---

\(^{11}\)Notice that when paying for electric power, it is actually energy (not power) that we pay for.

\(^{12}\)Whenever an electric current flows through a material with non-zero resistance, heat is created.
2. ELECTRIC POWER

(and usefully) be expressed as:

\[ p = i^2 \cdot r \tag{6} \]

This last expression shows that power is more sensitive to changes in the current than to changes in the resistance (because resistive heating depends then on the square of the current). Since resistance may not be able to vary without varying the current, and resistance and current are inversely proportional (that is, if one doubles the other is halved), if we fix the voltage, the net effect of decreasing resistance is to increase power. Resistive heating, which is sometimes intended by design (like in a toaster), is undesired in power transmission lines. The purpose of a transmission line is to transmit energy and not to dissipate it.

Energy converted into heat along a transmission line is in effect lost, and, in extreme cases (that is, when the thermal limit constraints are violated), such heat can melt down the lines. It is useful then to differentiate between two types of power: the power dissipated by a line and the power transmitted by a line. One way to avoid the power dissipated by a line (i.e., to avoid undesirable large resistive heating – which depends on the square of the current) is by increasing voltage. At a high voltage less current is needed to transmit the same amount of power. This explains why high voltages, obtained by using transformers, are normally chosen for power transmission networks.

Ease of voltage conversions is one of the main reasons of why AC electric power systems are preferred over DC systems. While AC systems allow voltage level changes with ease by using transformers, voltage level changes in DC systems is both more sophisticated and more expensive. A drawback, unfortunately, is that, since in an AC system both voltage and current are functions of time, it becomes more complex to handle power in this context. In an AC system, at a given instant, we expect power to be the product of the voltage and the current at the same instant; in other words, we expect that at any instant \( t \):

\[ p(t) = e(t) \cdot i(t) \tag{7} \]

However, since instantaneous power (per se) is not interesting in the context of power systems, both alternating current and voltage are averaged over entire cycles. A meaningful form of average taken then, that offers a good representation of the amount of current (or voltage) supplied, is the root mean square. A root mean square (rms) value is computed by first squaring the entire function (as to eliminate negative values), then taking the mean, and finally taking the square root of the mean. The rms values of both voltage and current, denoted respectively by \( \tilde{e} \) and \( \tilde{i} \), are:

---

13 Transformers allow arbitrary voltage conversions, and voltage levels in transmission networks have grown up to 1000 [kV] (being the most common voltages in the range of 100 to 500 [kV]).

14 In contrast, for example, to the amplitude values \( \hat{i} \) (or \( \hat{e} \)), that would not be good representations, or, to the average values \( \frac{1}{T} \cdot \int_0^T i(t) \, dt \) (or \( \frac{1}{T} \cdot \int_0^T e(t) \, dt \)), that are zero due to the symmetry of the curve.
2. THEORY BASICS

\( \tilde{e} = \frac{\sqrt{2}}{2} \cdot \tilde{e} \) and \( \tilde{i} = \frac{\sqrt{2}}{2} \cdot \tilde{i} \), since:

\[
\tilde{e} = \sqrt{\frac{1}{T}} \int_0^T e^2(t) \, dt = \tilde{e} \cdot \sqrt{\frac{1}{T}} \int_0^T \cos^2(w \cdot t + \theta_e) \, dt
\]

\[
= \tilde{e} \cdot \sqrt{\frac{1}{T}} \int_0^T \frac{1 + \cos(2w \cdot t + 2\theta_e)}{2} \, dt = \tilde{e} \cdot \sqrt{\frac{1}{T}} \left[ \frac{t}{2} \right]_0^T
\]

\[
= \tilde{e} \cdot \frac{\sqrt{2}}{2} \cdot \tilde{e}, \text{ and similarly for } \tilde{i}.
\]

For example, in a standard 120 Volts outlet, the 120 value is the rms \( \tilde{e} \) value, and the amplitude \( \tilde{e} \) value is actually 170 [V].

With the rms values at hand (and a third significant value introduced shortly), it is fortunately easy to express the \textit{average power} \( p \) in an AC power system. From the instantaneous power definition:

\[
p(t) = i(t) \cdot e(t) = \tilde{i} \cdot \tilde{e} \cdot \cos(w \cdot t + \theta_i) \cdot \cos(w \cdot t + \theta_e)
\]

\[
= \tilde{i} \cdot \tilde{e} \cdot \frac{1}{2} \left( 2 \cdot \cos \left( \frac{(2w \cdot t + \theta_e + \theta_i) - (\theta_e - \theta_i)}{2} \right) \right.

\[
\cos \left( \frac{(2w \cdot t + \theta_e + \theta_i) + (\theta_e - \theta_i)}{2} \right)
\]

\[
= \frac{1}{2} \cdot \tilde{i} \cdot \tilde{e} \cdot (\cos(2w \cdot t + \theta_e + \theta_i) + \cos(\theta_e - \theta_i)).
\]

We take then the average; that is to say, we average power over an entire cycle:

\[
\overline{p} = \frac{1}{T} \cdot \int_0^T p(t) \, dt
\]

\[
= \frac{1}{2} \cdot \tilde{i} \cdot \tilde{e} \cdot \left( \int_0^T \cos ((2w \cdot t + (\theta_e + \theta_i)) \, dt + \int_0^T \cos ((\theta_e - \theta_i)) \, dt \right)
\]

\[
= \frac{1}{2} \cdot \tilde{i} \cdot \tilde{e} \cdot \left( \cos (\theta_e - \theta_i) \cdot \int_0^T dt \right) = \frac{\sqrt{2}}{2} \cdot \sqrt{2} \cdot \tilde{i} \cdot \tilde{e} \cdot \cos (\theta_e - \theta_i) \cdot T
\]

\[
= \left( \frac{\sqrt{2}}{2} \right) \cdot \tilde{i} \cdot \frac{\sqrt{2}}{2} \cdot \tilde{e} \cdot \cos (\theta_e - \theta_i) = \tilde{i} \cdot \tilde{e} \cdot \cos \theta_{ei}
\]

In the resulting expression, \( \theta_{ei} \) is the \textit{phase shift} between voltage and current defined as \( \theta_e - \theta_i \). The value for the \textit{average power} \( \overline{p} \) given by this expression in the context of AC power systems is also called: \textit{real power, active power, or true power}. The value of \( \cos \theta_{ei} \) is called the \textit{power factor}, and it determines the amount of power that is actually transmitted (or consumed by the load) with respect to the quantity \( \tilde{i} \cdot \tilde{e} \) which, denoted by \( s \), is called the \textit{apparent power}.

\[15\] The rms value for the AC voltage turns out to be the equivalent DC voltage with the capacity to perform the same amount of work.
Note that even though instantaneous power can be flowing in both directions, the average power is always positive, and in this sense, we say that active power flows in one direction only. The average power is the power that actually does work, and in this sense, we say that it is the real power. We would like real power to be as high as possible, ideally \( p = s \), which happens when voltage and current are in phase; i.e., when \( \theta_{el} = 0 \). However, many times voltage and current are not in phase, and since in a sense not all of the power is delivered to the consumer, we say that not all of the power is real because part of it only oscillates back and forth.\(^{17}\)

Phase difference can arise, e.g., due to the reactance of the transmission line.

In the same way we say that resistance is a property of a material (or device) that allows it resist the flow of current, we also say that reactance is a property of a device (or material) that influences the relative timing (that is, the phase shift) of an alternating voltage and current. Reactance comes in two flavours: inductive which does not like to see current changes, and capacitive which does not like to see voltage changes. The effect of either capacitive or inductive reactance is a phase shift of the current with respect to the voltage but in opposite directions.

An ideal inductor (i.e., a device with inductive reactance) causes the current to lag by \( \frac{\pi}{2} \) (1/4 cycle) behind the voltage, while an ideal capacitor (i.e., a device with capacitive reactance) causes the current to lead by \( \frac{\pi}{2} \) the voltage. In any case, whenever voltage lags current or vice versa (that is, whenever there is any phase shift) not all apparent power \( \tilde{i} \cdot \tilde{e} \) is active, only \( \tilde{i} \cdot \tilde{e} \cdot \cos \theta_{el} \). From there, reactive power, denoted by \( q \), is defined as \( \tilde{i} \cdot \tilde{e} \cdot \sin \theta_{el} \). We could express all apparent, active, and reactive power in Watts, but in order to be able to identify readily from the units which kind of power we are talking about, we measure active power in Watts, apparent power in Volt-Amperes, denoted by VA, and reactive power in Volt-Amperes-reactive, denoted by VAr (all of the three are equivalent).

To reinforce our intuition about reactive power, let us analyse AC power in the context of Figure 2.1. Note that when current and voltage are in phase, as shown in Figure 2.1(a), they cross zero at the same time and the instantaneous power is always positive. That is, when current and voltage are in phase, we say that power is flowing in one direction only. Therefore, all the power is real in the sense that apparent power and active power are the same at all times (while reactive power is zero). However, when current and voltage are out of phase (that is, when one is shifted in time relative to the other), the instantaneous power is not always positive because sometimes it takes negative values; that is, we say that power is flowing in two directions: besides power flowing in the direction we would like, there is also back and forth movement of power. This power, which does not make work, is not zero, and we refer to it as reactive power. However, a little of caution is necessary.

\(^{16}\)The sinusoidal wave for instantaneous power involves both positive and negative values (see Figure 2.1(c), for instance).

\(^{17}\)In the worst case, \( \theta_{el} = \frac{\pi}{2} \) causes \( p = 0 \) indicating that all power is in a back and forth movement (see Figure 2.1(d)); without making any work and causing power losses since current is anyway flowing through components-with-resistance (such as transmission lines and transformers).
Figure 2.1. Current $i$, voltage $e$, and power $p$ in an AC system (adapted from Kassakian and Schmalensee’s [20] Figure B.2). (a) Ideally $e$ and $i$ are in phase so that all of the apparent power $s$ is active and the reactive power $q$ is 0. (b) If $i$ lags $e$, the reactive power is positive. (c) If $i$ leads $e$, the reactive power is negative. (d) In the worst case, with a $\pi/2$ radians phase shift, all of the apparent power is reactive and the average power $\overline{p}$ is zero.
Even though reactive power can be either positive or negative, unlike instantaneous power, its sign does not indicate direction. In this case, the sign indicates the relative phase shift between current and voltage. A positive reactive power, as shown in Figure 2.1(b) indicates that current lags voltage (due to inductive reactance). Similarly, a negative reactive power as shown in Figure 2.1(c) indicates that current leads voltage (due to capacitive reactance). In the worst case, as shown in Figure 2.1(d), when $\theta_{ei} = \frac{\pi}{2}$, the real power is zero, and all of the apparent power is reactive. In the best case, the power factor $\cos \theta_{ei}$ (that is to say, the ratio of real power to apparent power) is maintained to unity; thus all of the apparent power is active as shown in Figure 2.1(a).

Since impedance of transmission lines are unfortunately comprised of inductive reactance, it is expected that current lags voltage. To compensate for this, elements with capacitive reactance (i.e., capacitors) are connected to the transmission lines. The positive phase shift caused by these capacitors cancels out the negative shift due to the inductive reactance of the transmission line (making voltage and current to be in phase); this process is called line compensation. Moreover, capacitors are sometimes also connected near large inductive loads (that is to say, they are connected to buses) to compensate for their positive reactive power. Equipment that draws negative reactive power is often said to be ‘supplying’ reactive power.

In the light of the phase shift problem on AC systems, we can define the AC power as the following complex number:

$$S = p + j \cdot q$$

where $p$ and $q$ are the active and the reactive power respectively.\(^{18}\) The polar form (magnitude∠phase angle) of the complex power is particularly interesting:

$$S = s\angle \theta_{ei} = \tilde{e} \cdot \tilde{e}(\theta_e - \theta_i)$$

$$= \tilde{e} \angle (\theta_e) \cdot \tilde{i} \angle (-\theta_i)$$

because, given that AC voltage and current are defined as:

$$E = \tilde{e} \angle \theta_e \quad \text{and} \quad I = \tilde{i} \angle \theta_i$$

the complex power becomes:\(^{19}\)

$$S = I^* \cdot E$$

which, in a good degree, resembles our first definition of power. Furthermore, in the complex (imaginary) world, resembling Ohm’s law, voltage and current are related through the impedance:

$$E = I \cdot Z$$

where the impedance $Z$ aims to take into account not only the resistance but the

\(^{18}\)It is in this sense that sometimes we say AC power is the ‘sum’ of real and reactive powers. The active power is the real part of the complex apparent power, and the reactive power the imaginary part thereof.

\(^{19}\)Notice the asterisk (*) denotes the complex conjugate of a complex number.
reactance as well. Solving for $Z$ we get:

$$Z = \frac{e^{\angle \theta_e}}{i \angle \theta_i} = \frac{e^{\angle (\theta_e - \theta_i)}}{i} = \frac{e^{\angle \theta_e}}{i}$$

which means that when voltage and current are in phase, $e = Z \cdot i$ as expected. Therefore, $Z = r$. However, this is usually not the case, and we express $Z$ in rectangular coordinates as:

$$Z = r + j \cdot x$$

where $r$ is the resistance and $x$ is the value we call reactance. Mainly to ease later the expressions of our formulas, we also define the admittance, denoted by $Y$, as the inverse of the impedance:

$$Y = \frac{1}{Z} = g + j \cdot b$$

where the values $g$ and $b$ are called, respectively, conductance and susceptance. Notice that when the phase shift between voltage and current is zero, as one may expect, $g = \frac{1}{r}$. However, this is not usually the case, and in general $g = \frac{r}{|Z|^2}$ and $b = -\frac{x}{|Z|^2}$. Resistance, reactance, and impedance are all measured in Ohms, denoted by Ω. Similarly and conversely, conductance, susceptance, and admittance are all measured in Mhos, denoted by Ω.

### 3. Non-linear AC power-flow constraints

Ohm’s law constrains any power transmission network as per the non-linear AC power-flow constraints. In very succinct notation, such constraints ask that the power at each bus is equal to the product of the voltage $E$ and the conjugate of the current $I$:

$$\forall n \in \mathcal{B} : S_n = E_n \cdot I_n^*$$

Unfortunately, this succinct notation, although nice for displaying the complex non-linear non-convex power-flow constraints, is both easy to misinterpret and not really useful in formulating experimental models.\(^{20}\) Therefore, in this section, we massage this expression from this nice succinct form to a useful expanded form by explaining (in sufficient detail only) the main parts of the process. Although arguably a bit redundant, mainly for reference purposes, we relegate a less explained but more complete derivation to Section 1 of Appendix A.\(^{21}\) The resulting expanded non-linear non-convex power-flow constraints obtained by the end of this section are realistic for power systems of large scale [1]; we aim to satisfy such constraints in our experiments.

---

\(^{20}\)Since algorithms to solve such models when running in a computer are restricted to the worlds of integer and floating-point numbers.

Notice that since each bus can have more than one circuit connecting it to other buses, the succinct expression necessarily needs to consider the sum of the power entering or leaving all possible routes. That is:

\[ \forall n \in B : S_n = E_n \cdot \sum_{n \sim m} I^*_n \sim m \]  

where \( n \sim m \) denotes a circuit in the set of all circuits \( n \sim m \) connected to bus \( n \).

Moreover, as per the circuit abstraction (that works thanks to the unified π-model introduced in the next paragraph), a circuit connecting two buses can take the shape of either a transmission line circuit or a transformer circuit.

The main properties of a circuit that are used shortly for analysis as per the abstraction shown in Figure 2.2 are:

- the voltages at points \( n, m, p, \) and \( q \), that is, \( E_n = \tilde{e}_n \angle \theta_n, E_m, E_p, \) and \( E_q \) respectively;
- the currents \( I_{nm} \) and \( I_{mn} \) as seen at nodes \( n \) and \( m \);
- the impedance of the circuit \( Z_{nm} = r_{nm} + j \cdot x_{nm} \) with components that represent, respectively, resistive losses and leakage reactance;
- the shunt admittances \( Y_{nm}^{sh} \) and \( Y_{mn}^{sh} = \frac{1}{Z_{nm}^{sh}} = g_{nm}^{sh} + j \cdot b_{nm}^{sh} \); and
- the tap ratio for transformers at both sides of the circuit, by definition, \( T_{nm} = a_{nm} \angle \phi_{nm} \) and \( T_{mn} \).

Although not illustrated in Figure 2.2, shunt elements exist not only in the circuits; they are sometimes installed at the buses themselves (say at bus \( n \) or at bus \( m \) in Figure 2.2). With this last consideration, the power-flow constraints in what is called their nodal formulation ask that:

\[ \forall n \in B : S_n = E_n \cdot \left( -I^*_{n} + \sum_{n \sim m} I_{n \sim m} \right) \]

Note that in Figure 2.2, besides terminals \( n \) and \( m \), other two points of interest are points \( p \) and \( q \), both points are used to explain what happens when the circuit takes the shape of a transformer. (Please, do not confuse the \( p \) and \( q \) used here with the \( p \) and \( q \) symbols used to denote power and reactive power.)
which reduces to:

(22) \[ \forall n \in \mathcal{B} : S_n = (\bar{c}_n^2 \cdot g_n^{sh} - j \cdot \bar{c}_n^2 \cdot b_n^{sh}) + \sum_{n \sim m} (\mathcal{F}_{n \sim m}^p + j \cdot q_{n \sim m}) \]

so that for a given bus \( n \) (by separating the real and imaginary components of \( S_n \)):

(23) \[ \bar{p}_n^p - \bar{p}_n^q = \bar{c}_n^2 \cdot g_n^{sh} + \sum_n \mathcal{F}_{n \sim m}^p \quad \text{and} \quad q_n^p - q_n^q = -\bar{c}_n^2 \cdot b_n^{sh} + \sum_n q_{n \sim m} \]

Where: \( \mathcal{F}_{nm} \) represents the active power in the circuit at the \( n \) end (respectively for \( \mathcal{F}_{nm}^p \)), and \( (\mathcal{F}_{nm}^p - \mathcal{F}_{nm}^q) \) represents the difference between active power generation and demand (or load) at bus \( n \) (and similarly for reactive power).

To get the expressions for \( \mathcal{F}_{nm}^p \) and \( q_{nm} \) (that is to say, to get the components of \( S_n = E_n \cdot I_{nm} \)), it is necessary to analyse both \( E_n \) and \( I_{nm} \). We analyse first the voltage. When a transformer \( T_{nm} = a_{nm} \angle \varphi_{nm} \) is present, it affects both the phase and magnitude of both \( E_n \) and \( E_n \), yet it keeps the expected ratio:

(24) \[ \frac{E_n}{E_n} = T_{nm} = a_{nm} \cdot e^{j \varphi_{nm}} \quad \text{therefore,} \quad \theta_n = \theta_n + \varphi_{nm} \quad \text{and} \quad \bar{e}_n = a_{nm} \cdot \bar{e}_n. \]

For an in-phase transformer (that is, when \( \varphi_{nm} = 0 \)), the (ideal) voltage magnitude ratio is \( \frac{E_n}{E_n} = a_{nm} \) (as expected) which is also the ratio between \( E_n \) and \( E_n \) (since \( \theta_n = \theta_n \)). Now, we analyse the current. Since there are no power losses in an (ideal) transformer:

(25) \[ E_n \cdot I_{nm}^* + E_n \cdot I_{qp}^* = 0 \]

so that by using \( \frac{E_n}{E_n} = T_{nm} \), we get:

(26) \[ \frac{I_{nm}}{I_{qp}} = -T_{nm}^* \]

Ignoring for a moment the shunt elements, the expressions for the currents are:

(27) \[ I_{nm} = -T_{nm}^* - T_{nm} \cdot E_{qp} \cdot Y_{nm} = (a_{nm}^2 \cdot E_n - T_{nm}^* \cdot T_{mn} \cdot E_m) \cdot Y_{nm} \]

(28) \[ I_{nm} = -T_{nm}^* - T_{nm} \cdot E_{pq} \cdot Y_{nm} = (a_{nm}^2 \cdot E_n - T_{nm}^* \cdot T_{mn} \cdot E_m) \cdot Y_{nm} \]

which are symmetric in the sense that we obtain the correct expressions by exchanging the \( n \) and \( m \)-end symbols. Adding similar terms for the circuit shunt elements, we get the currents for the circuit in Figure 2.2:

(29) \[ I_{nm} = (a_{nm}^2 \cdot E_n - T_{nm}^* \cdot T_{mn} \cdot E_m) \cdot Y_{nm} + a_{nm}^2 \cdot E_n \cdot Y_{nm}^{sh} \]

(30) \[ I_{nm} = (a_{nm}^2 \cdot E_m - T_{nm}^* \cdot T_{mn} \cdot E_n) \cdot Y_{nm} + a_{nm}^2 \cdot E_m \cdot Y_{nm}^{sh} \]

As per the circuit abstraction, for example, to get the currents for a transmission line, we take \( T_{nm} = T_{mn} = 1 \). Similarly, to get the currents for a phase-shifter transformer with tap located on the bus-\( m \) side, we take: \( Y_{nm}^{sh} = Y_{mn}^{sh} = 0, T_{nm} = 1, \) and \( T_{mn} = a_{nm} \cdot e^{-j \varphi_{nm}}. \)

\[ \text{The transformer part of Figure 2.2 is filtered in Figure A.2.} \]
Knowing the current, it is straightforward to derive the power for a circuit \( n-m \) from the expression \( S_{nm} = E_n \left( (a_{nm}^2 \cdot E_n - T_n \cdot T_m \cdot E_m) \cdot Y_{nm} + a_{nm}^2 \cdot E_n \cdot Y_{nm}^2 \right)^2 \).

Once the algebra is done, separating the real and imaginary components, we get the expressions for the remaining terms of the non-linear power-flow constraints. That is, the terms for the active power and the reactive power in a circuit \( n-m \):

\[
(31) \quad \bar{p}_{nm} = (a_{nm} \cdot \bar{e}_n)^2 \cdot g_{nm} - (a_{nm} \cdot \bar{e}_n) \cdot (a_{nm} \cdot \bar{e}_m) \cdot g_{nm} \cdot \cos (\theta_{nm} + \varphi_{nm} - \varphi_{mn}) - (a_{nm} \cdot \bar{e}_n) \cdot (a_{nm} \cdot \bar{e}_m) \cdot b_{nm} \cdot \sin (\theta_{nm} + \varphi_{nm} - \varphi_{mn})
\]

\[
(32) \quad q_{nm} = -(a_{nm} \cdot \bar{e}_n)^2 \cdot (b_{nm} + b_{nm}^e) + (a_{nm} \cdot \bar{e}_n) \cdot (a_{nm} \cdot \bar{e}_m) \cdot b_{nm} \cdot \cos (\theta_{nm} + \varphi_{nm} - \varphi_{mn}) - (a_{nm} \cdot \bar{e}_n) \cdot (a_{nm} \cdot \bar{e}_m) \cdot g_{nm} \cdot \sin (\theta_{nm} + \varphi_{nm} - \varphi_{mn})
\]

As per the circuit abstraction, for example, if the circuit is a transmission line (i.e., if \( a_{nm} = a_{mn} = 1 \) and \( \varphi_{nm} = \varphi_{mn} = 0 \)) we get:

\[
(33) \quad \bar{p}_{nm} = g_{nm} \cdot \bar{e}_n^2 - g_{nm} \cdot \bar{e}_n \cdot \bar{e}_m \cdot \cos \theta_{nm} - b_{nm} \cdot \bar{e}_n \cdot \bar{e}_m \cdot \sin \theta_{nm}
\]

\[
(34) \quad q_{nm} = -(b_{nm} + b_{nm}^e) \cdot \bar{e}_n^2 - g_{nm} \cdot \bar{e}_n \cdot \bar{e}_m \cdot \sin \theta_{nm} + b_{nm} \cdot \bar{e}_n \cdot \bar{e}_m \cdot \cos \theta_{nm}
\]

In a similar fashion, for a transmission line, we get:

\[
(35) \quad \bar{p}_{mn} = g_{nm} \cdot \bar{e}_m^2 - g_{nm} \cdot \bar{e}_n \cdot \bar{e}_m \cdot \cos \theta_{nm} + b_{nm} \cdot \bar{e}_n \cdot \bar{e}_m \cdot \sin \theta_{nm}
\]

\[
(36) \quad q_{mn} = -(b_{nm} + b_{nm}^e) \cdot \bar{e}_m^2 + g_{nm} \cdot \bar{e}_n \cdot \bar{e}_m \cdot \sin \theta_{nm} + b_{nm} \cdot \bar{e}_n \cdot \bar{e}_m \cdot \cos \theta_{nm}
\]

In like manner, expressions for transformers can be obtained, for example, by taking \( a_{nm} = 1, \varphi_{nm} = 0 \) and \( Y_{nm}^sh = Y_{mn}^sh = 0 \) for an in-phase transformer, or by taking \( a_{nm} = 1, \varphi_{mn} = 0 \) and \( Y_{nm}^sh = Y_{mn}^sh = 0 \) for a phase-shifting transformer.

Furthermore, for a transmission line circuit, the active power losses are:

\[
(37) \quad \bar{p}_{nm} + \bar{p}_{mn} = g_{nm} \cdot (\bar{e}_n^2 - 2 \cdot \bar{e}_n \cdot \bar{e}_m \cos \theta_{nm} + \bar{e}_m^2) = g_{nm} \cdot |E_n - E_m|^2
\]

since:

\[
(38) \quad |E_n - E_m|^2 = \Re(E_n - E_m)^2 + \Im(E_n - E_m)^2 = \bar{e}_n^2 - 2 \cdot \bar{e}_n \cdot \bar{e}_m \cos \theta_{nm} + \bar{e}_m^2;
\]

similarly, the reactive power losses are given by:

\[
(39) \quad q_{nm} + q_{mn} = -b_{nm}^h \cdot (\bar{e}_n^2 + \bar{e}_m^2) - b_{nm} \cdot (\bar{e}_n^2 - 2 \cdot \bar{e}_n \cdot \bar{e}_m \cos \theta_{nm} + \bar{e}_m^2) = -b_{nm}^h \cdot (\bar{e}_n^2 + \bar{e}_m^2) - b_{nm} \cdot |E_n - E_m|^2
\]

In the previous:

- \( |E_n - E_m| \) represents the voltage drop across the line,
- \( g_{nm} \cdot |E_n - E_m|^2 \) represents the active power losses,
- \( -b_{nm} \cdot |E_n - E_m|^2 \) represents the reactive power losses, and
- \( -b_{nm}^h \cdot (\bar{e}_n^2 + \bar{e}_m^2) \) represents the reactive power generated by the shunt elements.
Finally, getting back to the power-flow constraints for a circuit, we define for convenience $t^{sh}$ as $\frac{b^{c}_{nm}}{t_{nm}}$ and $a^{nm}$ as $\frac{1}{t_{nm}}$ (since in our current test cases $b^{c}_{nm}$ and $t_{nm}$ are given). Moreover, (since there are no phase-shifting transformers in our study cases and all in-phase transformers have one tap ratio at one side of the circuit), we take $\varphi_{mn} = \varphi_{nm} = 0$ and $a_{mn} = 1$. The general power-flow constraints for a circuit reduce then to:

\[
\begin{align*}
\overline{p}_{nm} &= \tilde{e}_{n}^{2} \cdot \frac{g_{nm} - \tilde{e}_{m} \cdot \tilde{e}_{m}}{t_{nm}} \cdot (g_{nm} \cdot \cos \theta_{nm} + b_{nm} \cdot \sin \theta_{nm}) \\
\overline{q}_{nm} &= -\tilde{e}_{n}^{2} \cdot \frac{b_{nm} + b^{c}_{nm}}{2t_{nm}} \cdot (g_{nm} \cdot \sin \theta_{nm} - b_{nm} \cdot \cos \theta_{nm})
\end{align*}
\]

which are the asymmetric non-linear non-convex power-flow constraints that we should aim to satisfy in the experiments.
CHAPTER 3

Literature review

In this chapter, we provide a review of the related literature with two main goals: to expose current knowledge about the type of problem we aim to solve and to look for any information that sets ground for a potential research question. We make note of the methods that are used to do research in the area (that we may use as well), and aim to provide more background and context for our work (as well as information about its significance). We start by tracing back this type of problem to the first formulation of the AC-OPF problem that includes the power-flow constraints. From there, we give a structured account of what has been done (relevant to the problem at hand) to date. We aim to fit well our problem in the existing scholarship, and, by summarising and discussing previous research, prepare well for formalising the problem in the next chapter and derive the research question after that. We finalise the chapter by summarising the relevant studies in a table.

The AC Optimal Power Flow (AC-OPF) problem, since (arguably) first formulated in 1962 (by Carpentier [8] as cited by Low [22] and others [7]), enjoys today several decades of research with good reason. Some form of this problem is solved for systems planning every year and for day-ahead markets every hour or even every five minutes [7]. Algorithms designed to solve the economic dispatch formulation\(^1\) of the optimisation problem\(^2\) (for instance) are run in computer centres every 5 to 10 minutes to determine the dispatch for the next hour and to send the appropriate signals to all generating units accordingly [20] – to mention one example only.

An efficient algorithm or technique to solve the AC-OPF problem could therefore save billions of dollars [7]. Since no fast and robust solution technique is known today, engineering is widely used in both academy and industry to search for techniques to find acceptable solutions to the optimisation problems. As a result, the existing literature on this problem is very extensive, with many different formulations\(^3\) of the optimisation problem under study. The number of different formulations reach a point where one can doubt the necessity of a common view of the problem formulations or even claim [7] that the research community lacks of a common understanding of the optimisation problem constraints and objectives.

\(^1\)That is, to allocate projected energy demand to committed generating units while minimising the overall production cost. Carpentier’s key contribution in 1962 was actually to include the power-flow constraints into the economic dispatch formulation of that time [7].

\(^2\)The OPF problem has different formulations; hence, we can safely use the term OPF here to also collectively refer to a set of power-flow optimisation problems.

\(^3\)The different formulations vary both in form (for example, different objectives or constraints) and shape (for example, polar or rectangular).
From a more general perspective, the AC-OPF problem is an important application that probably falls among the most important instances of constrained non-linear non-convex optimisation problem currently under research.\textsuperscript{4} However, in a good part of the literature, the AC-OPF problem, which is already known to be computationally expensive is further complicated by introducing discrete (binary or integer) decision variables into the constraint optimisation model in order to decide the status of different components of the network. For example, binary variables are introduced to state whether generating units or circuits are on or off, or integer variables to decide how many component options of a given kind (like circuit options for a determined corridor) are added to the expansion plan. The resulting problem is a large mixed-integer non-linear program which is also non-convex.\textsuperscript{5}

Such additional discrete decision variables allow more complex formulations of the optimisation problem like, for instance, those found in the literature for generating unit commitment\textsuperscript{6} or transmission switching\textsuperscript{7}. Such formulations become problem instances of constrained mixed-integer non-linear non-convex optimisation problems\textsuperscript{8} and are known to be even more difficult to solve. For example, it is commonly understood in practice that generating unit commitment is more complex and time consuming than economic dispatch \textsuperscript{20}. In any case, it is known that non-convexity (which can arise easily due to a bilinear term, for instance, as a product of circuit status and voltage phase angle decision variables) implies that reaching global optima cannot be guaranteed in polynomial time. Even with linearised power-flow constraints, due to the discrete decision variables, the resulting programs become non-linear and furthermore non-convex.

Our research about working out an algorithm to find low cost expansion plans, subject to both the set of non-linear power-flow constraints (an expression of Ohm’s law) and the set of operational constraints (like the thermal limits on the circuits), fits in this last category – entering into the realm of discrete optimisation, since using discrete decision variables in the model becomes essential in order to choose expansion options to create an expansion plan.\textsuperscript{9} Furthermore, the natural constrained optimisation model for our problem resembles those explored in the literature in the context of the transmission network expansion planning formulation.\textsuperscript{10} Among the ultimate goals of problems of this kind, in the literature, is to include on its

\textsuperscript{4}Efficient global optimisation algorithms for general non-convex non-linear optimisation problems is a big area of research.

\textsuperscript{5}Convexity is always highly desired when aiming to find the global optimal solution.

\textsuperscript{6}Generating unit commitment roughly asks to decide when generating units should be online while minimising the associated costs.

\textsuperscript{7}Transmission switching roughly asks to minimise generation cost by changing the network topology; that is to say, by switching lines.

\textsuperscript{8}Efficient global optimisation algorithms for general non-convex non-linear mixed-integer optimisation problems is also a big area of research (see, for instance, http://www.math.hu-berlin.de/~stefan/B19/ accessed on October 2014).

\textsuperscript{9}One cannot add a fractional part of a circuit (or any other kind of) option to a power transmission network.

\textsuperscript{10}A traditional AC-OPF formulation is contrasted with the transmission network expansion planning formulation in previous work [41].
formulation security constraints (as those defined by the $n-1$ reliability criterion). Even though different transmission network expansion planning formulations have been studied for more than a decade now, most of them are based on a form of the linearised DC power-flow constraints.

In a few words, the transmission network expansion planning problem (TNEP for short) asks to choose, at the minimum cost, an expansion plan (i.e., a subset of a set of network component options – generally only circuit options) such that once the power-flow constraints are enforced by nature in the expanded network, under steady state, a projected energy demand is satisfied together with the operational contraints. Classical constrained optimisation models for TNEP include: transportation models, hybrid models, DC models, and disjunctive models [31] which aim to deal with the non-convexity of the optimisation problem and to be compatible with the branch-and-bound technique used to solve mixed-integer programs.

From another perspective, the TNEP problem is described in the literature as a network design problem for which it is known that classical optimisation models that rely on DC power-flow constraints can lead to poor designs because their solutions are likely to require a reinforcement step to satisfy all of the network constraints under the AC model. The difficulty of TNEP can trivially be justified with the non-existence of efficient global optimisation algorithms to deal with the inherent resulting large non-linear non-convex programming problems [29] that are most of the time considered unsolvable.

One step further, some publications talk about multi-stage expansion planning, others include the role of competitive markets; some aim for joint planning, and others show other variations that most of the time can be seen as extensions of the TNEP problem described so far. In multi-stage expansion planning, for example, expansions are planned not only for a single year but for several years [42]. In a sense, this form of the problem not only asks to decide ‘which’ options should be added to the network but also ‘when’. Similarly, considering the role of competitive markets translates (in practice) to including electric power market constraints into the TNEP optimisation problem model, which can be important in some settings.\footnote{11For instance, there have been delays in expanding Australian networks due to such constraints [10].}

Finally, aiming a joint planning of the expansion entails a potential necessity to formulate the problem as a multi-objective optimisation problem where the objective has not only terms to minimize the investment cost of installing circuit options, but also to optimise something else, as, for example, to minimise the cost of installation of reactive power sources.\footnote{12It is not uncommon to read that the objective of TNEP is to maximise the ‘social welfare’ which most of the times translates into minimising both the expansion investment cost and any other operational costs.}

Here we focus on static planning where expansion options are said to be chosen either for one year only or for a single planning horizon (that is to say, here we do not care about the ‘when’). We give emphasis on those models that take into account
an accurate form of the AC power-flow constraints (in other words, those that take into account other forms than LDC approximations – that are more accurate), and pay more attention to those that aim to satisfy the $n - 1$ reliability criterion (that is, to those that consider the reliability criterion that generates our scenarios set). Nonetheless, note that static TNEP formulations in the literature can normally be extended (although not explicitly done) to take into account multiple scenarios as the ones defined by the $n - 1$ reliability criterion (of course with an increased computational burden). Likewise, the extension to a multi-objective formulation of the problem can be straightforward, but an extra difficulty arises in the optimisation since one needs to find the genuine optimal solution with respect to more than one objective; that is to say, one needs to find the Pareto optimal solutions.

The two main solution approaches proposed in the literature to the kind of TNEP formulations we aim to investigate are heuristics and mixed-integer programming. Heuristic approaches offer no guarantee on the quality of a solution (that is, there is no way to tell whether a solution is optimal or near optimal or far away from the optimal, and they are sometimes heavily criticised because of this). Conversely, mixed-integer programming (MIP) approaches usually give a measure on the optimality gap (indeed, mathematical programming methods can usually guarantee optimality albeit after potentially very long time; that is, a time easily larger than a lifetime). Heuristic approaches include, for example, constructive heuristics and genetic algorithms, and mixed-integer programming state-of-the-art solvers are likely to apply, for example, sophisticated branching strategies and advanced cutting plane algorithms. Under a more general perspective, mixed-integer programming solving can be seen as a systematic search technique. Conversely, approaches like genetic algorithms can be seen as local search techniques.

Local search techniques, heuristics with meta-heuristics, are attractive, not only because they may reduce the computational effort to find good solutions for large instances of the problem, but also because they can be applied to find solutions which are feasible under the AC non-linear power-flow constraints.\textsuperscript{13} For example, by decoupling the discrete decision variables from the model and solving on top of an interior-point method applied to the AC model – as done by Rider, Garcia, and Romero\textsuperscript{29}. Mixed-integer programs are also attractive because state-of-the-art solvers that can be used for some problems of this type are of industrial-strength. However, even though research on non-convex mixed-integer non-linear programming\textsuperscript{14} may seem promising, or even more promising the active research on mixed-integer second-order cone programming (which may soon give implementations of algorithms of similar performance to those of mixed-integer linear programming), only mixed-integer linear programming (MILP) can be considered mature technology\textsuperscript{37}.

\textsuperscript{13}That is, to find solutions that otherwise may not be possible to find for not small networks.
\textsuperscript{14}Current available mixed-integer non-linear programming solvers are discussed elsewhere\textsuperscript{41}. 
As a result, most mixed-integer programming models and heuristics in the literature rely on a linearised version of the power-flow constraints. However, note that several of these heuristic approaches are nonetheless not really sensitive to the underlying power-flow model. Actually, when it is acceptable to ignore reactive power, under normal operation conditions and with some adjustment for circuit losses, the LDC (linearised DC or sometimes just DC) power-flow constraints (derived in Appendix A for reference, which may adopt many forms as discussed elsewhere [34]) produce a reasonably accurate approximation of the AC power-flow constraints for active power [9] only. Such constraints, due to both their efficiency and simplicity, are widely used in both mixed-integer programming and heuristic approaches.

When efficiency is desired, the LDC power-flow constraints are the approximation of choice in the literature for optimisation. For example, Bienstock and Mattia [4] aimed to find minimum cost expansion plans to protect power transmission networks from potential large-scale cascading blackouts. Notably, they present computations on a (large) test case with 600 buses and 827 circuits by using the LDC power-flow constraints. They used binary decision variables to indicate an increase in circuit capacity (a modelling device that may not work well when moving to the AC model), and gave models that could be solved with MIP technology. They also developed algorithms that are not tied with the LDC power-flow constraints, yet it has not been experimented with the AC power-flow constraints.

MIPs are known to be effective up to medium-size networks; for instance, the computational effort increases with the number of discrete decision variables. Consequently, a heuristic approach may be more effective on large networks. Since, generally speaking, they often succeed in practice where more conventional algorithms are unable to produce (even suboptimal) solutions for large problems. Gallego, Monticelli, and Romero [12] proposed, in this line, an extended genetic algorithm, without assuming any particular linear or convex set of power-flow constraints, and solved the problem using the LDC power-flow constraints for a network with 89 buses. Later, Silva et al. [33] proposed using the LDC power-flow constraints as well, a specialised genetic algorithm including constraints for the \( n-1 \) sub-networks, and solved the problem for a 46-bus network.

The reliability problem, that is, to include constraints to satisfy the \( n-1 \) reliability criterion in the transmission network expansion planning formulation, is usually solved in two phases when approached in the literature. In the first phase, the problem is solved without the reliability criterion, and in the second phase, to avoid high computational effort, the expanded network is secured by adding circuits in some kind of iterative approach until the reliability constraints are satisfied. Such approach is not only likely to produce suboptimal expansion plans that are highly influenced by the plan obtained in the first phase, but also (unless captured somehow in the \( n-1 \) analysis) produces an expansion plan that will make the

---

15That in turn are likely to be a function of the size of the network.
16The \( n-1 \) criterion is discussed in more detail somewhere else [40].
network satisfy the $n - 1$ reliability criterion with respect to the plan found in the first phase. Hence, the resulting expanded network is likely to not be reliable itself. For instance, Silva et al. [33] observed that, when solving the same problem in one single phase (first) with and (then) without a reliability criterion, some circuits of the expansion plan found without security constraints did not belong to the expansion plan found with security constraints.

Furthermore, when the $n - 1$ reliability criterion is considered in the literature, many times the full $n - 1$ sub-networks are not taken into account; most likely, to avoid the computational burden implied. In many studies, the set of scenarios is not strictly given by the failure of any single one circuit, but reduced in one way or another. For example, sometimes only a subset of circuits which may cause credible single line outages is taken [39]; sometimes it is argued that only a limited subset of circuits would cause serious overloads [40], or, for instance, that only the subset of line circuits needs to be considered (and transformer circuits can be disregarded).

Another study with large networks that takes into account the $n - 1$ reliability criterion was done by Hedman et al. [15]. They solved the problem using the LDC power-flow constraints for 96 and 118-bus networks, and showed that the cost can be reduced and the constraints satisfied in a network by incorporating switching into the dispatch. However, solutions under the AC power-flow constraints including the $n - 1$ reliability criterion are almost non-existent. Indeed, the literature becomes scarce already when considering any optimisation problem under the AC power-flow constraints. Accurate approximations or relaxations to the AC power-flow constraints are by themselves scarce; LDC approximations here persisted over time. A debate exists about to which point the widely used LDC approximation is a good one (a good approximation is an accurate one and a good relaxation a tight one). For example, it has been observed [40], on a 6-bus network, that expansion plans obtained with a LDC approximation can be problematic in the AC network (due to the approximation inaccuracies [3], in more than one sense).

At any rate, there are evident efforts in the literature to move from LDC problem solving towards problem solving using the AC power-flow constraints. For example, Xu et al. [39], on a 14-bus network, applied a branch-and-bound algorithm using the relaxed non-linear program (of the original mixed-integer non-linear program) to solve a multiobjective formulation and ensured $n - 1$ feasibility by using an AC-OPF program. More recently, Zhang et al. [40] solved the expansion planning problem using a linearised approximation of the AC model. For a 118 bus network, they ensured $n - 1$ feasibility with an iterative approach and claimed such approach is applicable to large systems planning. To the best of our knowledge,

---

17. An approximation or relaxation of a model should be easier to solve than the original model. Furthermore, a relaxation should not cut solutions and bound the minimum below (for a minimisation problem), or the maximum above (for maximisation).

18. Note that, despite of its bad name whose origin is discussed elsewhere [34], the ‘LDC or DC approximation’ is an approximation of the AC model; observe it involves phase angles. However, please, be aware that when we talk about an ‘approximation or relaxation to the AC model’ we refer to an approximation or relaxation, other than DC, that is more accurate.
the most accurate linear approximation of the AC power-flow constraints are the LPAC [9] constraints that we use later on.

Furthermore, some of the authors who work using the linearised DC power-flow constraints make efforts to create solutions agnostic to the underlying power-flow model (for example, Bienstock and Mattia [4] or Gallego, Monticelli, and Romero [12]). Even without doing so, some authors state that approaching the problem under the AC power-flow constraints should be done in the future. However, this is rarely done by the same authors (or others) probably due to the intrinsic complexity of such non-convex non-linear constraints, or simply because the intractability associated to the AC version of the problem gives little hope. Moreover, when this is done with approximations or relaxations to the AC power-flow constraints (by using conic or linear relaxations, for instance) the solutions obtained are rarely validated under the AC model which means that the expanded network may not be AC feasible (without a reinforcement) after all. It has been observed [41], for example, that optimal expansion plans obtained by using relaxed or approximated models can turn out to be infeasible in AC power-flow studies so that a validation step is really necessary to ensure feasibility (not to mention that the DC models are more likely to fail a feasibility test to the original AC problem [37]). Even though a later reinforcement step may be possible to do in order to cope with AC power-flow infeasibility or operational constraints violations, transmission networks are evolving towards a state where such reinforcement may no longer be possible [3]. Solutions should ideally be AC feasible in the sense that they should cause no violations in the operational constraints once the AC power-flow constraints are enforced by nature.

Solving the problems with the actual AC power-flow constraints, or with an accurate simplification of them that leads to AC feasible solutions, is of significance not only because it would allow for the avoidance of a reinforcement step necessary due to the approximation or relaxation, but also because it would reflect more accurately the value of the objective function on the optimisation problems, consider both reactive power and voltage magnitudes (in order, e.g., to avoid solving a separate problem to allocate reactive power), and incorporate circuit losses in a natural way.19 These are all known problems associated with DC models [29] which can occur in many different settings. For example, expansion plans found with the AC model have been observed [41] to be more expensive and to be composed of more expansion options than expansion plans found with relaxed models.20 Indeed, underestimation of the real expansion cost is sometimes reported in the literature when solving with simplified versions of the non-linear power-flow constraints [3].

In the near future, the transmission network expansion planning problem under the AC model may demand high quality solutions more than ever; studies show that

---

19 There are studies in the literature that aim to include circuit losses when the LDC power-flow constraints are used, for example, by trying to incorporate circuit losses in the known disjunctive model.

20 Likely due to voltage and reactive power issues.
large scale networks with renewable energy sources may cause common assumptions used in typical approximations to not hold anymore [3]. Reactive power, e.g., may become more and more relevant, for example, with the growing incorporation of energy sources like wind and solar [3]. However, transmission network expansion planning under the AC power-flow constraints is still extremely challenging today, therefore new approaches to solve the problem more accurately are needed.

The first solid steps in this direction were taken by Rider, Garcia, and Romero [29] who gave a constructive heuristic algorithm (CHA) to solve the expansion planning problem under the AC power-flow constraints; by using CHA on 6, 24, and 46-bus networks, they showed that it is viable to think about solution techniques based on the (full) AC model. CHA relies on an interior-point method to solve the underlying non-linear programming problems of their solution technique; their solutions are not optimal, although of good quality since the algorithm finishes when an AC solution is found. CHA can consider allocation of reactive power sources, and the authors did so in a preliminary way, and, furthermore, advised to consider the cost of the installation of reactive power sources in the objective function. Although shown on a 6-bus network only, later Mahmoudabadi et al. [23] showed, by using the CHA method, that the cost of the expansion plan can be reduced by allocating reactive power sources. It was also later observed [3] that when (inductive or capacitive) compensation is allowed in all buses, the expansion plans are considerably less expensive; in fact, investment on reactive power sources is known to be less costly than investment on circuit expansion, so a joint optimisation of reactive power sources and circuit reinforcement can both result in lower investment costs and help to satisfy the voltage operational constraints [18]. Note CHA has also been embedded in the branch-and-bound framework [30].

When using heuristics to approach a solution to the problem under the AC model, it is common to take the information about how loaded circuits are in order to choose circuit expansion options. For example, CHA uses a sensitivity index to do so in practice; a circuit option is chosen for the expansion plan when it has a similar with the greatest apparent power-flow load in the non-linear (relaxed) problem solution. In the same spirit, later, an indicator was used [23], obtained after a power-flow analysis is done, to identify weak buses for reactive power allocation. Similarly, the solution technique in Xu’s [39] uses a performance index that indicates the overloading in the system. One step further, contrasting in some degree the indexes just described, some more elaborated heuristics to choose circuit expansion options are discussed by Bent et al. [3]; for example, ‘flow diversion’ and ‘alternate path around’.

In the systematic search track, Taylor and Hover proposed linear [37] and conic [36] relaxations for AC expansion planning, resulting in mixed integer and conic programs respectively. They showed, using 6 and 46-bus networks, improvements over existing LDC, and, using a 6-bus network, improvements over non-linear models, respectively. Zhang et al. [41] applied existing mixed-integer-nonlinear
solvers to find strictly AC feasible expansion plans; they compared solutions, using 6 and 24-bus networks, with those of the non-linear relaxation obtained by relaxing the discrete variables (a potentially easier problem) and a linearised reformulation. Later [40], they solved a mixed-integer program for expansion planning, by linearising the AC power flow constraints, and reinforced the network with additional circuits using an iterative approach to ensure the resulting network complies with the $n - 1$ reliability criterion. Jabr [18], alternatively, proposed a conic relaxation to define a mixed-integer conic program that takes into account both voltage magnitudes and reactive power. The model can be solved with a branch-and-cut algorithm, for his case study, in principle, in polynomial time. He showed the efficiency of his method in 6, 24 and 46-bus networks, validated his solutions under the AC model, and gave model extensions to incorporate VAr and multistage planning.

All of these results for systematic search, to a certain extent, put in evidence the viability of mixed-integer programming techniques to solve the problem at hand; from these formulations, the mixed-integer non-linear program is the most challenging one, and the mixed-integer linear program is the most attractive due to the existence of mature implementations of optimisation algorithms for MILP (that have improved in tandem with computers computational capabilities). Another advantage of using linear models is that they can be used in the same fashion as DC models [37], inheriting in a sense, the benefit of extensive previous research.

In the local search track, Bent, Toole, and Berscheid proposed [3] a discrepancy-bounded local search algorithm (DBLS), guided by both feasibility and optimality, that aims to incorporate non-linear power-flow models for transmission network expansion planning. DBLS shares similarities with branch-and-bound search, but limits its exploration, and, at the same time, generalizes existing constructive heuristics from the literature. DBLS was tested in a particular 900-bus real network under the AC power-flow model. In a similar line, Gallego et al. [11] proposed a genetic algorithm (another form of local search) to solve the TNEP problem, by integrating the AC model using an interior-point method. Later, Rahmani et al. [27] proposed a genetic algorithm as well, which further incorporated reactive power planning based on AC-OPF. None of these incorporated the $n - 1$ reliability criterion.

Genetic algorithms as well as the DBLS can be seen as optimisation-simulation approaches, where a simulator with a potential blackbox implementation (which may contain non-linear models) is queried about the quality of a solution while the optimisation algorithm continuously tries to find solutions of higher and higher quality. Such approaches may be appropriate for solving the problem at hand because, with an astronomical number of possible solutions to be explored, there is currently no known way to evaluate a possible solution both quickly and accurately. In other words, a traditional complete neighbourhood evaluation seems out of scope. Furthermore, since TNEP has local optima, a solution method for the TNEP may have a large change to be trapped in a local optimum [11]; therefore, meta-heuristics
are likely to be needed as an escape mechanism (as for example, tabu search or simulated annealing).

Unfortunately, genetic algorithms and optimisation-simulation approaches in general are known to require extensive parameter tuning. Main parameters for a genetic algorithm include, for instance, the size of the population, the algorithm to initialise such population, the selection operation for choosing individuals as operands for the crossover and mutation operations (used to evolve the population), the mutation and crossover operations themselves, and the criterion to terminate the algorithm. For the problem at hand, usually a large population is preferred, and ways used for choosing individuals for the initial population include selection based on system information and human knowledge [39], randomness, and methods which rely on simpler models or more conventional suboptimal algorithmic techniques [12]. The codification of the individuals of the population, usually as binary strings, plays also an important role for the efficiency and semantics of the mutation and crossover operations (as discussed elsewhere [12]).

Some guidelines exist in the literature, nevertheless, to choose these parameters for the TNEP problem. For example, it has been reported [12] that mutation rates need to be larger than for other problems, and that a two point crossover operation behaves well with a high crossover probability. Despite this, in a subsequent approach, a one point crossover operation was used anyway when considering an AC model [11]. In any case, there is no agreement on a reliable way to tune such parameters which are, moreover, not always independent as the values chosen for some can have impact on others and they are, furthermore, sometimes changed dynamically as the population evolves. A variation of the traditional genetic algorithm,21 called CBGA in the literature, alleged to be more competitive in large problems, has also been applied to transmission network expansion planning using both the DC [33] and the AC [11] models; the latter via an interior-point method, in 6 and 24-bus networks, by allocating reactive power sources and incorporating techniques to improve both infeasibility and optimality of individuals.

In summary, there is clear interest in the community to move from DC-based problem solving strategies to more accurate ones that would ideally guarantee both optimality (that is, the lowest cost expansion plan) and feasibility (that is, absolutely no violations on the operational constraints) under the enforced AC power-flow constraints. Practical solving strategies, with a good balance between efficiency and accuracy, could bridge this existent gap between AC and DC solving. For the problem at hand (that we have casted here into a TNEP problem with the \( n - 1 \) reliability criterion) Table 3.1 summarises what has been done in the literature in order to find solutions by using some accurate form of the AC model. These papers, together with the LPAC (linear) power-flow constraints [9] (derived in Appendix A for reference) motivate our work; we aim to contribute to this line of research. Note

---

21The origin of genetic algorithms can be traced back to the 1970s when they became popular due to the work by John Henry Holland.
Table 3.1. TNep papers using some accurate alternative form of the AC model.

| Paper          | AC form         | Solution method   | Networks | \(a\)  \(b\)  \(c\) |
|---------------|-----------------|-------------------|----------|--------|--------|
| Xu et al. [39] | non-linear      | branch-and-bound  | 14-bus   | yes    | no     |
| Rider et al. [29] | non-linear     | constructive heur. | 6, 24, 46-bus | no    | yes    |
| Gallego et al. [11] | non-linear    | GA                | 6, 24-bus | no    | yes    |
| Taylor et al. [37] | linear-relax.  | MILP              | 6, 46-bus | no    | no     |
| Taylor et al. [36] | semidefinite, conic-relax. | MISOCO               | 6-bus   | no    | no     |
| Zhang et al. [41] | non-linear      | MINLP NLP MILP    | 6, 24-bus | no    | yes    |
| Bent et al. [3]    | non-linear      | DBLS              | 24, 900-bus | no    | no     |
| Zhang et al. [40] | linear-relax.   | MILP              | 6, 118-bus | yes   | no     |
| Jabr [18]          | conic-relax.    | MISOCP            | 6, 24, 46-bus | no    | yes    |
| Bent et al. [2]    | linear-approx.  | MILP              | 6, 24, 46-bus | no    | yes    |

\(a\) Does it consider the \(n-1\) sub-networks?
\(b\) Does it validate solutions under the AC model? ('-' means that it is not necessary)
\(c\) Does it consider a form of reactive power allocation?

that in Table 3.1, the networks used in the different papers, although of the same size, are not necessarily the same (we review our study cases in Appendix B). From Table 3.1, we can argue that by using an accurate linear approximation of the AC power-flow constraints (read, the LPAC linear power-flow constraints), MILPs and genetic algorithms are good choices to experiment with.\(^{22}\)

\(^{22}\)DBLS [3] has been tested in our study networks by Bent without much success.
CHAPTER 4

Description of the problem

We pose the problem in this chapter, and we do it formally so that there is no ambiguity about what the solutions we try to find (with the experiments) are. We state what the variables, the constraints, and the objective of the optimisation problem are. We count the number of constraints and variables and illustrate how the power-flow constraints make the network-flow optimisation problem harder to solve. We conclude with a constrained optimisation model as the unambiguous description of the problem at hand.

At a large scale, a power transmission network \( \mathcal{PN} \) is modeled by a set \( \mathcal{B} \) of buses\(^1\) together with a set \( \mathcal{C} \) of circuits.\(^2\,3\) At each bus we can have either injection of power (produced by generating units), demand of power (consumed by loads), or both, or none. Each circuit can be either a transmission line or a transformer. Given \( \mathcal{PN} \), we expect that, in all of the scenarios given by a set \( S \), the generation of power meets the demand by having power (or, more accurately, energy) flowing through the circuits while both power-flow and operational constraints are satisfied.

The main constraints on \( \mathcal{PN} \) are the complex asymmetric non-linear power-flow constraints, an expression of Ohm’s law (as explained in Section 3 of Chapter 2), which ask that at each network described by a scenario \( s \in S \), at each bus \( n \):

\[
\begin{align*}
\bar{p}_n &= \bar{e}_n^2 \cdot g_n + \sum_{n \sim m} \bar{p}_{n \sim m} \quad \text{and} \quad q_n = -\bar{e}_n^2 \cdot b_n^h + \sum_{n \sim m} q_{n \sim m}
\end{align*}
\]

where \( n \sim m \) denotes a circuit in the set of all circuits \( k \) connected to bus \( n \), and:

\[
\begin{align*}
\bar{p}_{nm} &= \bar{e}_n^2 \cdot \frac{g_{nm}}{t_{nm}} \cdot \bar{e}_m \cdot \cos \theta_{nm} + b_{nm} \cdot \sin \theta_{nm} \\
q_{nm} &= -\bar{e}_n^2 \cdot \frac{b_{nm}}{t_{nm}} \cdot (b_{nm} + \frac{b_{nm}^c}{2}) \cdot \bar{e}_m \cdot \cos \theta_{nm} - b_{nm} \cdot \sin \theta_{nm}
\end{align*}
\]

\(^1\)The term ‘bus’ is one of those that have different meanings in different contexts; within computer science, in fact, this is an overloaded noun. The term ‘bus’ – a clipped form of the Latin ‘omnibus’ which means ‘(carriage) for all’ – is, in the context of power transmission networks, literally a bar of metal (or busbar) to which all incoming and outgoing circuits are connected.

\(^2\)The general constrained optimisation model shown in this chapter in Model 4.1 appears using a different notation and with slight variations in Model 1 of our PSCC 2014 proceedings paper \[2\]; that is, Model 1 in the mentioned paper makes explicit the inputs and decision variables, does not show the generating unit operational constraints, and the power-flow constraints shown there disregard the lines charging, bus shunts, and transformers model variables which are introduced as extensions in Subsection II.A thereof.

\(^3\)With the term ‘circuit’ we mean either a transmission line or a transformer (see Figure 2.2).

The pair \( (\mathcal{B}, \mathcal{C}) \) constitutes a multigraph, still \( \mathcal{PN} \) can be thought as a graph.
These non-linear power-flow constraints (realistic for large scale power systems, and sometimes called physical constraints) are non-convex, 2 per bus, each with as many terms as 3 times the number of circuits connected to the bus. The size of the problem (i.e., of a network) is usually measured as a function of the number of buses; the larger the number of buses, the larger the number of power-flow constraints (and the larger the number of variables). The power-flow constraints are not the only constraints in the system. However, we comment first on the constraint system defined so far by these complex constraints and introduce the rest later on.

The decision variables are (roughly speaking) the voltages \( \tilde{e}_n \angle \theta_n \) (that is, 2 per bus), and solving for these, even in the system described so far, is difficult because this is a system of non-linear equations. Numerical methods exist, however, as the Newton-Raphson method, iterative in nature and not guaranteed to converge, that can be used to solve the system, and, upon success, evaluate a solution (that is, to check if all of the operational constraints are satisfied for a given expansion plan). A solution to these equations, for a given network, is what we call an AC power-flow solution; we aim that all of the operational constraints (described shortly) are satisfied in an AC power-flow solution for each network of each scenario. In other words, we aim that the network operates normally under all of the scenarios.

Since Ohm’s law holds by nature, the non-linear power-flow in the network is enforced by nature, unfortunately, regardless of other existing operational constraints (like the thermal limit constraints on the circuits). By themselves, these power-flow constraints make the problem depart from a standard flow problem. The power-flow constraints make the decision about which circuit options to choose for an expansion plan hard to make. We illustrate part of this phenomena in Figure 4.1.

Consider the 7-bus network in Figure 4.1(a). For the sake of simplicity, in this example and in what follows, we take the widely used LDC approximation of the AC power flow-constrains (that is to say, we take the linear DC power-flow constraints – derived in Section 2 of Appendix A for reference) which reduces the corresponding AC power-flow equality constraints to:

\[
\begin{align*}
\bar{p}_{nm} &= \tilde{e}_n^2 \cdot g_{nm} - \frac{\tilde{e}_n \cdot \tilde{e}_m}{t_{nn}} \cdot (g_{nm} \cdot \cos \theta_{nm} + b_{nn} \cdot \sin \theta_{mn}) \\
q_{mn} &= -\tilde{e}_m^2 \cdot \left( b_{nm} + \bar{b}_{nm} \right) + \tilde{e}_m \cdot \tilde{e}_n \cdot (b_{nm} \cdot \cos \theta_{mn} - g_{nn} \cdot \sin \theta_{mn}).
\end{align*}
\]

Moreover, to simplify further the arithmetic, we make \( b_{nm} = 1 \) for all circuits such that, in the analysis, \( \bar{p}_{nm} = -\theta_{nm} \).

\[4\text{To have a good understanding of the notation and the meaning of the variables, recall Section 2 in Chapter 3, or eventually visit Appendix A. See also Table 1.1 for reference.}

\[5\text{And impractical to use without a computer; especially, when the network is not small.}

\[6\text{For our current study cases (described in Appendix B), given a solution configuration, such methods typically converge so we use them to evaluate a solution.} \]
4. DESCRIPTION OF THE PROBLEM

(a) The feasible regular network flow in this 7-bus network is infeasible under the LDC power-flow constraints (network figure adapted from Bienstock and Mattia’s [4] Figure 1).

(b) The network flow (see left) becomes infeasible under the LDC power-flow constraints upon adding one high capacity circuit (see right).

**Figure 4.1.** Constrained network flow example. (a) In a standard flow problem, generation can meet the demand with the shown topology but the flow becomes infeasible under the LDC power-flow constraints [4]. (b) Adding a circuit can make the flow of power infeasible (here under the LDC power-flow constraints) in counterintuitive ways. The labels on the circuits can be seen as power-flow capacities.

In the 7-bus network in Figure 4.1(a), the total capacity of the generating units connected to buses 2 and 3 (that is, of all generators) is $75 + 35 = 110$ which equals the total demand of power required at buses 1, 6 and 7 (that is, at all loads). The labels on the circuits are their thermal limits which can be seen as power-flow capacity constraints. In this network, it is clear that to meet the demand of bus 1, necessarily $p_{21} = p_{31} = 5$; thus $p_{24} \leq 70$ and $p_{35} \leq 30$. In a standard flow problem, demand of buses 6 and 7 can be met by making $p_{46} = 50, p_{45} = 20$, and $p_{57} = 50$.

However, if the network is constrained by the power-flow constraints, the load of bus 7 cannot be met because: $p_{24} > p_{35} \Rightarrow \theta_2 - \theta_4 < \theta_3 - \theta_5$ and $p_{21} = p_{31} \Rightarrow \theta_2 = \theta_3$; then $-\theta_4 < -\theta_5$, which means $p_{45} = -(\theta_4 - \theta_5) < 0$. Therefore $p_{57} < 30 < 50$, so that generation cannot meet the demand after all.

Moreover, the power-flow constraints make it harder to try to deduce where we should expand the network; the effect of adding a circuit can easily trick our intuition. Consider, for instance, the 3-bus network in Figure 4.1(b); without circuit 2-3 (Figure 4.1(b)-left), it is clear that buses 2 and 3 can be served. However, upon adding a circuit 2-3 (with a high thermal limit as in Figure 4.1(b)-right) the load of bus 3 can no longer be served since $p_{12} > p_{13} \Rightarrow \theta_2 > \theta_3$. Therefore $p_{32} = -(\theta_3 - \theta_2) > 0$, so that the constraints cannot be satisfied.

This kind of counterintuitive behaviour which may seem sometimes chaotic, although not random, makes it difficult to predict what is going to happen either when one or more circuits are lost or whenever the network is expanded by adding one or more circuits. At any rate, we would like the network to be able to operate normally at all times; for a given scenario, the network operates normally if and only if the power-flow constraints are satisfied together with the following operational...
4. DESCRIPTION OF THE PROBLEM

constraints:

\[ \sqrt{p_{nm}^2 + q_{nm}^2} \leq \hat{s}_{nm}, \quad \text{for any circuit } n-m \text{ (thermal limits)} \]  

\[ \tilde{e}_n \leq \hat{e}_n \leq \check{e}_n, \quad \text{for any bus } n \text{ (voltage bounds)} \]

\[ \hat{p}_{nk}^g \leq p_{nk}^g \leq \check{p}_{nk}^g, \quad \text{for any generating unit } n-k \text{ where } \sum_k p_{nk}^g = p_{n}^g \quad \text{(generating unit capacities)} \]

Note that the circuit operational constraints that exist in order to model the thermal limits of the circuits (i.e., the thermal limit constraints), although convex, are non-linear. The operational constraints on the buses (i.e., the voltage bounds constraints) asking that the voltage root mean square values are kept between certain values, are likely to be set by design. In contrast, the thermal limit constraints are set by nature in the sense that we can only indirectly influence them. When the configuration of generation of power (that is, the dispatch) is part of the optimisation process, the generating-unit operational constraints ask that all of the units respect their energy generation capabilities (note also that a bus \( n \) can have many generating units \( n-k \) connecting to it).

The constraint system described so far (for one or many networks in \( S \)) plus the objective of minimising a function like generation cost or power-loss is what is called the AC optimal power flow (or AC-OPF) problem in the literature. Our expansion problem, however, adds an extra layer of complexity to this already difficult problem by introducing binary decision variables, \( \psi_{nm} \), to denote the status of a circuit option \( n-m \) (that is, 1-on or 0-off; fixed to 1-on if the circuit is not an option). The presence of discrete decision variables in a non-linear systems is known to make the optimisation problem much harder. For example, existing numerical methods become problematic; their convergence becomes a real problem.

Consider a network with \( m \) corridors, if we allow one circuit option per corridor, we need \( m \) binary decision variables; hence, there are \( 2^m \) possible expansion plans.\(^7\) Therefore, the population of candidate expansion plans can grow exponentially. For a given candidate expansion plan, we (try to) determine the dispatch (and even the voltages for the voltage controlled buses). All of these, the expansion plan, the \( \tilde{p}-\tilde{e} \) values for voltage controlled buses, and the \( \tilde{p}-q \) values for the rest of the buses (setting \( \theta = 0 \) in the slack bus), make up a (network) configuration. All the variables involved in a configuration are decision variables; the rest of the variables in the optimisation model are model variables corresponding to properties of buses or circuits. Once we have a candidate solution configuration, we configure the network accordingly and let the network run. If the network operates normally in all of the scenarios, then the configuration is a solution configuration. We look for the optimal solution; that is, the solution configuration that involves the minimum cost expansion plan.

\(^7\)Sometimes it is difficult to grasp exponential growth: put a single grain of rice on the first chess square of a chess board and double it on every consequent one.
4. DESCRIPTION OF THE PROBLEM

Minimize:
\[ \sum_{n,m} \gamma_{nm} \cdot \psi_{nm} \quad \text{That is, the expansion investment cost.} \]

Subject to:
For all scenario \( s \) in \( S \), \( \triangleright \) Note we do not include \( 's' \) in the notation
(\( \triangleright \) except when strictly necessary).
\( \triangleright \) Note \( \psi_{nm} \) are shared among all \( s \) in \( S \).

For all bus \( n \), \( \triangleright \) The power-flow constraints:
\[ p_n^d - p_n^l = \frac{e_n^2}{t_{nm}} \cdot g_{nm} - \frac{e_n \cdot e_m}{t_{nm}} \cdot (g_{nm} \cdot \cos \theta_{nm} + b_{nm} \cdot \sin \theta_{nm}) \]
\[ q_n^d - q_n^l = -e_n^2 \cdot b_{nm} + \frac{e_n \cdot e_m}{t_{nm}} \cdot (g_{nm} \cdot \sin \theta_{nm} - b_{nm} \cdot \cos \theta_{nm}) \]

where:
\[ \tilde{e}_n = \frac{e_n}{t_{nm}} \]
\[ \tilde{g}_{nm} = \frac{e_n \cdot e_m}{t_{nm}} \cdot (g_{nm} \cdot \cos \theta_{nm} + b_{nm} \cdot \sin \theta_{nm}) \]
\[ \tilde{q}_{nm} = \frac{e_n \cdot e_m}{t_{nm}} \cdot (g_{nm} \cdot \sin \theta_{nm} - b_{nm} \cdot \cos \theta_{nm}) \]

For all circuit \( n-m \) in \( n-m \), \( \triangleright \) The circuit operational constraints:
\[ \sqrt{p_{nm}^2 + q_{nm}^2} \leq \tilde{s}_{nm} \cdot \psi_{nm} \]
\[ \sqrt{p_{nm}^2 + q_{nm}^2} \leq s_{nm} \cdot \psi_{nm} \]

For all bus \( n \), \( \triangleright \) The bus operational constraints:
\[ \tilde{e}_n \leq e_n \leq \tilde{e}_n \]

For all generating unit \( n-k \), \( \triangleright \) The gen-unit operational constraints:
\[ \tilde{p}_{nk} \leq p_{nk} \leq \tilde{p}_{nk} \]

where:
\[ \sum_k p_{nk} = p_n^g \]

Model 4.1. AC-TNEP as the general constrained optimisation model for the problem we try to solve. The solutions that we look for should satisfy this model.

Model 4.1 shows the general constrained optimisation model for our problem at hand (in the model, \( \gamma_{nm} \) denotes the cost of installing a circuit option in corridor \( n-m \)). We change the model during experimentation; for example, some variables may become constants or the objective may change to study some property, or we may change the cardinality of the scenarios set. Therefore, in the next part of the thesis, when designing the experiments, we explain or make the constrained optimisation models more concrete before presenting the results in subsequent chapters. In any case, the solutions that we look for should satisfy Model 4.1; when a solution configuration satisfies this model, we say that the solution is ‘AC feasible’.

For practical purposes, we refer to this model as the ‘transmission network expansion planning’ problem when there is only one scenario in \( S \). That is, when a projected future energy demand is given and the network needs to be expanded so
that the increased generation capacity can meet the projected demand. Likewise, for practical purposes, we refer to this model as the ‘$n - 1$’ or ‘reliability’ problem when $S$ includes, in addition, the sub-networks defined by the $n - 1$ reliability criterion. In the next chapter, we derive the research question we address with the experiments and then move on to the second part of the thesis.
CHAPTER 5

Derivation of the research question

We investigate the problem described and formalised in the previous chapter (concretely, the constrained optimisation model shown in Model 4.1). Solutions to such non-convex non-linear program (aside the binary decision variables for a moment) usually rely on iterative (numerical) methods whose convergence is studied most of the times using the mathematics of real analysis.\(^1\) The integrality constraints on the binary decision variables in the problem,\(^2\) used to model the status of the circuit options, make the problem harder to solve (that is, existing solution methods are problematic to use; for example, they do not converge).

As per the literature review, linear approximations and convex relaxations of the non-linear power-flow constraints exist that can lead to mixed-integer linear or conic programs for which efficient implementations of global optimisation algorithms exist. Furthermore, such linear approximations and convex relaxations of the non-linear power-flow constraints may also be used in heuristic approaches.\(^3\) By using these techniques, relying on an accurate approximation or relaxation to the AC power-flow constraints, we can aspire to find AC feasible solutions.

At any rate, different approximations or relaxations to the non-linear power-flow constraints (naturally) take different approaches and make some assumptions on their own, so that the quality of the solutions produced when relying on these are likely to be different. We (of course) look for both quality and efficiency, but we need to satisfy the optimisation model shown in Model 4.1, ultimately for a set of scenarios defined by the \(n - 1\) reliability criterion.

The notable advantage of using linear programs or mixed-integer linear programs is the existence of mature algorithms that benefit from decades of research, in both academy and industry, for which industrial-strength implementations exist (that are regarded as very efficient). Therefore, linear constraints are the ones currently promising a better performance for potential program solving techniques; nevertheless, second-order cone constraints are also attractive because implementations of algorithms of similar performance are thought to be at the doorstep.

\(^1\)The 3-bus study on TNEP solution methods, presented and discussed in this chapter, is presented and discussed in a similar fashion in Section IV of our PSCC 2014 proceedings paper [2]. Tables 5.1, 5.2, and 5.3 and Figure 5.1 here are, respectively Tables I, II, III and Figure 1 there.

\(^2\)Real analysis is a branch of mathematical analysis dealing with the analysis of both real numbers and real functions.

\(^3\)That is to say, the discrete nature of the domains of the binary variables (usually not specified as explicit constraints in the model).

\(^3\)Heuristics that, by the way, are likely to decouple the discrete decision variables from the model.
Figure 5.1. A 3-bus network with its data. The network has a 100 [MVA] base and the voltages are also given in p.u. terms. Two variants of this network are considered for the preliminary study, one with a low thermal limit of 5 [MVA] in circuit 2-3 and the other with tight voltage magnitude bounds of ±0.01 [V p.u.].

Table 5.1. AC power-flow solution for the 3-bus network of Figure 5.1.

We conducted a preliminary study, in the context of the transmission network expansion planning (TNEP) problem, to discuss potential solution approaches and illustrate their tradeoffs. We compared between:

- our best known AC feasible solution AC*-TNEP,\(^4\) with
- a destructive heuristic solution HAC-TNEP\(^5\) (that contrasts the typically proposed constructive heuristics as explained in Chapter 7); with
- two MILP formulations, the DC-TNEP (as per the known disjunctive model [31]) and the LPAC-TNEP (as explained in Chapter 7), that rely, respectively, in the popular linearised DC power-flow constraints (frequently visited and revisited [34]) and in the more accurate LPAC power-flow (linear) constraints [9]; with
- two MISOCP formulations, the SOCP-TNEP and the SOCP*-TNEP, that rely, respectively, in Jabr’s second order cone relaxation of the AC

\(^4\)Obtained by using the Bonmin (Basic Open-source Non-linear Mixed-INteger programming) COIN-OR Project, see http://projects.coin-or.org/Bonmin – Accessed on October 2014.

\(^5\)One of the interests in HAC-TNEP is its ability to find primal solutions for our use that can serve as starting points for large scale MIPs which are known to have trouble finding one.
5. DERIVATION OF THE RESEARCH QUESTION

Table 5.2. Results for the 3-bus network with a low thermal limit of 5 [MVA] in circuit 2-3.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Total cost [$]</th>
<th>Circuit options</th>
<th>Thermal violation [MVA]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC*-TNEP</td>
<td>27</td>
<td>0 15 12</td>
<td>0</td>
</tr>
<tr>
<td>HAC-TNEP</td>
<td>72</td>
<td>0 72 0</td>
<td>0</td>
</tr>
<tr>
<td>SOCP*-TNEP</td>
<td>19</td>
<td>0 8 11</td>
<td>31.3</td>
</tr>
<tr>
<td>SOCP-TNEP</td>
<td>1</td>
<td>0 0 1</td>
<td>184</td>
</tr>
<tr>
<td>LPAC-TNEP</td>
<td>27</td>
<td>0 15 12</td>
<td>0</td>
</tr>
<tr>
<td>DC-TNEP</td>
<td>27</td>
<td>0 14 13</td>
<td>0</td>
</tr>
</tbody>
</table>

We bench, in this preliminary study, all of these six methods using two variations of the 3-bus network shown in Figure 5.1. However, before doing so, we let the 3-bus network run and show the AC power-flow solution in Table 5.1. As this table reflects, the network is designed so that the power flows on the paths 1-3 and 1-2-3 are roughly the same. In the first variation of the 3-bus network that we study, circuit 2-3 has a low thermal limit of 5 [MVA], and, in the second variation, we make the voltage magnitude bounds tight as ±0.01 [V].

In the first variation of the 3-bus network (i.e., the one with one low thermal limit circuit), AC*-TNEP finds an expansion plan with 15 + 12 = 27 additional circuits (that is, a 27 \$ expansion plan for one dollar cost circuit options). On the one hand, since the approximations capture effectively congestion caused by cycles in the network, the results for both LPAC-TNEP and DC-TNEP are expansion plans that enable 27 circuit options as well. On the other hand, since the SOCP model does not capture cycle-based circuit congestion in the network, SOCP-TNEP underestimates the expansion needed and chooses an expansion plan with only one circuit option; expanding the network with this single circuit option leads to huge thermal limit violations. SOCP*-TNEP does it better by choosing an expansion plan with 19 circuit options, but there are still significant thermal limit constraint violations in the circuits. The results of this study are shown in Table 5.2.

In the second variation of the 3-bus network (i.e., the one with tight voltage bounds), AC*-TNEP finds an expansion plan of 5 additional circuits. Since DC-TNEP has no notion of voltage magnitudes (due to the underlying approximation), it suggests no expansion. SOCP-TNEP, perhaps a little better, chooses an expansion plan that enables only 2 circuit options that leads to violations in the bus operational constraints. All of the rest, same as AC*-TNEP, choose expansion plans of 5 circuit options. These results are shown in Table 5.3.

6In the domain of power systems analysis it is common to see a per-unit system where quantities are actually divided by a base unit. We made this explicit in Figure 5.1 (e.g., by giving the MVA base or specifying [V p.u.] as units), but from now on we take this for granted; we do not make it explicit anymore (e.g., the cost may be expressed in ‘dollars’, but the unit may be \( \approx 10^6 \)).
Table 5.3. Results for the 3-bus network with tight voltage magnitude bounds tight of ±0.01 [V].

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Total cost [$$]</th>
<th>Circuit options</th>
<th>Voltage violation [V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC$^2$-TNEP</td>
<td>5</td>
<td>2-3</td>
<td>0</td>
</tr>
<tr>
<td>HAC-TNEP</td>
<td>5</td>
<td>3-2</td>
<td>0</td>
</tr>
<tr>
<td>SOCP$^2$-TNEP</td>
<td>5</td>
<td>2-3</td>
<td>0</td>
</tr>
<tr>
<td>SOCP-TNEP</td>
<td>2</td>
<td>1-1</td>
<td>0.0256</td>
</tr>
<tr>
<td>LPAC-TNEP</td>
<td>5</td>
<td>2-3</td>
<td>0</td>
</tr>
<tr>
<td>DC-TNEP</td>
<td>0</td>
<td>0-0</td>
<td>0.1567</td>
</tr>
</tbody>
</table>

The goal of the 3-bus network study was to illustrate the general shortcomings of the alternative approaches. In summary, for the 3-bus network of Figure 5.1, the LPAC-TNEP formulation provides a good tradeoff between accuracy and efficiency (in this case, LPAC-TNEP finds the best-known solutions for the two variants of the 3-bus network). Since we later observed similar trends for other networks we use, it is reasonable to choose, among the different alternative options to AC-TNEP, LPAC-TNEP as a potential good approximation of the challenging mixed-integer non-linear program (see also the correlations plots in Figure 5.2 mentioned below).

In addition to this 3-bus network preliminary study, we conducted other preliminary experiments; for example, to clarify our instruments and some other aspects of our methodology (explained in Chapter 7). We mention two of them here before stating the research question and moving on to the second part of the thesis. The first has to do with my implementation of Coffrin and Van Hentenryck’s LPAC power-flow constraints [9] (the cold-start flavour of them to be more specific), and the second has to do with the effectiveness of the MILP solver we use.

The implementation of the LPAC power-flow constraints (derived in Section 2 of Appendix A for reference), on its own, is a non-trivial task; therefore, it seems mandatory to check its resulting behaviour before writing a LPAC-TNEP program. For this reason, we benchmarked the accuracy of our implementation of the LPAC power-flow constraints with the PyPower implementation of the non-linear AC power-flow constraints (that is based on Newton’s method)\(^8\) using the MatPower OPF benchmarks. The best correlation plots were obtained for the 24-bus network and they are shown in Figure 5.2 (for reference, another set of plots for the 14 and 57-bus benchmarks are given in Figures A.3 and A.4, respectively).\(^9\) To the best of our knowledge, the LPAC are the most accurate linear power-flow constraints.

---

\(^7\)PyPower is a port of MatPower to the Python programming language.

\(^8\)In short, Newton’s algorithm attempts to construct a sequence $x_n$ from an initial guess $x_0$ that converges towards $x^*$ such that $f'(x^*) = 0$. The basic idea is that of giving to the computer a starting point $x_0$, take the tangent of the curve at $x = x_0$, and take the point $x_1$ where this tangent meets the x-axis. Then repeat. It turns out that $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$.

\(^9\)Note that in the titles of these figures, for a circuit n-m, with ‘power to’ we mean the power at the m-end, and with ‘power from’ we mean the power at the n-end.
Our solver of choice is Gurobi; thus we use it to find solutions to both linear and mixed-integer linear programs. To the best of our knowledge, Gurobi represents the state-of-the-art. One collateral advantage of using it is the existence of its Python library named GurobiPy that can be used in tandem with PyPower.\textsuperscript{10} As part of the preliminary experiments, we implemented DC-TNEP using Gurobi (as per the disjunctive model that is used many times in the literature, shown here in Chapter 7) and compared against published results that use the same model on the 24-bus IEEE-RTS-based test cases named g1, g2, g3, and g4, proposed originally by Fang and Hill\textsuperscript{10} (from where we can see the effectiveness of the MILP solver we were released by October 2014 while the latest version of PyPower was released by December 2010 (as checked in October 2014). As the creator of both libraries explains ‘PyPower is a more direct translation of MatPower than Pylon. If nothing else, it should be faster’.

\textsuperscript{10}Note there exists a Python library called Pylon as well. However, the latest version of PyPower was released by October 2014 while the latest version of Pylon was released by December 2010 (as checked in October 2014). As the creator of both libraries explains ‘PyPower is a more direct translation of MatPower than Pylon. If nothing else, it should be faster’.
use). These four test cases, as variations of the 24-bus IEEE reliability test system, have the same load profile (in other words, they have the same demand), but they have different configurations in the amount of power each generating unit supplies (that is to say, they have a different dispatch). The network data is explained by Fang and Hill [10], but we got, for a better unambiguous comparison, text files with the network data from Bent et al. [3] (who experimented with them recently).

Best known and best found objective values are compared in Table 5.4. In all cases, the MIP for DC-TNEP runs to optimality in less than one minute and reproduces or improves the results reported in the literature. The solution for $g_1$ is the same solution reported by Bent et al. [3], and the solution for $g_3$ is the same solution found by the constructive heuristic of Romero et al. [32]. The objective values (that is to say, the cost) of the solutions to test cases $g_2$ and $g_4$ improved the costs of reported best known solutions [3].

Getting back to the AC model and to our problem at hand (that is, to try to find solutions to Model 4.1 by using state-of-the art resources within our means), the specific research question that we investigate in this MPhil thesis project is whether the LPAC power-flow constraints, as a linear approximation of the AC constraints, is a good mechanism to find low cost AC feasible expansion plans in the context of transmission network expansion planning, if possible satisfying the $n - 1$ reliability criterion (or even an arbitrary set of scenarios). The methods and experiments we use to address this question follow in the next part.

**Table 5.4.** Best known/best found objective values comparison for DC-TNEP solutions for the $g_1$, $g_2$, $g_3$, and $g_4$ test cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>Best known</th>
<th>Best found</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_1$</td>
<td>390 [3]</td>
<td>390</td>
</tr>
<tr>
<td>$g_2$</td>
<td>424 [3]</td>
<td>392</td>
</tr>
<tr>
<td>$g_3$</td>
<td>218 [32]</td>
<td>218</td>
</tr>
<tr>
<td>$g_4$</td>
<td>354 [3]</td>
<td>342</td>
</tr>
</tbody>
</table>
Experiments
The first part of the thesis focused on both explaining and formalising a hard combinatorial optimisation problem that falls in the power systems domain and lies in the intersection of electrical engineering, computing science, and mathematics. The second part of the thesis focuses on explaining and analysing computational evaluations of algorithms designed to solve the problem formulated in the first part. All of the work done relevant to answer the research question is presented in this second part.

We propose both models and algorithms (this is the more innovative task) that may be used to solve the problem at hand, evaluate all of our solutions to see how effective these are in finding AC feasible configurations (using classic test case networks), then make further experiments with VAr compensation and power market considerations (in a more observational task), and, at last, adopt an evolutionary approach targeting AC solutions to the problem that satisfy the proper $n - 1$ reliability criterion.

Since an optimisation problem (per se) is a set of instances of an optimisation problem, with no exception, the optimisation problems we try to solve here are sets of instances of optimisation problems as well. In this sense, the study presented herein is a case-study that takes 6, 24, and 46-bus network instances (as common cases seen in the TNEP literature), and expects that the results found for these networks may apply to other instances of similar characteristics.

This second part is organised as follows:

Chapter 6 develops an understanding of the underlying optimisation algorithms that we use in the experiments (more precisely, it explains in a general context both branch-and-bound and genetic algorithms) and makes the ideas of the optimisation algorithms more concrete by solving (for illustrative purposes only) a small instance of the knapsack problem.

Chapter 7 designs the experiments of our work (that is to say, it explains both data and methods that we use to investigate the research question). In this chapter, we describe the experiments we run, the algorithms we use to run the experiments, and make concrete the measures and criteria that we use to evaluate the results of the experiments.

Chapter 8 presents the results of the experiments and discusses them.

Chapter 9 concludes the thesis, draws together the contributions made to the topic, and gives directions for future work.
CHAPTER 6

Practice basics

When solving models for which no polynomial time algorithmic solution exists, we usually resort to the idea of *intelligently* exploring the configurations space until a solution configuration is found. In an optimisation problem, we usually can do this until a solution configuration is found to be the best among a good set of neighbouring configurations in the space (that is, a *local search*); in the best case, we can do this until a solution configuration is found to be the best among all configurations (that is, a *complete search*). In any case, when saying that we resort to the idea of exploring ‘intelligently’ the configurations space, we say it in the sense that we need to do it much better than enumerating and testing all of the possible configurations (in the space or even in a neighbourhood) of a constrained optimisation model (and then choose the best). Unfortunately, an additional difficulty in our problem at hand is that configurations are not easy to test for feasibility (that is, to check if they are solutions) or to generate at random some configurations that are solutions.

In this chapter, we give the basics of the techniques we use and the intuitions underlying the two algorithms chosen for the experiments described in the next chapter. Concretely, these two algorithms are branch-and-bound search (that is, an intelligent complete search technique) and evolutionary search (that is, an intelligent local search technique). Although proprietary software, as our solver of choice *Gurobi*, is almost by definition not open source (i.e., it is not really possible to have a look at what they are using under the hood), MILP problems (i.e., MILPs) are generally solved using an LP-based branch-and-bound algorithm (in consistence with *Gurobi*’s documentation about optimising MIP problems).\footnote{\textit{Gurobi}’s documentation also says that they use many other ingredients as well (not mentioned here) in order to improve the capabilities of their algorithm. For more information, visit \url{http://www.gurobi.com/resources/getting-started/mip-basics}. Accessed on October 2014.} We also use *Gurobi* to solve LPs (i.e., LP problems), and the algorithm known to be the most efficient and reliable to solve LPs is the simplex\footnote{The simplex algorithm was originally proposed by George Dantzig (Portland 1914 - 2005 Stanford) in 1947, and it is well studied and widely used today.} (that we do not believe necessary to explain here). Nevertheless, other alternatives to the simplex method include the barrier or the interior-point methods, and in practice different LP problems may find one or another method more effective. As for the evolutionary search, we rely in our own implementation of a genetic algorithm (in other words, the implementation is verbatim to what is schematized here and described in the next chapter) that we use in the spirit of integrating simulation with optimisation.
In the first section of this chapter, we explain the branch-and-bound framework (a systematic search technique of the configurations space). We explain first the algorithm in a general context, and then we make the idea more concrete by solving a small MILP problem. Similarly, in the second section, we explain the genetic algorithm we use in a simulation-optimisation framework (a more stochastic local search of the configurations space). We explain first the algorithm in a general context, and then we make the idea more concrete by solving a small problem.

In both sections, the small problem we use (for illustrative proposes only) is a small instance of the knapsack problem (chosen for simplicity). The story tells:

In a risky expedition, you luckily found a set of \( n \) objects of weights \( w_1, w_2, \ldots, w_n \) and values \( v_1, v_2, \ldots, v_n \) respectively that you wish to collect and take home. Unfortunately, you have only one knapsack that can hold at most a weight of \( K \). Determine the subset of objects of maximum value that you can take home in your knapsack.

The knapsack constrained optimisation problem is similar to our problem at hand in the sense that both of them are intractable (that is, there is no efficient way to search for optimal solutions to them); in other words, there is no efficient way to check exhaustively all possible combinations of objects (as the time grows exponentially with the number of objects). However, by using the techniques described in this chapter, we can sometimes find the optimal (that is, the best) solution configuration, or a solution configuration that is good enough or close to the optimal (without guarantees). After concluding this chapter, in the next, we design the experiments that we run for this thesis using the two techniques presented next.

1. Branch-and-bound algorithm

The branch-and-bound algorithm relies in two main operations: branching and bounding. Given a set of configurations (usually depicted as a node), the branching operation partitions the set in two (or any finite number of) smaller mutually exclusive sets (read sub-problems), and the bounding operation gives a lower bound on the value of the objective function of any configuration in a set of configurations. These two operations are pretty much what the algorithm needs, so there is really no requirement about linearity neither in the objective function nor in the constraints of the model. Therefore, the branch-and-bound algorithm is applicable to most combinatorial optimisation problems. However, if the bounding operation relies on solving the LP relaxation of a MILP problem, then linearity requirements apply.

---

3The knapsack problem, posed by Tobias Dantzig (Present-day Latvia 1884 - 1956 Los Angeles), has been around for more than a century now. Tobias Dantzig was the father of George Dantzig (who developed the simplex algorithm to solve LP problems).

4Of course that other techniques exist as well that can be used for the same purpose (for example, dynamic programming), but we (of course) give the basics only for those two techniques that we use in the experiments. The optimisation toolbox is a big one.

5Otherwise such particular case problem would not be a MILP problem after all. Recall a MILP is like a LP but with some of their decision variables constrained to take integer values.
The intuition behind the branch-and-bound algorithm is that of splitting an optimisation problem that we cannot solve into smaller (or simpler) subproblems so that the best solution among the solutions of all of these subproblems is the best solution to the original problem. The key is that for any of these subproblems, if we still cannot solve it, we can split it again but before doing so we can quickly determine what is the best cost we could get among all of its subproblems (as described so far, that would be among all of the corresponding sub-subproblems, but we can have subproblems at any depth). The trick is that if we have already found a solution to any solved subproblem of better cost than the best cost we could get in an unsolved subproblem, then we can avoid wasting time in solving such unsolved subproblem (since it will give a suboptimal solution anyway).

A little more concrete, given a combinatorial optimisation problem that we cannot solve, we split (read branch) it to smaller subproblems. For each of these subproblems \( s \), we get a relaxed solution of cost \( c \) (read bound) that is a lower bound (for a minimisation problem) on the cost of the solutions of the subproblems of \( s \) – the tighter the relaxation for bounding the better the algorithm behaves. Sometimes it is not possible to find a relaxed solution to \( s \), and that usually means that there is no solution to the subproblem. Sometimes the solution to the relaxed \( s \) is actually a solution to \( s \), thus simultaneously a solution to the original problem (i.e., the relaxed solution is a solution to the non-relaxed \( s \)). In this last case, since \( s \) is solved, there is no need to split \( s \) any more. While running the algorithm, we keep track of the best of such solutions found so far together with its cost \( c^* \); if for any subproblem \( s \) the cost \( c \) of its relaxed solution is worse than \( c^* \), then we ignore \( s \) and do not waste time trying to solve it – for some large problems, the time saved with this strategy is significant to the point that the algorithm can terminate its job. When the branch-and-bound algorithm terminates, we know the optimal solution has cost \( c^* \) – a mathematical view of this process is that of constructing a proof that a solution \( \text{Solution}^* \) of cost \( c^* \) is optimal (to the original problem) by splitting the solutions space once and again until no more splitting is possible.

Consider an optimisation problem \((\mathcal{F}, f)\); that is to say, consider an optimisation problem defined by both the set \( \mathcal{F} \) of all solution configurations (i.e., the set of all feasible configurations) and an objective function \( f \) (by which we determine which of all of the solution configurations is the optimal). Furthermore, let \( \mathcal{A} \) be the set of all possible (solution and non-solution) configurations of the problem; the branch-and-bound algorithm for such combinatorial optimisation problem is given in Algorithm 6.1 (note that, in the algorithm, obviously \( c = f(\text{RelaxedSolution}) \) and \( c^* = f(\text{Solution}^*) \)). Notice that, in the algorithm, besides the \text{Branch} and \text{Bound} operations, there are other two operations that need to be chosen or defined wisely. Namely: the \text{Select} operation and the iteration operation that sets the order by which the elements of the index set \( \{1, ..., n\} \) are iterated. Furthermore, note that we would like that the operation of checking if a relaxed solution is a solution configuration (that is, to check whether \( \text{RelaxedSolution} \in \mathcal{F} \)) is efficient.
\[ \text{LiveSubproblems} \leftarrow \{A\} \]
\[ c^*, \text{Solution}^* \leftarrow \text{None, None} \]

\[ \textbf{while} \ \text{LiveSubproblems} \neq \{\} : \]
\[ \text{Subproblem} \leftarrow \text{SELECT(LiveSubproblems)} \]
\[ \text{LiveSubproblems} \leftarrow \text{LiveSubproblems} \setminus \{\text{Subproblem}\} \]
\[ \text{Subsubproblem}_1, \ldots, \text{Subsubproblem}_n \leftarrow \text{BRANCH(Subproblem)} \]
\[ \text{for } i \in \{1, \ldots, n\} : \]
\[ c, \text{RelaxedSolution} \leftarrow \text{BOUND(Subsubproblem}_i) \]
\[ \text{if } c \text{ is better than } c^* : \]
\[ \text{if } \text{RelaxedSolution} \in \mathcal{F} : \]
\[ c^*, \text{Solution}^* \leftarrow c, \text{RelaxedSolution} \]
\[ \text{else:} \]
\[ \text{LiveSubproblems} \leftarrow \text{LiveSubproblems} \cup \{\text{Subsubproblem}_i\} \]

**Algorithm 6.1.** Branch-and-bound algorithm. The two main operations are branching and bounding. The algorithm terminates with the global optimal solution \( \text{Solution}^* \) of cost \( c^* \).

For example, a good choice is to select the sub-subproblem that is most likely to get killed first so that the tree (with the root on the top) is shallow (we explain the upside-down tree representation shortly). Another good choice is to branch on the most fractional value, but there are really no rules about this – intuition and experience come into play most of the time, in a problem or even in a problem instance basis, and other considerations may need to be taken as well, like (storage) space problems.

In the branch-and-bound algorithm for MILPs, basically:

- the **BOUND** operation solves the LP that results from relaxing the integrality constraints on the domains of the discrete decision variables of the MILP; that is to say, the operation solves the LP (natural) relaxation to obtain an optimistic relaxed solution with a cost that is a lower bound on the optimal solution cost, and

- the **BRANCH** operation splits a problem into smaller subproblems by adding linear constraints to the model such that the resulting subproblems are both collectively exhaustive and mutually exclusive.

A bit more concrete, for any MILP that we cannot solve, by removing the integrality constraints on the domains of the discrete decision variables, we get a LP that we know how to solve efficiently\(^6\) and by solving such LP, we get an optimistic relaxed solution. For any of the integer (that is, discrete) decision variables \( x_i \) of the MILP that we cannot solve, \( x_i \) (in a solution configuration) is necessarily either less-or-equal than \( \lfloor a \rfloor \) or greater-or-equal than \( \lceil a \rceil \) (for a fractional value \( a \) within its relaxed domain). We take \( a \) from the relaxed solution and branch (if necessary) using such constraints (that is to say, using the \( x_i \leq \lfloor a \rfloor \) and \( x_i \geq \lceil a \rceil \) constraints).\(^7\)

---

\(^6\)The gap between solving a LP and a MILP may be the gap between polynomial time and NP-complete. Nonetheless, although LPs can typically be solved in polynomial time, the simplex algorithm (the most efficient and reliable in our days) is exponential in the worst-case.

\(^7\)Of course there are alternative ways to do this.
Maximise:
\[
\sum_{i \in I} v_i \cdot x_i
\]
Subject to:
\[
\sum_{i \in I} w_i \cdot x_i \leq K
\]
\[
x_i \in \{0, 1\}, \forall i \in I
\]

(a) Knapsack problem

<table>
<thead>
<tr>
<th>Item i</th>
<th>Weight $w_i$ [Kg]</th>
<th>Value $v_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>48</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>35</td>
</tr>
</tbody>
</table>

K = 10 [Kg]; $I = \{1, 2, 3\}$

(b) Instance data

Figure 6.1. The knapsack problem and data for one instance of it. The knapsack problem (a) asks to fill a knapsack of weight capacity $K$ with items of weight $w_i$ and value $v_i$ such that the total value of the items that fit in the knapsack is the maximum. For example, for the instance shown in (b), the best we can do is to take items 1 and 3 (which sum up 80 $\$.)

Branching is not necessary if either the optimistic relaxed solution shows that the problem cannot be solved, or if the cost of the optimistic relaxed solution is no better than the cost of the best solution found so far (i.e., the one with the star).

Consider, for example, the knapsack optimisation problem shown in Figure 6.1(a), and the one instance of it shown in Figure 6.1(b). The branch-and-bound solution process is many times visualised using an upside-down tree (see Figure 6.2) where:

- the root of the tree is the (original) optimisation problem,
- the nodes are the subproblems,
- the branches are the result of the branching operation, and
- the leafs are subproblems that were solved (that is, solution configurations) or subproblems that either can only give a suboptimal solution or are not feasible (that is, nodes that we pruned in the tree).

The LP relaxation of the knapsack problem (in other words, the program obtained by replacing the constraints $x_i \in \{0, 1\}$ by $0 \leq x_i \leq 1$ in the program shown in Figure 6.1(a)) decides to take items by value-per-weight priority (so that we can get the maximum possible value in our knapsack of limited weight capacity $K$). A fractional value for a binary decision variable in the relaxed solution means that we cannot take the respective whole item, so we must decide whether we take the item or leave it – this is the intuition behind the branching operation for this problem. The small instance of the knapsack problem shown in Figure 6.1(b) can be solved to optimality with only one branching operation, as explained next.

The execution of the branch-and-bound algorithm for the instance of Figure 6.1(b) is summarised in Figure 6.2. At the root note, the relaxed solution has a fractional value on $x_2$, so we need to decide whether we take or leave item 2. We first try leaving item 2, and we find a relaxed solution with no fractional values that becomes our best solution (to the non-relaxed problem) found so far with cost

---

8This example was taken from the lectures on Discrete Optimisation (Coursera) course released by The University of Melbourne. See https://www.coursera.org/course/optimization. Accessed on October 2014.
Fig. 6.2. Knapsack problem branch-and-bound tree. The instance of the problem shown in Fig. 6.1(b) can be solved to optimality with one single branching. At the root node (a) the relaxed solution has a fractional value on \( x_2 \); therefore, we branch (b) by using the constraints \( x_2 = \lfloor \frac{1}{4} \rfloor \) and \( x_2 = \lceil \frac{1}{4} \rceil \). In the left branch (c) all \( x_1, x_2, \) and \( x_3 \) have integral values in the relaxed solution; therefore, we save it as the best solution found so far with a value of \( c^* \) (that is, we give it the star and we do not need to branch here anymore). In the right branch (d), the relaxed solution has a value \( c \leq c^* \); therefore, we cannot improve \( c^* \) on this branch, and consequently we prune it (that is, we kill it and we do not need to branch here any more).

80 \( \text{[$\$\$]} \). Then we try taking item 2, and we find an optimistic relaxed solution of value 71.3 \( \text{[$\$\$]} \). Therefore, since we already know a solution of better value, we prune the node. Since there are no more nodes alive, the maximum (that is, the global optimum) total value of the items that we can take home in our knapsack is 80 \( \text{[$\$\$]} \).

The branch-and-bound framework can be applied in different settings. For example, it can be applied to the search tree used in constraint programming (CP) where a branching operation is already defined and constraint propagation is performed in every node. Every time a solution is found, a new additional constraint is added to the constraint system asking that any new solution produced is better than those already known; CP can be used to solve optimisation problems this way.

2. Genetic algorithm

The basic idea of simulation-optimisation is that of running a simulation to measure the constraint violations on a system for a given configuration (i.e., to predict by means of simulation whether the behaviour of a system responds as
expected to a given configuration), and then using optimisation to improve the configurations both to eliminate all violations on the hard constraints of the system (as in a solution configuration) and to get an objective function value as close as possible to the global optimum. The simulator (i.e., the machinery that runs a simulation) is many times seen as a black box that may learn from successive runs.

Optimisation algorithms used for simulation-optimisation usually rely on an evolutionary approach from which a genetic algorithm is probably the best representative. Evolutionary approaches, in general, search for optimal configurations by evolving the population of configurations\(^9\) into better and better ones in terms of both feasibility and optimality. The genetic algorithm, in particular, evolves a population (inspired by the ‘survival of the fittest’ principle formulated originally by Charles Darwin\(^10\)) by mimicking the mechanics of natural selection.

The optimisation-simulation framework, in contrast to the branch-and-bound framework described in the previous section, delivers a form of local search. The general local search algorithm, although ultimately inspired in the oldest optimisation idea of trial-and-error, has shown to be powerful when used intelligently (witness, for example, the k-opt algorithm for solving the traveling salesman problem\(^11\) or the success of constraint-based local search on solving large problems that are beyond the capacities of constraint programming), and this fact also motivates to look for ‘intelligence’ in nature to give rise to new bio-inspired algorithms.

A standard local search algorithm, from our perspective, would rely on exploring neighbourhoods of configuration (in contrast to attempting to explore the whole configurations space) by using both violation and differentiation functions; that is, respectively, by using functions to measure the violations on the constraint system and to predict the violations on the constraint system upon a change in the configuration (e.g., upon a move) that can be implemented efficiently. However, in problems where simulations are involved, usually, measuring the constraint violations on the constraint system or predicting what is going to happen with the constraint violations upon a change in the configuration is very likely to require a simulation (in other words, it is not efficient to run).

The focus of optimisation algorithms used in simulation-optimisation frameworks is to try to explore promising configurations using the less amount of simulations. However, some of their disadvantages (that share with most local search algorithms) are the very little theory available to determine the parameters of the algorithm, the necessity to use extra mechanisms to escape from local optima, and

---

\(^9\)Note that, in the context of a genetic algorithm, with ‘population’ we do not mean the set of all configurations, but rather a subset (a sample, in a sense) of optimal solution configurations candidates that we evolve.

\(^10\)Charles Darwin (Shrewsbury 1809 - 1882 London) was the first to create a theory of evolution that was first published in his book ‘On the Origin of Species’ in 1859. A century later, DNA studies revealed evidence of such theory.

\(^11\)The traveling salesman problem is one of the most studied problems in discrete optimisation that is known to be NP-hard. Given a set of cities connected by roads, it asks: what is the shortest route that visits every city and returns to the starting city?
their lack of guarantees about the optimality quality of the solution configuration produced. For many large problems, nevertheless, local search has great success where other techniques fail. Unfortunately, when designing local search algorithms, there are many questions that most of the time need to be answered empirically (therefore, the task of making a genetic algorithm effective remains an art).

Moreover, sometimes within a simulation-optimisation framework, a simulator that relies on a meta-model (instead of the proper model) is used, in addition, for different purposes like generating/identifying potentially feasible configurations, or filtering out potential sub-optimal or bad configurations. For example, a LP can be used, first, to find a relaxed solution configuration to a non-linear program (say a model with a non-linear objective), and, then, such solution configuration can be used together with the simulator to hopefully find a better proper solution.\footnote{In the experiments, we abstract away any kind of simulation run as an operator to be used in our genetic algorithm; for example, an operation to check whether a configuration $a$ is a solution configuration (that is, to determine whether $a \in \mathcal{F}$ or not) is implemented as a simulation run.}

A genetic algorithm normally relies in three main operations: selection, crossover, and mutation. Given a set of configurations (called a population), the selection operation chooses two configurations (that is, it chooses two individuals) from the population that will be operands for the crossover operation. The crossover operation (i.e., the reproduction between two individuals) produces two new configurations (that is, the offspring) by exchanging information between its operands. Each of the new configurations, or one of them, is in turn an operand to the mutation operation which modifies information in order to both try to gain/recover valuable genes and act as a local optima escape mechanism. Genetic algorithms are used in different settings and started to become popular in the early 1970s.\footnote{The piece of work that made genetic algorithms popular is the book ‘Adaptation in Natural and Artificial Systems’ by John Henry Holland published in 1975.}

Note that there are really no recipes on how to create the initial generation of the population. For instance, it may be composed of known sub-optimal solution configurations, sub-optimal solution configurations created at random, configurations apparently close to become a solution, (non-solution) configurations created at random, etc. Note also that, in the context of a genetic algorithm, we speak of configurations in terms of individuals. A configuration needs to be represented (i.e., encoded) so that the algorithm can operate efficiently over this representation. Although sometimes we use the terms configuration and individual interchangeably, an individual is more concretely the representation of a configuration.

Similarly, there are really no recipes on how to choose a codification of configurations into individuals. Although it is very common to see configurations encoded into binary strings (that is, into sequences of 1s and 0s) – that is to say, into sequences of bits (or bit-strings, or, in the context of genetic algorithms, into sequences of genes) – there are other possible representations (for example, as trees, or arrays of integer numbers) that may be more appropriate depending on the problem. The fitness of an individual is determined by a fitness function, and the
intuition behind a fitness function is that of an objective function that operates on individuals (possible incorporating additional components as, for example, a feasibility component). Note genetic algorithms are usually specified as maximisation problems since we look for the fittest individual.

Given the initial generation of the population, new generations are created, one after another, by executing (under the principle of the survival of the fittest) cycles of selection, crossover, and mutation operations. Under these cycles, only the fittest individuals are allowed to have offspring in the next generation. The intuition is that the average fitness of the population will improve one generation after another; therefore, high fitness individuals can probably be found in the population after many cycles. The algorithm terminates when a given termination criterion has been met; for example, when the population has evolved a given maximum number of generations, or when the fittest individual found so far is a good enough solution configuration, or when the fittest individual has not improved its fitness over a given number of generations.

The intuition behind the selection, crossover, and mutation operations is that of reproduction; first, parents are selected from the population at random favouring those with better fitness, and, then, offspring is created from the two selected parents by means of crossover. Individuals with better fitness have better genes, and those are the genes inherited by their offspring. The combination of genes inherited from parents may also be altered in the offspring by means of mutation. Once two parents are selected, however, reproduction does not always happen; crossover and mutation are usually limited to happen with a certain probability. Moreover, selection is sometimes limited as well so that there is no elitism in the sense that the same group of parents do not dominate reproduction.

Before presenting the general genetic algorithm, we make more concrete the crossover and mutation operations. For this purpose, we assume a binary representation of the configurations (that is, we assume an individual is encoded into a binary string, and consequently the genes of an individual are bits).

In the genetic algorithm, basically:

- The Crossover operation, given two parent individuals as operands, produces two children individuals by combining the parents’ genes in some way; usually, by copying in alternation gene segments (that is, sub-bit-strings) defined by crossover points, usually chosen at random (that is to say, given $n$ points chosen at random that define $n + 1$ segments in each of the parent bit-strings, the two children bit-strings are produced by copying in alternation such segments of bits from one and the other parents). The crossover operation is illustrated and exemplified in Figure 6.3.

\[\text{Note that, in this chapter (and the whole thesis), we use the term ‘gene’ in its colloquial sense: ‘this individual has good genes’ or ‘this is the hair color gene’. Some people may argue that our language is not the more precise here, for example, that ‘allele’ is a better term for what we call a ‘gene’, but this makes no difference for our purposes.}\]
6. PRACTICE BASICS

(a) Parents

Parents:

01110100101
1010100101

(b) Offspring

Two parents:

Parents:

01110100101
00011101010

One crossover point:

Two children:

01110100101
10101011010

01110011010
10101100101

Two crossover points:

10101.100101

000.1110.1010

01111100101
00010101010

Two children:

(c) One point crossover operation

(d) Two point crossover operation

Figure 6.3. Crossover operation. Offspring (b) is obtained from parents (a) by copying in alternation segments defined by crossover points. For example, there are two segments for a one-point crossover operation (c), and there are three segments for a two-point crossover operation (d).

(c) One point crossover operation

(d) Two point crossover operation

example.

11011100001110
1101101100110100

1100111000011110
1101010010110100

(a) Original offspring

(b) Mutated offspring

Figure 6.4. Mutation operation. Mutated individuals (b) are obtained by flipping (usually at random) a few bits (usually selected at random) from the originals (a).

- The mutation operation changes offspring’s genetic material by flipping one or more genes (these are, bits) from 1 to 0 or from 0 to 1. The mutation operation is exemplified in Figure 6.4; bits to be flipped are usually chosen at random.

In our experiments, we use a binary encoding; therefore, the previous are good descriptions of how the crossover and mutation operations that we implemented operate. However, again, crossover operations can be rather complicated, and, as well as mutation operations, are highly dependent on the encoding. The genetic algorithm for solving combinatorial optimisation problems is shown in Algorithm 6.2; in the algorithm, we assume that:

- the termination criterion is that either a maximum number of generations $j$ has been reached or the fitness of the fittest has not improved in $j_{\text{stop}}$ generations,
- an initial generation $\text{Generation}_0$ is provided (say, generated at random),
- the crossover and mutation operations are statistically controlled, respectively, by probabilities $p_c$ and $p_m$, and
- the fitness function is $f$. 
2. GENETIC ALGORITHM

Generation ← \( \text{Generation}_0 \) \( \triangleright \) e.g., with individuals generated at random
\( \text{sizePopulation} \leftarrow |\text{Generation}| \)

for \( i \in \text{Generation} \):
    \( i.\text{fitness} \leftarrow f(i) \)

\( i^*, f^* \leftarrow \text{Best}(\text{Generation}) \) \( \triangleright \) where \( f^* = i^*.\text{fitness} \)

\( j, j_{\text{best}} \leftarrow 0, 0 \)

while \( j \leq j \) and \( j_{\text{best}} \leq j_{\text{stop}} \):
    \( \text{NextGeneration} \leftarrow \{ \} \)
    while \( |\text{NextGeneration}| < \text{sizePopulation} \):
        \( i_1, i_2 \leftarrow \text{SELECT}(\text{Generation}) \)
        \( c_1, c_2 \leftarrow \text{CROSSOVER}(i_1, i_2, p_c) \)
        for \( i \in \{ c_1, c_2 \} \):
            \( m \leftarrow \text{MUTATE}(i, p_m) \)
            \( m.\text{fitness} \leftarrow f(m) \)
            \( \text{NextGeneration} \leftarrow \text{NextGeneration} \cup \{ m \} \)
        \( i^*_j, f^*_j \leftarrow \text{Best}(\text{NextGeneration}) \)
        if \( f^*_j \geq f^* \):
            \( i^*, f^* \leftarrow i^*_j, f^*_j \)
            \( j_{\text{best}} \leftarrow 0 \)
        else:
            \( j_{\text{best}} \leftarrow j_{\text{best}} + 1 \)
    \( \text{Generation} \leftarrow \text{NextGeneration} \)
    \( j \leftarrow j + 1 \)

ALGORITHM 6.2. Genetic algorithm. The three main operations are selection, crossover, and mutation. The algorithm terminates with the best individual found so far \( i^* \) of fitness \( f^* \).

In Algorithm 6.2, note that besides the \text{SELECT}, \text{CROSSOVER}, and \text{MUTATE} operations, there are other aspects that may need special attention. For example, the \text{SELECT} operation itself may limit the number of offspring an individual can have to a few percent of the new generation population. Furthermore, sometimes an \text{ACCEPT} operation is incorporated as well as a mechanism to decide whether an individual is accepted or not in the new generation (that is to say, sometimes mutation does it well to escape from local optima, but sometimes an additional meta-heuristic like tabu search or simulated annealing may be incorporated as well). When using meta-heuristics, however, other parameters come into play, and (again) there are really no rules about how to tune parameters (therefore, intuition and experience come into play most of the time in a problem or even in a problem instance basis).

Parameter tuning aside, to finalise this chapter, consider (as a toy example) the knapsack problem instance shown in Figure 6.1. If we encode configurations into bit-string individuals, with one bit per item option (where 1 means an item is chosen, and 0 means an item is not chosen to be taken home in your knapsack), the

\(^{15}\)For example, the idea of simulated annealing \([21]\) – which gets its name and inspiration from the form of crystallisation that occurs as molten metals are cooled – is that of accepting a non-improving individual in the population with the metropolis probability. Such probability has a temperature parameter \( t \) that decreases as the population evolves, and its goal is to accept many moves initially and then move towards small values of \( t \).
fitness function can measure the total value of the chosen objects with a significant bonus if the total weight of the chosen objects does not exceed the capacity of the knapsack:

\[ \sum_{i \in I} v_i \cdot x_i + \begin{cases} 128 & \text{if } \sum_{i \in I} w_i \cdot x_i \leq K \\ \text{otherwise} & \end{cases} \]

We create a population by a random walk that results in: (001; 000; 011; 110) with fitness (163; 128; 83; 93) respectively, and evolve it:

Select (001; 000; 011; 110) = (001; 000)
Crossover (00.1; 00.0) = (000; 001)
Mutate (000) = (010)
Mutate (001) = (001)
Select (001; 000; 011; 110) = (001; 100)
Crossover (0.01; 1.00) = (101; 000)
Mutate (101) = (101)
Mutate (000) = (100)

so that the new generation becomes: (010; 001; 101; 100) with fitness (176; 163; 208; 173) respectively, where the best individual \( i^* \) is 101 with a fitness value \( f^* \) of 208. Then we repeat until the termination criterion is satisfied.

Of course this instance of the knapsack problem is too small and applying to it either a branch-and-bound or a genetic algorithm is equivalent to using a sledgehammer to crack a nut.\(^{16}\) However, for our problem on power transmission networks expansion, we believe these algorithms can become really useful. We design our experiments in the next chapter.

\(^{16}\)That is to say, it is just disproportionate force to overcome such a small problem.
CHAPTER 7

Design of the experiments

This chapter describes all of the experiments that we run\(^1\) in order to investigate the research question and justifies some of the decisions we had to make as, for example, the selection of the methods and network instances that we use in the study. There seems to be a limitation in terms of availability and quality of network-instances data for this problem in the sense that, especially when considering the relevant data for an AC load flow, data is not always available or clear. For our study, we make a case with the most common instances seen in the literature for transmission network expansion planning problems (see Table 3.1 for reference). These are 6, 24, and 46-bus networks (that can be found in several publications; e.g., [13], [10], and [14], respectively). We take the 6-bus network from [29], the 24-bus network from MatPower’s distribution case files, and the 46-bus network from [14]. However, there are some variations of these networks in the literature, and a few that we make on our own, so we review all of them in detail in Appendix B. Note that we take the investment cost data from paper [29] for both the 6 and the 24-bus networks, and from paper [14] for the 46-bus network.

A casual reader may argue that the size of the chosen networks cannot be realistic. However, the networks are realistic in the sense that these are the cases most published work used for research, and represent the most detailed available data.\(^2\) Moreover, most studies in the literature show how their proposed methods work in one small network only, say the 6-bus network, and, in addition, take any other larger network (to do the same) for which the network data is many times unavailable or not entirely available. The approach that we take here is to, in addition to make a case-study with the 3 mentioned classic networks, generate TNEP network instances from the MatPower’s distribution test cases (as explained in Appendix B) in order to have a wider collection of cases to test our methods.

\(^1\)The HAC-TNEP algorithm shown in Figure 7.2 in this chapter is explained using words in Subsection III.A of our PSCC 2014 proceedings paper [2]. Similarly, the expressions to measure the thermal limits and voltage bounds constraints violations shown, respectively, in equations 50 and 51 in this chapter are shown and explained in Subsection III.E of the mentioned paper. Furthermore, the experimental design for the evaluation on classic test systems, TNEP with VAr compensation, and power market considerations experiments presented in this chapter are explained in like manner at the beginning of Sections V, VI, and VII of the same paper.

\(^2\)It may be pertinent to mention here that Bent et al., as an exception for AC transmission network expansion planning studies, studied a one order of magnitude larger test case [3]; that is, a network with 900 buses and 1020 corridors that comes from the state of New Mexico in the United States of America.
on. From these test networks, we aim to have an idea on how our case-study could generalize to other networks. These are 9, 14, 30, 39, 57, and 118-bus networks that we call the test case networks or the test-set.³

As made concrete in Chapter 5, our goal is to investigate whether the LPAC constraints are a good mechanism to finding low cost AC feasible expansion plans in the context of transmission network expansion planning, if possible satisfying the n – 1 reliability criterion. In the experiments, we explore the problem:⁴

1. using mixed-integer linear programming formulations together with state-of-the-art industry standard tools, and

2. using a genetic algorithm, in hope of scalability, that relies on a meta-simulation that uses the LPAC (linear) constraints.⁵

The method consists then in both exploring the modelling part of the problem, by experimenting with alternative models together with state-of-the-art implementations of algorithms,⁶ and exploring the search part of the problem, by experimenting with the way a genetic algorithm operates.⁷ Once a network solution configuration is found, we first make note of its cost, then configure the network as per the solution, and finally let the network run to measure the quality of the solution in terms of both thermal limits and voltage bounds constraint violations.

Thermal limit violations measure how overloaded circuits are. The relative thermal limit violation of a circuit n–m is defined as the maximum relative thermal violation perceived at either end of the circuit:

\[
\max \left( \frac{\sqrt{p_{nm}^2 + q_{nm}^2}}{s_{nm}}, \frac{\sqrt{p_{mn}^2 + q_{mn}^2}}{s_{mn}} \right)
\]

therefore, we say that a circuit is overloaded (or, sometimes, ‘violated’) when this expression evaluates to a value above 100%.⁸

Voltage bounds violations measure in what degree voltage root mean square values are either above or below the desired limits. The voltage bounds violation at a bus n is defined as the maximum violation either above or below the respective voltage bounds:

\[
\max \left( 0, \tilde{e}_{n} - \tilde{\tilde{e}}_{n}, \tilde{\tilde{e}}_{n} - \tilde{e}_{n} \right)
\]

³As a machine learning practitioner would probably do.
⁴Note the basics of these two algorithms are reviewed in a general context in Chapter 6.
⁵Roughly obtained from the mixed-integer linear programs.
⁶Note there is also parameter tuning involved when using such implementations.
⁷Genetic algorithms, as an intelligent way to perform a local search, have very little rigorously established about performance or convergence (in contrast, for example, to mathematical programming), but it often works remarkably well in some large problems; hence (as per the literature review), it is worth to try.
⁸For example, consider two circuits n₁–n₂ and n₃–n₄, both with a thermal limit of 100 [MVA]. Say \((p_{n₁n₂}, q_{n₁n₂}) = (52.19, 55.55)\) [MW, MVAR] and \((p_{n₂n₁}, q_{n₂n₁}) = (-49.35, -49.48)\) [MW, MVAR], and, similarly, \((p_{n₃n₄}, q_{n₃n₄}) = (-110.39, -9.97)\) and \((p_{n₄n₃}, q_{n₄n₃}) = (110.69, 21.16)\). As per the thermal limit violations measure, circuit n₁–n₂ has a load of 76.22% = \(\max(0.7622, 0.6888)\), and circuit n₃–n₄ is overloaded by 12.69% since 112.69% = \(\max(1.1084, 1.1269)\).
therefore, we say a bus respects the voltage bounds if and only if this expression evaluates to zero.\footnote{For example, consider two buses \( n_1 \) and \( n_2 \), both with voltage bounds of 0.95 and 1.05 [V]. Say, \( \bar{e}_{n_1} = 0.93 \) [V] and \( \bar{e}_{n_2} = 1.08 \) [V]. As per the voltage bound violations measure, bus \( n_1 \) has a voltage bound violation of 0.02 = \( \max(0, 0.02, -0.12) \), and bus \( n_2 \) has a voltage bound violation of 0.03 = \( \max(0, -0.13, 0.03) \).}

We would like both that no circuit is overloaded and that all voltage magnitudes are kept within their bounds; that is, we would like that the network operates normally (in all of the scenarios).

In the next chapter, we report for a given network the maximum and average operational constraint violations among all circuits and buses. Furthermore, for any of these two measures, when evaluating a network against a set of scenarios of cardinality greater than one, the maximum and average violations are computed (for circuits and buses) among all the scenarios. In subsequent discussions, we take the constraint violations (as defined by the two measures described above) as the evidence that we can use to say that we can find AC feasible configurations.

It is convenient to recall at this point that an expansion plan is only part of the configuration. In other words, to configure the network (and to do a run), we also need to know the dispatch, the power (injected and) demanded at the \( p-q \) buses, and the \( p \) and \( \tilde{e} \) values for the voltage-controlled buses. We try to find the AC feasible configuration that involves the lowest cost expansion plan as per the constrained optimisation model shown in Model 4.1 (that is, as per the core model).

In the core model, the decision variables are roughly:\footnote{Recall also, when looking at the core model, that we defined \( \theta_{nm} \) as \( \theta_n - \theta_m \).}

- \( \tilde{e}_n \angle \theta_n \); i.e., the voltage at bus \( n \),
  \[-p_{n-m} + j \cdot q_{n-m}; \] i.e., the flow of power in a circuit with ends at buses \( n \) and \( m \).
- \( p_n^g + j \cdot q_n^g \); i.e., the power injected at bus \( n \),
  \[-p_{nk}; \] i.e., the real power injected at bus \( n \) by the \( k \) generating-unit connected to it.
- \( \psi_{nm} \in \{0, 1\} \); i.e., the status of a \( k \) circuit option in corridor \( n-m \).

As per the LPAC (linear) power-flow constraints (derived in Section 2 of Appendix A for reference), \( \tilde{e}_n = 1 + \phi_n \) by definition. Therefore, the decision variables \( \tilde{e}_n \) become (in a sense) the voltage changes \( \phi_n \) decision variables, and, for convenience, we define \( \phi_{nm} \) as \( \phi_n + \phi_m \). As per the LPAC (linear) power-flow constraints, a polyhedral relaxation to the cos function is also introduced that translates into \( \hat{\cos}_{nm} \in [0, 1] \) decision variables that appear in the model to approximate the values of \( \cos \theta_{nm} \). Finally, as per the LPAC (linear) power-flow constraints notation, the variables for reactive power on the circuits are expressed as a sum of target and delta components; that is, \( q_{n-m} \) is expressed as \( q_{n-m}^t + q_{n-m}^\Delta \).

In the LPAC-TNEP model, the decision variables are roughly:

- \( \phi_n, \theta_n, \) and \( \hat{\cos}_{nm} \); i.e., the voltage (magnitude deviation and angle) at bus \( n \), and an approximate to \( \cos \theta_{nm} \),
  \[-p_{n-m} + j \cdot (q_{n-m}^t + q_{n-m}^\Delta); \] i.e., the flow in circuit \( n-m \).
• $\mathbf{P}_n^j + j \cdot Q_n^j$; i.e., the power injected at bus $n$,
  \(- P_{nk}$; i.e., the real power injected at bus $n$ by the $k$ generating-
  unit connected to it.

• $\psi_{nm} \in \{0, 1\}$; i.e., the status of a $k$ circuit option in corridor $n-m$.

In the LPAC-TNEP model, the objective function is the same as in the core model; that is to say, the objective is to minimise the expansion investment cost.

As for the constraints, in the LPAC-TNEP model, the non-linear non-convex AC power-flow constraints are replaced by the LPAC (linear) power-flow constraints. Concretely:

For all bus $n$,

\begin{align}
\mathbf{P}_n^j - \mathbf{P}_n^l &= (1 + 2 \cdot \phi_n) \cdot g_n^k + \sum_{n \sim m} \mathbf{P}_{n \sim m} \cdot \psi_{nm} \\
Q_n^j - Q_n^l &= -(1 + 2 \cdot \phi_n) \cdot b_n^k + \sum_{n \sim m} (Q_{n \sim m}^\Delta + Q_{n \sim m}^\Delta) \cdot \psi_{nm} \\
\end{align}

where:

\begin{align}
\mathbf{P}_{nm} &= \frac{1}{t_{nm}} \cdot g_{nm} - \frac{1}{t_{nm}} \cdot g_{nm} \cdot \cos \theta_{nm} - \frac{1}{t_{nm}} \cdot b_{nm} \cdot \theta_{nm} \\
\mathbf{P}_{mn} &= g_{nm} - \frac{1}{t_{nm}} \cdot g_{nm} \cdot \cos \theta_{nm} - \frac{1}{t_{nm}} \cdot b_{nm} \cdot \theta_{mn} \\
Q_{nm}^l &= -\frac{1}{t_{nm}} \left( b_{nm} + \frac{b_{nm}^2}{2} \right) - \frac{1}{t_{nm}} \cdot g_{nm} \cdot \theta_{nm} + \frac{1}{t_{nm}} \cdot b_{nm} \cdot \cos \theta_{nm} \\
Q_{nm}^l &= -2 \cdot \phi_n \cdot \left( b_{nm} + \frac{b_{nm}^2}{2} \right) + \frac{\phi_{nm}}{t_{nm}} \cdot b_{nm} \\
Q_{mn}^\Delta &= -\frac{2}{t_{nm}} \cdot \phi_n \cdot \frac{b_{nm}^2}{2} + \frac{\phi_{nm}}{t_{nm}} \cdot b_{nm} \\
Q_{mn}^\Delta &= -2 \cdot \phi_m \cdot \frac{b_{nm}^2}{2} + \frac{\phi_{mn}}{t_{mn}} \cdot b_{nm} \\
\end{align}

Linearising the non-linear power-flow constraints is an active area of research, and, unfortunately, these linearised constraints become non-linear in the context of TNEP, and, furthermore, they become non-convex. A mixed-integer linear program does not result naturally because the discrete decision variables, $\psi$, become factors in the equality constraints for the power-flows:

\begin{align}
\sum_{n \sim m} \mathbf{P}_{n \sim m} \cdot \psi_{nm} &= \mathbf{P}_n - (1 + 2 \cdot \phi_n) \cdot g_n^k \\
\sum_{n \sim m} Q_{n \sim m} \cdot \psi_{nm} &= Q_n + (1 + 2 \cdot \phi_n) \cdot b_n^k \\
\end{align}

As a consequence, we resort to a disjunctive formulation \[31\] to deal with the resulting non-linearities as illustrated next. For the sake of simplicity, consider the DC power-flow linear constraints (where $\mathbf{P}_{nm} = -b_{nm} \cdot \theta_{nm} = -\mathbf{P}_{mn}$ and reactive power is ignored). The constraints of the form $\mathbf{P}_{nm} = -b_{nm} \cdot \theta_{nm}$ are restated as:

\begin{align}
|\mathbf{P}_{nm} + b_{nm} \cdot \theta_{nm}| &\leq M \cdot (1 - \psi_{nm}) \\
\end{align}
for those circuits that are options; thus, the constraints hold (for these circuits) when $\psi_{nm} = 1$ and are ignored when $\psi_{nm} = 0$. One disadvantage of this approach is the necessity to define a suitable big value for $M$ [5]. Nonetheless, here we use the thermal limits of the circuits as a big enough value.

The linear power-flow constraints in both the LPAC-TNEP and DC-TNEP models that we use are formulated as per the disjunctive model. For this purpose, we define the set $C_o$ as the set of circuit options such that, for all circuits in this set, the equality constraints for $p_{nm}$, $p_{mn}$ and $q_{nm}$, $q_{mn}$ (that is, for $p_{nm}$, $p_{mn}$ and $q_{nm}$, $q_{mn}$ in the LPAC-TNEP model) are formulated as per the disjunctive model. As a result, the power-flow equality constraints:

$$\sum_{n\sim m} p_{n\sim m} = p_n - (1 + 2 \cdot \phi_n) \cdot g_n^h \quad \text{and}$$

$$\sum_{n\sim m} q_{n\sim m} = q_n + (1 + 2 \cdot \phi_n) \cdot b_n^h$$

are now linear.

Another set of non-linear constraints in the AC model is the set of thermal limit constraints:

$$\sqrt{p_{mn}^2 + q_{mn}^2} \leq \hat{s}_{mn} \quad \text{(65)}$$

$$\sqrt{p_{mn}^2 + q_{mn}^2} \leq \hat{s}_{mn} \quad \text{(66)}$$

that, in the case of DC-TNEP, since reactive power is ignored (i.e., since $q_{nm} = q_{mn} = 0$), become $|\overline{p}_{nm}| \leq \hat{s}_{nm}$ (hence linear). However, in the case of LPAC-TNEP, we linearise the thermal limit constraints on the circuits by using a polyhedral outer approximation whose constraints are generated with Algorithm 7.1 where $nbSegments$ is the number of segments to use in the approximation that we set to about 20.\textsuperscript{12}

\begin{verbatim}
PWL<cap>(p, q, \hat{s});
    current, step ← 0, (2 * π) / nbSegments
    for i ∈ [0, nbSegments];
    a, b ← sin(current) \cdot \hat{s}, cos(current) \cdot \hat{s}
    CONSTRAIN(a \cdot p + b \cdot q ≤ \hat{s}^2)
    current ← current + step
\end{verbatim}

Algorithm 7.1. PWL<cap> algorithm. Thermal limit constraints on the circuits are generated using a polyhedral outer approximation.

\textsuperscript{11}Unfortunately, this can make the linear relaxation a poor approximation to be optimal since an assignment of 0.5 instead of 0 or 1 could work. Handling disjunctive or ‘on/off’ constraints is an active research area on its own discussed elsewhere [16].

\textsuperscript{12}The derivation of these constraints is straightforward from the equation of a line in a point of tangency $(x_0, y_0)$: $y - y_0 = -\frac{x_0}{y_0} \cdot (x - x_0)$. From there, $y_0 - y_0^2 = -x_0 \cdot x + x_0^2 \Rightarrow x_0 \cdot x + y_0 \cdot y = x_0^2 + y_0^2 \Rightarrow x_0 \cdot x + y_0 \cdot y = r^2$. Since we want the points inside the circles, the constraints have the form $x_0 \cdot x + y_0 \cdot y ≤ r^2$. The number of segments is the number of tangents. We make a note here that by the end of the experiments, when using the genetic algorithm, we switched to the original quadratic constraints which is straightforward to do using the Python port of Gurobi. By the very end, we also used an AC-OPF program.
Minimize:
\[ \sum_{n,m} \gamma_{nm} \cdot \psi_{nm} \]  
That is, the expansion investment cost.

Subject to:
For all scenario \( s \in \mathcal{S} \),  
> Note we do not include ‘\( s \)' in the notation  
> Note \( \psi_{nm} \) are shared among all \( s \) in \( \mathcal{S} \).

For all bus \( n \),  
> The power-flow constraints:
\[
P_{mn}^p - \bar{P}_{mn} = g_{nm}^p + 2 \cdot \phi_n \cdot g_{nh}^p + \sum_{n' \neq m} (q_{n'n} \cdot \phi_{n'n} + \bar{q}_{n'n} \cdot \bar{q}_{m'm}) \quad \text{if } n \notin \mathcal{G}
\]
where:

If \( n \sim m \in \mathcal{C} \setminus \mathcal{C}_0 \),  
> Note \( k \) is implicit in the notation
\[
P_{mn}^p = \frac{1}{\bar{r}_n} \cdot g_{nm} - \frac{1}{\bar{r}_m} \cdot g_{nm} \cdot \cos \theta_{mn} - \frac{1}{\bar{r}_m} \cdot b_{nm} \cdot \theta_{mn}
\]
\[
q_{mn}^p = -\frac{1}{\bar{r}_n} \cdot \left( b_{nm} + \frac{\bar{v}_n}{\bar{r}_n} \right) - \frac{1}{\bar{r}_m} \cdot g_{nm} \cdot \theta_{mn} + \frac{1}{\bar{r}_m} \cdot b_{nm} \cdot \cos \theta_{mn}
\]
\[
q_{mn}^q = -\frac{2}{\bar{r}_m} \cdot \left( b_{nm} + \frac{\bar{v}_n}{\bar{r}_n} \right) + \frac{g_{nm}}{\bar{r}_m} \cdot b_{nm}
\]
\[
q_{mn}^\circ = -2 \cdot \phi_m \cdot \left( b_{nm} + \frac{\bar{v}_n}{\bar{r}_n} \right) + \bar{q}_{nm} \cdot b_{nm}
\]
else:
> For all circuit \( n \sim m \in \mathcal{C}_0 \)
\[
\left| \frac{1}{\bar{r}_n} \cdot g_{nm} - \frac{1}{\bar{r}_m} \cdot g_{nm} \cdot \cos \theta_{mn} - \frac{1}{\bar{r}_m} \cdot b_{nm} \cdot \theta_{mn} - P_{mn} \right| \leq M \cdot (1 - \psi_{nm})
\]
\[
g_{nm} = -\frac{1}{\bar{r}_m} \cdot g_{nm} \cdot \cos \theta_{mn} - \frac{1}{\bar{r}_m} \cdot b_{nm} \cdot \theta_{mn} - P_{mn} \leq M \cdot (1 - \psi_{nm})
\]
\[
-\frac{1}{\bar{r}_m} \cdot g_{nm} + \frac{\bar{v}_n}{\bar{r}_n} \left( b_{nm} + \frac{\bar{v}_n}{\bar{r}_n} \right) - \frac{1}{\bar{r}_m} \cdot g_{nm} \cdot \theta_{mn} + \frac{1}{\bar{r}_m} \cdot b_{nm} \cdot \cos \theta_{mn} - q_{mn}^q \leq M \cdot (1 - \psi_{nm})
\]
\[
-\frac{2}{\bar{r}_m} \cdot \left( b_{nm} + \frac{\bar{v}_n}{\bar{r}_n} \right) + \frac{g_{nm}}{\bar{r}_m} \cdot b_{nm} - q_{nm}^q \leq M \cdot (1 - \psi_{nm})
\]
\[
-2 \cdot \phi_m \cdot \left( b_{nm} + \frac{\bar{v}_n}{\bar{r}_n} \right) + \bar{q}_{nm} \cdot b_{nm} - q_{nm}^\circ \leq M \cdot (1 - \psi_{nm})
\]

\( \theta_{slackBus} = 0 \)

For all circuit \( n \sim m \) in \( n \sim m \).
> The circuit operational constraints:
\[
PWL<\text{CAP}>(\bar{P}_{mn}, \bar{q}_{nm}^p, \bar{q}_{nm}^q), \delta_{nm}, \text{nbSegments}_{cap}
\]
\[
PWL<\text{CAP}>(\bar{P}_{mn}, \bar{q}_{nm}^p, \bar{q}_{nm}^q), \delta_{nm}, \text{nbSegments}_{cap}
\]

For all bus \( n \),
> The bus operational constraints:
\[
\bar{c}_n \leq 1 + \phi_n \leq \bar{c}_n
\]

For all generating unit \( n-k \),
> The gen-unit operational constraints:
\[
PWL<\cos>(\cos \theta_{nm} - \frac{\bar{v}_n}{\bar{r}_n} + \frac{\bar{v}_n}{\bar{r}_n}, \text{nbSegments}_{cos}) \quad \text{cos approximation}
\]
\[
PWL<\cos>(\cos \theta_{nm} - \frac{\bar{v}_n}{\bar{r}_n} + \frac{\bar{v}_n}{\bar{r}_n}, \text{nbSegments}_{cos})
\]

For all \( k \in [1, \psi_s] \),
> Symmetry breaking constraints
\[
\psi_{nmk} = \psi_{nm}(k+1)
\]

Model 7.1. LPAC-TNEP model. A linear constrained optimisation model for TNEP based on the LPAC power-flow constraints.

In the experiments, we also use about 20 segments for the cos approximation
\[
PWL<\cos>^{13}
\] implemented as per the LPAC approximation algorithm [9]. Note that, when using PWL<CAP>, \textit{14} we additionally ensure to make zero the power-flows in any circuit option that has a 0-th status.

\textsuperscript{13}PWL stands for ‘piecewise linear’.

\textsuperscript{14}CAP stands for ‘capacity’. Thermal limit constraints can be seen as capacity constraints.
Minimize:  
$$\sum_{n-m} \gamma_{nm} \cdot \psi_{nm} \quad \triangleright \text{That is, the expansion investment cost.}$$

Subject to:
For all scenario $s$ in $S$,  
$$\triangleright \text{Note we do not include } 's' \text{ in the notation}$$
$$\triangleright \text{Note } \psi_{nm} \text{ are shared among all } s \text{ in } S.$$  
For all bus $n$,  
$$\triangleright \text{The power-flow constraints:}$$
$$\bar{p}_n^k - \bar{p}_m = \sum_k p_{nm} - \sum_k p_{mn}$$
where:
- If $n \sim m \in \mathcal{C} \setminus \mathcal{C}_o$,  
$$p_{nm} = -b_{nm} \cdot \theta_{nm}$$
- Else:  
$$|b_{nm} \cdot \theta_{mn} - p_{nm}| \leq M \cdot (1 - \psi_{nm})$$
$\theta_{slackBus} = 0$

For all circuit $n-m$ in $n-m$,  
$$\triangleright \text{The circuit operational constraints:}$$
$$|p_{nm}| \leq \hat{s}_{nm} \cdot \psi_{nm} \quad \text{if } n-m \in \mathcal{C}_o$$
else  
$$|p_{nm}| \leq s_{nm}$$
For all generating unit $n-k$,  
$$\triangleright \text{The gen-unit operational constraints:}$$
$$\hat{p}_{nk} \leq p_{nk} \leq \tilde{p}_{nk}$$
where  
$$\sum_k p_{nk}^k = p_{gn}$$
For all corridor $n-m$:
For all $k \in [1, \hat{\psi}]$:  
$$\triangleright \text{Symmetry breaking constraints}$$
$$\psi_{nmk} \geq \psi_{nm(k+1)}$$

**Model 7.2. DC-TNEP model.** A linear constrained optimisation model for TNEP based on the DC power-flow constraints.

We give the full linear model (for reference) in Model 7.1 (where $\mathcal{G}$ is the set of voltage-controlled buses). In the model, the voltage bound constraints become:

(67)  
$$\hat{e}_n \leq 1 + \phi_n \leq \bar{e}_n \quad \text{for all bus } n$$

Note that in the model we include, in addition, constraints asking that for all corridor $n-m$, where we allow up to $\hat{\psi}$ circuit options, the following holds:

(68)  
$$\forall k \in [1, \hat{\psi}], \quad \psi_{nmk} \geq \psi_{nm(k+1)}$$

These are symmetry breaking constraints. Each circuit option has one binary decision variable associated to it and circuit options in the same corridor are indistinguishable; thus we choose circuit options in order. For reference, the full DC-TNEP model is given in Model 7.2.

In the first set of experiments, described shortly, we also propose an AC-based destructive heuristic (HAC-TNEP) that contrasts several constructive heuristics proposed in the literature. HAC-TNEP behaves better than Bent et al. constructive heuristic [3], produces high quality primal solutions, thus gives good basis for comparison. HAC-TNEP starts with all of the circuit options added to the network (there are sufficiently many to guarantee AC feasibility). Iteratively, HAC-TNEP attempts to remove one circuit option from the network, circuits are selected in increasing order of relative loads, and tests if the resulting network is still AC feasible in an AC optimal power flow model (AC-OPF).\textsuperscript{15} The algorithm terminates when

\textsuperscript{15}Relative loads are obtained from the AC-OPF solution (and such information could also be used to have an idea on where could be helpful to install reactive power sources – not done here).
Algorithm 7.2. HAC-TNEP algorithm. It starts with all circuit options. On each iteration, it attempts to remove one circuit option prioritising them by their relative loading (remove least heavily loaded circuits first). Once no circuit option can be removed, the algorithm finishes at that local minimum with the resulting expansion in \( \text{expansion} \).

no circuit option can be removed from the network (that is, when the cost cannot be reduced). We show the HAC-TNEP algorithm, for reference, in Figure 7.2. The purpose of this algorithm is not to be a state-of-the-art solution, but to provide a feasible AC-TNEP solution as a point of reference for the other methods.\(^\text{16}\)

We could argue at this point that, so far in this chapter, we cared more about the semantics of the problem rather than in its representation. The next approach is orthogonal since, by trying a genetic algorithm, we focus more in the representation of the problem (as a set of strings)\(^\text{17}\) rather than in its semantics. We assign semantics to such strings, but it seems of little help to solve the problem since, apparently, this problem has scarce structure to exploit for engineering a solution.

We explain the genetic algorithm we used in the context of Algorithm 6.2.\(^\text{18}\) That is to say, we explain how we structured and configured the genetic algorithm within the framework described in Section 2 of Chapter 6. The genetic algorithm we use has the more conservative structure that has been found to work for the TNEP problem in the literature (e.g., see [12] and [33]). More concretely, the algorithm relies on a fitness function that drives the search towards both feasibility and optimality, a two point crossover operation, and a mutation operation with both an increasing mutation probability and a filter based on the metropolis probability.

The individuals in the population are expansion plans that we encode, for simplicity and efficiency, into bit-strings. Each bit-string has one bit per circuit expansion option that takes the value 1 if the circuit option is chosen to be included in the expansion plan and the value 0 otherwise. The mutation and crossover

\(^{16}\)Note also that HAC-TNEP can be extended to use a set \( S \) of scenarios, by taking the line loads of the circuits among all the scenarios, but it is very likely to entail a prohibitive number of AC-OPF runs in our test cases (if we attempt to make an AC-OPF on each sub-network).

\(^{17}\)A problem (that is, a language) and a set of strings are the same thing.

\(^{18}\)One of the main goals to switch to a genetic algorithm was to see if we could handle the constraints for the \( n - 1 \) reliability criterion this way.
operations are, consequently, exactly as described in Chapter 6 and illustrated in Figures 6.3 and 6.4 (that is, a mutation is the result of a logical not operation in one or more bits, and a crossover the result of swapping in alternation bit substrings that are defined by crossover points chosen at random). The algorithm also relies on the meta-model obtained from Model 7.1 by fixing the discrete decision variables to test for potential AC feasibility of an expansion plan.

The fitness function has two components. The optimality component that specifies the investment of the chosen expansion plan, and the feasibility component that penalises the cost for each scenario where the network is not AC feasible. Testing for AC feasibility whenever the quality of an expansion plan needs to be evaluated (if possible) can be expensive to do, therefore we resort to the meta-model to both fill-in what remains empty in the configuration and have a good indicator that the expansion plan leads to a potentially AC feasible expanded network. By using the meta-model, checking for potential feasibility in all sub-networks entails running |S| linear programs; the idea is to let them run in parallel on different cores.

Genetic algorithms are usually expressed as maximisation problems; we do the same. The individual fitness function $f$ for an expansion of cost $c$ which is not feasible in $n$ scenarios is computed as:

$$
\frac{1}{c \cdot (n + 1)} + (1 \text{ if } n = 0 \text{ else } 0)
$$

where the cost is penalised by $n + 1$ instead of $n$ to avoid divisions by zero, and a value of 1 is added whenever the expansion is feasible under all scenarios (so that these valuable genes are much more prone to survive); the higher the fitness of an individual, the higher the chances that the individual participates in reproduction.

Note that the population is no more than an ordered set of bit-strings ranked by fitness that we implemented as a heap. Since we aim to maximise the fitness, we say that individuals with better fitness values are at the top of the population. The selection operation chooses two individuals that are at the top of the population, and, to avoid dominance of genes, each individual is allowed to participate in reproduction in proportion to its fitness. The maximum number of offspring an individual $i$ can have is given by:

$$
\frac{f(i)}{f(i^*)} \cdot \text{limitedSelection} \cdot |\text{Generation}|
$$

where $|\text{Generation}|$ is the size of the population. This way, the best individual $i^*$ can have limitedSelection% of the new generation population as offspring at the maximum (recall reproduction does not always succeed).

We determined experimentally that, among different variations of crossover operations, the two point crossover operation, where crossover points are chosen at random, is the best choice for this problem. Once two individuals are selected, reproduction by crossover happens with probability $p_c$. Moreover, from the two children born due to a successful crossover, we only let the fittest of them survive.
The survivor child, before passing to the next generation, is submitted to a mutation operation that materialises with probability $p_m$. Such probability increases slowly each new generation to counteract the homogenisation due to the crossover operation. However, a filter based on the metropolis probability is also included (in the spirit of a meta-heuristic) where the mutation is accepted with probability

\[ 1 \text{ if } \Delta \leq 0 \text{ else } e^{-\Delta} \]

where $\Delta = f(i) - f(\text{Mutate}(i))$ and the temperature $T$ (naturally) decreases with time (see optimisation by simulated annealing [21]). Note that a proper evaluation of the fitness function necessitates simulation runs; however, we resort to the linear meta-model based on the LPAC (linear) power-flow constraints (and it is in this sense that we say that we use meta-simulations to check for potential AC feasibility).

The cycles of selection, crossover, and mutation, happen until a new generation of the same population-size $x$ is created; thus approximately $x$ evaluations of the fitness function are required. If each of these evaluations needs to check for $n-1$ feasibility, this number increases by a factor of $n$. That is to say, this needs to run approximately $x \cdot n$ linear programs. Checking for $n-1$ feasibility in parallel can deliver a good speedup and with $n$ available processing units, we could check for $n-1$ feasibility in one parallel step (instead of in $n$ sequential steps). We attempt to speedup the implementation using such an idea by implementing a map-reduce technique on the maximum number of cores available in one of our machines. Additionally, we try other tricks to speedup the implementation.

Evaluating the fitness function is the operation that dominates the time in the genetic algorithm. In order to improve the efficiency, we employed a hash table. Consider our study cases; that is, the 6, 24, and 46-bus networks. The number of circuit expansion options for the largest of them (that is, the 46-bus network) is 491. Since we encode each expansion circuit option as a bit, an expansion plan would require 62 [Bytes] of memory plus 8 [Bytes] for its fitness value. With 1 [GB] of memory, more than 10 million expansion plans could be stored together with the fitness values. Therefore, we use a table (as a lookup mechanism) to avoid evaluating an expansion plan of known fitness more than once, and additionally to avoid duplicates in the population. Whenever the algorithm needs to evaluate the fitness of an individual, if the individual is not in the table, we compute its fitness and add a new entry for it; otherwise, we retrieve the fitness from the table. The genetic algorithm terminates either when a maximum number of generations has

---

19 Parallelising is really more about doing things independently than things happening at the same time. Here, by using $n$ processing units, it may give the illusion that $n$ evaluations are done (independently) in the time of one.

20 We rely for such purpose in the Multiprocessing Python libraries where the map and reduce operations are already implemented.

21 However, we are unlikely to do so because evaluating such number of expansion plans would take an amount of time much larger than the time we grant to the algorithm to terminate.

22 Another potential use of such a table is to implement a tabu list meta-heuristic (not done here).
been reached or when the fittest individual does not improve its fitness in a given number of generations; in the last case, we say the algorithm converged.

Both evolutionary optimisation algorithms (like genetic algorithms) and mixed-integer (linear) programming are active areas of research. So far in this chapter, we embedded in both MILP and genetic algorithms the LPAC (linear) power-flow constraints. We now describe the experiments we run in order to see what kind of solutions we can get by using the approximation. After that, we describe the machines we use to run the experiments. In the next chapter, we present the results.

The experiments are:

(1) **Evaluation on classic test systems.** - This is the baseline experiment that consists of evaluating the LPAC-TNEP solutions in terms of both expansion cost (i.e., objective function value) and quality (i.e., constraint violations under the AC model). For this purpose, we make an algorithmic comparison between DC-TNEP, LPAC-TNEP, and HAC-TNEP. DC-TNEP is the workable (more tractable) formulation used in order to attempt to find a solution to the problem, and HAC-TNEP provides a feasible AC-TNEP solution for comparison. We run HAC-TNEP until termination. And for both DC-TNEP and LPAC-TNEP, we force the algorithms to terminate after 2 hours.

We conduct, first, the experiment with a set $S$ of cardinality one. That is, as for the standard TNEP problem where the total (projected) demand of power (that is, $\sum_{n \in B} p^T_n$) cannot be satisfied by the current network topology (even with the increased generation capacity), and, as a consequence, new circuits must be added (that is to say, where the network needs to be expanded for a new scenario). We conduct, second, the experiment with a set $S$ defined by the $n - 1$ reliability criterion. Similarly, we first conduct the experiments with our study-case networks (as in an observational phase), and only if successful replicate the experiments using the test-set networks (as in a validation phase). The sequence described in this paragraph holds also for the other experiments.

(2) **TNEP with VAr compensation.** - In the previously discussed experiments, circuits are the only options available to expand the network and to satisfy the complex constraints. However, for example, tight voltage magnitude bounds may need a significant number of circuits in the expansion plan. As noted in previous work [19], VAr compensation equipment may also be installed throughout a network to satisfy voltage magnitude bounds constraints and it is cheaper to install than to install circuits options. We investigate the models in this experiment assuming unlimited VAr compensation at every bus; this can be modeled by transforming every bus into a synchronous condenser with unlimited reactive power injection capacity and a voltage set-point of 1.0. We call this profile ‘perfect voltage’, and generate accordingly perfect voltage profile versions of our networks.

---

23 Actually, some form of reactive power allocation is common to see when trying to solve this problem under the AC model. Witness our literature review.
The goal of this experiment is to study how the DC-TNEP and the LPAC-TNEP models behave under this profile; that is, to see what are the implications of VAr compensation and how cheaper a TNEP solution might be in this context. Note that for DC-TNEP, this has the same first step as in the previous experiment; it is only when converting the resulting expansion plan that the VAr compensation plays a role. In contrast, the LPAC-TNEP exploits VAr compensation in the first stage as well, since it models both reactive power and voltage magnitudes.

Furthermore, we note that the previous normal and perfect voltage profiles can be seen as special cases of a multi-objective optimisation problem. In a sense, the former assumes circuit options are cheap and VAr compensation is very expensive, and the latter assumes the reverse. In this experiment, we also enumerate some of the solutions along the Pareto frontier to understand the tradeoff between circuit expansion options and VAr compensation.

(3) **Power market considerations.**– As a final experiment with mixed-integer linear programs, we apply LPAC-TNEP in an economic model that was used to build four different competitive generation scenarios (named g1, g2, g3, and g4) by Fang and Hill [10]. Each scenario indicates the relative contribution of each generator to supply the required demand. We try LPAC-TNEP on them individually, and compare with a case gf where the dispatch takes part in the optimisation process.

The goal of this experiment is to study how the LPAC-TNEP model behaves under these networks; to see what are the implications of a competitive market and how much cheaper a TNEP solution might be in this context.

(4) **n − 1 reliability.**– The last experiment trying to exploit LPAC-TNEP is to use the genetic algorithm built on top of it in order to try to find AC feasible solutions that satisfy the set of scenarios defined by the n − 1 reliability criterion. As noted in previous work [33], when using a genetic algorithm, it is possible to separate the problem in independent LPs, and the computational effort of TNEP with reliability constraints in large systems could be leveraged by using parallel processing; this is the approach we take here that contrasts the typical way of iteratively adding circuits until the reliability criterion is satisfied.25

When using approximate models for TNEP, violations are expected and usually a second corrective stage is used to eliminate violations [38]. In our experiments, we propose an alternative approach called *constraint tightening*, where some of the model constraints are tightened a bit before starting the solution process. For example, reducing the thermal limits of the circuits by 10% may mitigate small

---

24 Of course, this was conditioned to observing good results in the evaluation of classic test systems for TNEP (which are good as shown in the next chapter).

25 We are limited to one node with 12 cores in our experiments. However, supposing this works, we know supercomputers exist, therefore we may scale the number of processing units (for instance, witness the IBM Blue Gene/Q hosted in Melbourne by VLSCI; a machine with 65536 cores and 65 TB of memory).
circuit loading violations and lead to AC-feasible solutions (note that reducing the circuits ratings is a strategy also mentioned in previous work [40]).

We run the described experiments in two types of machines:

(1) Dell PowerEdge 1950
   • CPU: 2 x 2.00 [GHz] Intel Quad Core Xeon E5405; 2x6 [MB] Cache.
   • Memory: 16 [GB].

(2) Dell PowerEdge R415
   • CPU: 2 x AMD 6-Core Opteron 4184, 2.8 [GHz]; 3/6 [MB] L2/L3 Cache.
   • Memory: 64 [GB].

where, roughly speaking, we use machine (2) primarily to run the map-reduce technique on 12 cores in the $n - 1$ reliability experiment.

To finalise this chapter, we remark that all the ‘magic’ in our implementation is possible largely due to the declarative nature of Python and its powerful libraries; in all of the experiments we use Gurobi as per its 5.5 version.\(^{26}\)

\(^{26}\)Note that when we are not using the cores in a map-reduce technique, we allow Gurobi to encompass 4 cores for the MIPs. However, as per the analysis of the automatic parameter tuner of Gurobi, this does not make a significant difference for this problem.
CHAPTER 8

Results, analysis, and interpretation

In this chapter, we report and discuss the results of the experiments described in the previous chapter, and in the next chapter we both explain our conclusions and suggest future work.

Once determined a solution configuration in any experiment, we first make note of the cost of the expansion plan prescribed, and then configure the network so that we can perform a run (to obtain an AC power-flow solution) and see whether the operational constraints are satisfied. If all of the operational constraints are satisfied, then we have evidence that (under the solution configuration found) the network operates normally (i.e., that the expanded network is feasible under the AC model); otherwise, we measure the operational constraints violations to judge the quality of the solution (that is, we judge the quality of a solution configuration in terms of both the thermal limit and the voltage bounds operational constraints violations exhibited by the expanded network). However, since the AC power-flow solutions are obtained using a method that is not guaranteed to converge, it can occur that a solution configuration cannot be evaluated; we call such configurations bad configurations, and note when this happens.

More concretely, the measures that we use to characterise the quality of a solution configuration are:

(1) the cost of the expansion plan (noting the number of circuit options involved in parentheses),
(2) the maximum and average violations on the relative thermal limit operational constraints (noting the number of overloaded circuits in parentheses),
(3) the maximum voltage bound operational constraints violation.1

When the sub-networks defined by the $n - 1$ reliability criterion are included, the measures are reported (for both circuit and buses operational constraints violations) among all of the scenarios.

---

1The numerical results for the evaluation on classic test systems, TNEP with VAr compensation, and power market considerations experiments presented in this chapter are presented in a similar fashion in Sections V, VI, and VII of our PSCC 2014 proceedings paper [2]; Tables 8.1, 8.2, 8.3, 8.4, and 8.5 and Figure 8.3 here are, respectively, Tables IV, V, VI, VII, and VIII and Figure 2 there. Furthermore, the positive results discussed herein are discussed in like manner in the mentioned paper in the respective sections.
1Note that measures (2) and (3) are not necessary for HAC-TNEP since an AC-OPF ensures the operational constraints hold.
Table 8.1. Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP.

<table>
<thead>
<tr>
<th>Case</th>
<th>Cost</th>
<th>DC-TNEP</th>
<th>Thermal limit violations</th>
<th>Voltage vio.</th>
<th>LPAC-TNEP</th>
<th>Thermal limit violations</th>
<th>Voltage vio.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Cost</td>
<td>Max. [%]</td>
<td>Mean [%]</td>
<td>Max. [V]</td>
<td>Cost</td>
<td>Max. [%]</td>
</tr>
<tr>
<td>6</td>
<td>160 (6)</td>
<td>110 (4)</td>
<td>37.32 (5)</td>
<td>21.90 (5)</td>
<td>0.13188</td>
<td>130 (5)</td>
<td>0.00 (0)</td>
</tr>
<tr>
<td>24</td>
<td>2310 (43)</td>
<td>152 (5)</td>
<td>AC-PF did not converge</td>
<td></td>
<td></td>
<td>689 (17)</td>
<td>01.02 (1)</td>
</tr>
<tr>
<td>46</td>
<td>569810 (47)</td>
<td>89889 (9)</td>
<td>AC-PF did not converge</td>
<td></td>
<td></td>
<td>310688 (29)</td>
<td>02.96 (1)</td>
</tr>
<tr>
<td>9</td>
<td>3 (3)</td>
<td>2 (2)</td>
<td>36.00 (5)</td>
<td>24.07 (5)</td>
<td>0.18994</td>
<td>2 (2)</td>
<td>04.45 (1)</td>
</tr>
<tr>
<td>14</td>
<td>15 (15)</td>
<td>1 (1)</td>
<td>20.44 (2)</td>
<td>16.17 (2)</td>
<td>0.11955</td>
<td>4 (4)</td>
<td>00.00 (0)</td>
</tr>
<tr>
<td>30</td>
<td>13 (13)</td>
<td>5 (5)</td>
<td>92.86 (16)</td>
<td>28.21 (16)</td>
<td>0.06847</td>
<td>10 (10)</td>
<td>10.84 (5)</td>
</tr>
<tr>
<td>39</td>
<td>47 (47)</td>
<td>26 (26)</td>
<td>13.46 (8)</td>
<td>08.21 (8)</td>
<td>0.08713</td>
<td>28 (28)</td>
<td>11.15 (2)</td>
</tr>
<tr>
<td>57</td>
<td>49 (49)</td>
<td>1 (1)</td>
<td>AC-PF did not converge</td>
<td></td>
<td></td>
<td>39 (39)</td>
<td>16.97 (13)</td>
</tr>
<tr>
<td>118</td>
<td>37 (37)</td>
<td>5 (5)</td>
<td>72.03 (17)</td>
<td>13.36 (17)</td>
<td>0.08198</td>
<td>7 (7)</td>
<td>16.97 (13)</td>
</tr>
</tbody>
</table>

1. Evaluation on classic test systems

- Table 8.1 shows the evaluation of the solutions obtained using all HAC-TNEP, DC-TNEP, and LPAC-TNEP in both the study-case and the test-case networks.

- Table 8.2 evaluates the quality of the HAC-TNEP, DC-TNEP and LPAC-TNEP solutions in both the study-case and the test-case networks by using constraint tightening at 10% (that is to say, by reducing the thermal limits of the circuits by 10% before starting the solution process). The curious reader can find analogous tables using constraint tightening at 5%, 15%, and 20% in Section 2 of Appendix C.

When expanding the set $S$ of scenarios to include the sub-networks defined by the $n - 1$ reliability criterion, it was not possible to find solution configurations because the MIPs failed to do so under the LPAC (linear) power-flow constraints (even using state-of-the-art industry standard tools). Note Oliveira et al. [25] have found (for a Venezuelan network of 134 buses and 271 circuits) that extending the mixed-integer disjunctive model under the DC linear power-flow constraints, to...
Table 8.2. Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP with constraint tightening in the thermal limit constraints at 10%.

<table>
<thead>
<tr>
<th>Case</th>
<th>DC-TNEP</th>
<th>HAC-TNEP</th>
<th>LPAC-TNEP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Thermal limit violations</td>
<td>Voltage vio.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cost</td>
<td>Cost</td>
<td>Max. [%]</td>
</tr>
<tr>
<td>6</td>
<td>160 (6)</td>
<td>130 (5)</td>
<td>05.91</td>
</tr>
<tr>
<td>24</td>
<td>2378 (44)</td>
<td>266 (8)</td>
<td>AC-PF did not converge</td>
</tr>
<tr>
<td>46</td>
<td>569810 (47)</td>
<td>130110 (13)</td>
<td>AC-PF did not converge</td>
</tr>
<tr>
<td>9</td>
<td>3 (3)</td>
<td>2 (2)</td>
<td>26.24</td>
</tr>
<tr>
<td>14</td>
<td>15 (15)</td>
<td>1 (1)</td>
<td>00.00</td>
</tr>
<tr>
<td>30</td>
<td>21 (21)</td>
<td>8 (8)</td>
<td>58.68</td>
</tr>
<tr>
<td>39</td>
<td>48 (48)</td>
<td>32 (32)</td>
<td>08.32</td>
</tr>
<tr>
<td>57</td>
<td>50 (50)</td>
<td>4 (4)</td>
<td>AC-PF did not converge</td>
</tr>
<tr>
<td>118</td>
<td>39 (39)</td>
<td>7 (7)</td>
<td>58.42</td>
</tr>
</tbody>
</table>

incorporate the \(n-1\) reliability criterion, can be successful when keeping the same number of discrete decision variables (as we do); our experimental evidence showed that this is not the case for the LPAC (linear) power-flow constraints.

- Figure 8.1 shows a more clear comparison of the expansion plans found by all HAC-TNEP, DC-TNEP, and LPAC-TNEP in terms of number of circuits.

Table 8.1 shows that there is a significant gap in the expansion investment cost between the TNEP solutions found by HAC-TNEP and the solutions found by DC-TNEP (confirming observations in previous work [3]). Such a gap can be bridged with the LPAC-TNEP model as illustrated in Figure 8.1 (constraint violations are discussed next).

Table 8.1 also shows that DC-TNEP solutions have significant violations to the original AC-TNEP constraints; in contrast, the LPAC-TNEP solutions have significantly less violations in both maximum and average values. The cost is higher, but still significantly smaller than those of the HAC-TNEP heuristic.

Table 8.2 shows that the constraint-tightening procedure makes marginal improvements in the violations in the DC-TNEP model, and, furthermore, that the
DC-TNEP solutions still have (major) issues with respect to AC power-flow feasibility (for example, some are bad configurations); in contrast, the constraint-tightening procedure has a great impact on the LPAC-TNEP solutions (that is to say, most of the thermal limit violations are eliminated).

Neither Table 8.1 nor Table 8.2 shows results incorporating the constraints for the $n-1$ sub-networks. It is known that when the number of decision variables and constraints keep increasing, a compromise needs to manifest at some point.\(^2\) For the LPAC (linear) power-flow constraints, such a compromise manifests already when incorporating in the model the constraints for the $n-1$ reliability criterion because the computational effort of the LP relaxation increases to a point where running the branch-and-bound algorithm is impractical.

Nevertheless, without incorporating the constraints for the $n-1$ sub-networks, the MIP does a good job overall; since it retains the benefits of linear programs, it should compare well with other alternative non-linear power-flow constraints.

### 2. TNEP with VAr compensation

- Table 8.3 shows the evaluation of the solutions obtained using all HAC-TNEP, DC-TNEP, and LPAC-TNEP in both the study-case and the test-case networks with perfect voltage profiles.

Since expansion costs are reduced with this approach, we compare the amount of active and reactive power (to have an idea on how much reactive power is) needed to keep this perfect voltage profile.

\(^2\)Recall that by using the LPAC (linear) power-flow constraints, the number of decision variables and constraints increase with the size of the network at a higher rate than by using the DC linear power-flow constraints.
Table 8.3. Solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP in the networks with perfect voltage profile.

<table>
<thead>
<tr>
<th>Case</th>
<th>HAC-TNEP</th>
<th>DC-TNEP</th>
<th>LPAC-TNEP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td></td>
<td>Max. [%]</td>
<td>Mean [%]</td>
<td>Max. [%]</td>
</tr>
<tr>
<td></td>
<td>Max. [V]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>130 (5)</td>
<td>110 (4)</td>
<td>0 (0)</td>
</tr>
<tr>
<td>24</td>
<td>573 (10)</td>
<td>152 (5)</td>
<td>0 (0)</td>
</tr>
<tr>
<td>46</td>
<td>277592 (22)</td>
<td>89889 (9)</td>
<td>0 (0)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>Cost</th>
<th>Cost</th>
<th>Cost</th>
<th>Cost</th>
<th>Cost</th>
<th>Cost</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max. [%]</td>
<td>Mean [%]</td>
<td>Max. [%]</td>
<td>Max. [V]</td>
<td>Max. [%]</td>
<td>Mean [%]</td>
<td>Max. [V]</td>
</tr>
<tr>
<td>6</td>
<td>0.00 (0)</td>
<td>0.00 (0)</td>
<td>0.00 (0)</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0.00 (0)</td>
<td>0.00 (0)</td>
<td>0.00 (0)</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>0.00 (0)</td>
<td>0.00 (0)</td>
<td>0.00 (0)</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8.4. Active and reactive power injection comparison for solutions to the networks with perfect voltage profile.

<table>
<thead>
<tr>
<th>Case</th>
<th>HAC-TNEP</th>
<th>DC-TNEP</th>
<th>LPAC-TNEP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Σπg∈Bp</td>
<td>Σπg∈Bq</td>
<td>Σπg∈Bp</td>
</tr>
<tr>
<td></td>
<td>[MW]</td>
<td>[MVAr]</td>
<td>[MW]</td>
</tr>
<tr>
<td>6</td>
<td>772.79</td>
<td>279.9</td>
<td>781.9</td>
</tr>
<tr>
<td>24</td>
<td>8819.93</td>
<td>4140.34</td>
<td>8802.53</td>
</tr>
<tr>
<td>46</td>
<td>7313.82</td>
<td>3618.02</td>
<td>7806.4</td>
</tr>
</tbody>
</table>

- Table 8.4 shows a comparison between the amount of active and reactive power injection needed in all of the networks to keep a perfect voltage profile.
8. RESULTS, ANALYSIS, AND INTERPRETATION

Figure 8.2. Reactive power injection needed to keep a perfect voltage profile

Table 8.5. Evaluation of solutions produced by LPAC-TNEP along the Pareto frontier for the 24-bus network with perfect voltage profile.

<table>
<thead>
<tr>
<th>λ</th>
<th>Cost [MVAr]</th>
<th>(\sum_{n \in B} q_n^{th})</th>
<th>Thermal limit violations</th>
<th>Voltage vio.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10834 (204)</td>
<td>231.82</td>
<td>Max. [%]</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>5609 (112)</td>
<td>1201.39</td>
<td>Max. [%]</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>2310 (51)</td>
<td>1856.36</td>
<td>Max. [%]</td>
<td>0</td>
</tr>
<tr>
<td>0.3</td>
<td>1642 (39)</td>
<td>2126.24</td>
<td>Max. [%]</td>
<td>0</td>
</tr>
<tr>
<td>0.4</td>
<td>1376 (34)</td>
<td>2270.73</td>
<td>Max. [%]</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>1214 (29)</td>
<td>2344.62</td>
<td>Max. [%]</td>
<td>0</td>
</tr>
<tr>
<td>0.6</td>
<td>910 (21)</td>
<td>2511.46</td>
<td>Max. [%]</td>
<td>0</td>
</tr>
<tr>
<td>0.7</td>
<td>745 (17)</td>
<td>2620.82</td>
<td>Max. [%]</td>
<td>0</td>
</tr>
<tr>
<td>0.8</td>
<td>645 (15)</td>
<td>2672.72</td>
<td>Max. [%]</td>
<td>0</td>
</tr>
<tr>
<td>0.9</td>
<td>573 (14)</td>
<td>2722.19</td>
<td>Max. [%]</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>509 (13)</td>
<td>2810.23</td>
<td>Max. [%]</td>
<td>0</td>
</tr>
</tbody>
</table>

• Figure 8.2 illustrates, more conveniently, the amount of reactive power needed.

To understand the tradeoff between circuit expansion and VAr compensation, we enumerated some of the solutions along the Pareto frontier.

• Table 8.5 evaluates some solution configurations obtained for the 24-bus instance, by using LPAC-TNEP in a joint circuit and VAr expansion optimisation; the classic TNEP formulation is captured by \(\lambda = 1\) and an extreme VAr minimization model is captured by \(\lambda = 0\).

• To illustrate further the tradeoff of joint circuit and VAr expansion, Figure 8.3 shows part of the Pareto frontier for the 6 and 24-bus networks, as found using the LPAC-TNEP model.
Figure 8.3. Pareto frontier solutions for the 6 (top) and 24-bus (bottom) networks produced by the LPAC-TNEP model.

Table 8.3 shows that, for the networks with a perfect voltage profile, LPAC-TNEP does it well in all of our networks exhibiting almost no violations (without constraint tightening). The gap between HAC-TNEP and DC-TNEP is much smaller than before; since the AC model integrates the nominal voltage assumption of the DC linear power-flow constraints, this is perhaps not surprising. However, despite this gap reduction, there are still significant violations in the DC-TNEP solutions.

The investment cost of circuit expansion is significantly reduced in the perfect voltage profile networks. However, the amount of reactive power injection required for this purpose is significant as shown in Figure 8.2. In the comparison made in Table 8.4, the total reactive power injection needed shows to be roughly half of the total active power injection. Depending on the cost of VAr compensation devices, it may be advantageous to co-optimise jointly circuit expansion (options) and VAr compensation costs.

When doing such a joint optimisation with the LPAC-TNEP, Table 8.5 shows that, for the 24-bus case, there is a huge range of network design possibilities
Table 8.6. Evaluation of LPAC-TNEP solutions produced for the $g_1$, $g_2$, $g_3$, $g_4$, and $g_f$ networks by using LPAC-TNEP with constraint tightening in the thermal limit constraints at 10%.

<table>
<thead>
<tr>
<th>Case</th>
<th>Cost</th>
<th>Thermal limit violations</th>
<th>Voltage vio.</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 $g_1$</td>
<td>1065 (25)</td>
<td>0.00 [0]</td>
<td>0.00 [0]</td>
</tr>
<tr>
<td>24 $g_2$</td>
<td>1171 (29)</td>
<td>0.00 [0]</td>
<td>0.00 [0]</td>
</tr>
<tr>
<td>24 $g_3$</td>
<td>1007 (21)</td>
<td>0.00 [0]</td>
<td>0.00 [0]</td>
</tr>
<tr>
<td>24 $g_4$</td>
<td>1084 (23)</td>
<td>0.00 [0]</td>
<td>0.00 [0]</td>
</tr>
<tr>
<td>24 $g_f$</td>
<td>705 (15)</td>
<td>0.00 [0]</td>
<td>0.00 [0]</td>
</tr>
</tbody>
</table>

Spanning from expansion plans with 13 new circuits to expansion plans with 200 new circuits (recall a scaling parameter $\lambda$ is used to balance the tradeoff between circuit expansion investment cost and VAr compensation).\(^3\) However, in all of these design possibilities, LPAC-TNEP is producing high-quality solutions with no violations at all.

The non-linear shape of both plots displayed in Figure 8.3 shows that the tradeoff of circuit expansions and VAr compensation is non-trivial thus the resulting network expansion plan may be heavily influenced by the cost models of circuit expansion and VAr compensation. Hence, it is best to optimise these two quantities jointly and possibly under different cost models.

3. Power market considerations

- Table 8.6 shows the evaluation for the solutions produced to the four different competitive generation scenarios $g_1$, $g_2$, $g_3$, and $g_4$, plus the case $g_f$ where the dispatch is part of the optimisation process.

Table 8.6 shows that, for this study of the $g_x$ networks \([10]\), the participation factors that constrain networks $g_1$, $g_2$, $g_3$, and $g_4$ to have a lower bound on active power generation $p_g$ can only increase the cost of the expansion plan (the last line in the table is for the equivalent network with no contribution factors; i.e., for $g_f$). The table illustrates that the investment cost of expansion can increase (roughly speaking) about 40% when generation is not flexible (a usually pessimistic view of the problem).

---

\(^3\)The avid reader will notice that Table 8.5 shows, for the 24-bus network, a cost of 509 (13) for $\lambda = 1$ while Table 8.3 shows a cost of 218 (6). Furthermore, $g_f$ in Table 8.6 shows a cost of 705 (15). First, the solutions displayed in Table 8.5 and Table 8.3 were obtained under different objectives. Recall the LPAC (linear) power-flow constraints need to maximise the sum of the cosine approximation decision variables to be accurate. In the latter, we do it after minimising the cost as in a lexicographic multi-objective function. In the former, we cannot do it like this because that would imply to change the VAr found by the joint optimisation. As a result, Gurobi arrived to different solution configurations. Second, the network $g_f$ is not the same network as the 24-bus network we use in the other experiments; the former is Fang and Hill’s version \([10]\) where we allow up to 4 circuit options per corridor and set to zero the lower bounds of generation capacity (that is to say, it is part of the $g_x$ networks).
4. $n-1$ RELIABILITY

Table 8.7. GA solutions to TNEP.

| Case | Cost | |S| | Max. [%] | Mean [%] | Nb of generations | Population size |
|------|------|---|---------|---------|----------|-----------------|-----------------|
| 6    | 130  | (5) | 1       | 0       | (0) 0    | 5               | 1604            |
| 24   | 1019 | (26)| 1       | 0       | (0) 0    | 49              | 1259            |
| 46   | 638633| (51)| 1       | 0       | (0) 0    | 28              | 1264            |

Table 8.8. GA solutions to TNEP with all $n-1$ sub-networks.

| Case | Cost  | |S| | Max. [%] | Mean [%] | Nb of generations | Population size |
|------|-------|---|---------|---------|----------|-----------------|-----------------|
| 6    | 200   | (8) | 9       | 0       | (0) 0    | 9               | 1667            |
| 24   | 1208  | (25)| 36      | 0       | (0) 0    | 30              | 1372            |
| 46   | Too computationally expensive to allow a few evolutions. |

4. $n-1$ reliability

- Tables 8.7 and 8.8 respectively display the results found with the genetic algorithm for TNEP and TNEP including the set of all $n-1$ sub-networks.

Unfortunately, there are situations where configurations approved by the metamodel are bad configurations; the genes of these bad configurations dominate generations and, as a result, the best configuration produced (that is, the configuration obtained when the algorithm terminates) cannot be evaluated under the AC model in all of the sub-networks. To avoid this behaviour, we used a filter to ensure that the best configuration in each generation can actually be validated under the AC model in all of the scenarios in the scenarios set. Once a new generation is produced, by using such a filter, we discard any bad configuration that holds the fittest position. This guarantees the best configuration obtained can be evaluated under the AC model in all of the scenarios. The advantage thereof is that the fitness function can be improved to take into account the information on circuit thermal limit violations under the AC model for the fittest individuals of each generation, and, as a result, solutions produced have absolutely no violations (thus require no further reinforcement). The results shown in both tables are those found under this scheme.

Table 8.8 shows that it is viable to find AC-feasible solutions that satisfy the $n-1$ reliability criterion by using the LPAC (linear) power-flow constraints, and lower the cost through a genetic algorithm. That is to say, the meta-model based on the LPAC (linear) power-flow constraints does a good job identifying potential $n-1$ AC feasible expansion plans (except for the bad configurations issue described previously which sabotages the optimisation algorithm). However, as Table 8.7 shows, the MIP approach seems to behave better for this problem; without the reliability constraints, the cost of the expansion plans involved in the solution configurations
is higher than those shown in Table 8.1. This gap can certainly be reduced by working on an instance basis.\footnote{It is legitimate to investigate a network in its own right instead of doing a case-study (the latter is of course more ambitious).} In fact, to persist with this experiment using the same algorithm, it seems reasonable to abandon the case study approach for a moment, to address the networks individually. Nevertheless, observe that the gap between the single scenario and the $n - 1$ expansion plan investment cost is not large, which is an encouraging result. Also note that by allowing up to 7 circuits per corridor and by adding all circuit options, it is possible to obtain $n - 1$ feasible solutions in these networks. From there, by taking such easy-to-get solutions as baseline, we can see significant improvements. That is, for the 6-bus network the investment cost improves from 4066 to 200, and for the 24-bus network improves from 12716 to 1028.

We remark to finalize this chapter, that all of those results discussed in this chapter that are of interest to the community, as declared in the Preface, were published in a peer-reviewed conference.
CHAPTER 9

Conclusion and future work

This study has explored the LPAC (linear) power-flow constraints to determine whether this approximation is a good mechanism to find low cost AC feasible expansion plans in the context of transmission network expansion planning with and without incorporating the $n - 1$ reliability criterion. Through a case-study, we have shown that TNEP solutions produced with the DC (linear) power-flow constraints both significantly underestimate the expansion investment costs and have significant violations under the AC power-flow model. In contrast, we have shown that, the TNEP solutions produced with the LPAC (linear) power-flow constraints have minimal operational constraints violations that, with a constraint tightening procedure, can often be eliminated. As a result, we confirmed the existence of a gap between the costs of TNEP solutions produced by DC (linear) power-flow constraints and AC power-flow heuristics, and demonstrated how to bridge this gap by using the LPAC (linear) power-flow constraints (which capture both reactive power and voltage root mean square values).

In addition, our case-study has shown that the LPAC (linear) power-flow constraints can be used for a joint optimisation of circuit expansion and VAr compensation. More concretely, the gap between DC-TNEP and AC heuristics solutions may be reduced with a two-stage VAr planning, but AC violations remain. In contrast, the LPAC-TNEP formulation leads to solutions with absolutely no operational constraints violations in the instances tested. The co-optimisation of circuit expansion and VAr compensation showed a delicate balance of both circuit expansions and VAr cost models on the resulting expansion plans. In the brief competitive market study, LPAC-TNEP showed that the expansion investment cost may increase to accommodate emerging energy markets. From these studies, we can see that the underlying TNEP formulation (e.g., including market considerations or adding VAr planning) can heavily influence the AC feasible expansion plans.

Furthermore, a LPAC-based meta-model was used to build on top of it a genetic algorithm; the meta-model has proven to identify good potential AC feasible expansion plans in all sub-networks of the scenarios set, and the optimisation algorithm

\footnote{A summary of the key findings itemised in this chapter appears in Section I of our PSCC 2014 proceedings paper [2]. Furthermore, the specific findings for the evaluation on classic test systems, TNEP with VAr compensation, and power market considerations experiments that are also itemised here are included, respectively, at the end of Sections V, VI, and VII of the mentioned paper. Moreover, Section VIII of the same paper presents the conclusions given in this (final) chapter leaving aside anything related to the $n - 1$ reliability criterion which was suggested as one possible way to extend the work presented there.}
was able to produce expansion plans for TNEP that lead to networks satisfying a proper $n-1$ reliability criterion. Such a meta-model may be used in other forms of heuristic approaches to the AC model as well.

Based on the above, it is reasonable to conclude that the LPAC (linear) power-flow constraints are a good mechanism for identifying expansion plans of reasonable cost that produce networks that are AC feasible, even for satisfying the $n-1$ reliability criterion. However, despite its drawbacks, the DC (linear) power-flow constraints are superior in efficiency. In the MIP context, even though the LPAC power-flow constraints are linear, and more accurate, its performance cannot compare with the performance of the LDC constraints\(^1\) based model.\(^2\)

To finalise this work, which is the first one that has taken into account the LPAC-TNEP formulation, we make a list with the key findings, list some observations we believe may be useful to anyone interested in continuing this work, and then suggest venues for future work. The key findings are:

- The expansion plans (as solutions to the TNEP problem) found with the DC (linear) power-flow constraints can significantly underestimate the investment cost of the expansion. Furthermore, when these expansion plans are evaluated under the AC model, they exhibit large violations on both thermal limits and voltage bounds operational constraints.
- Expansion plans (as solutions to the TNEP problem) can be produced with the LPAC (linear) power-flow constraints such that the expanded networks have little or no violations on the operational constraints. The investment cost implied by these expansion plans seem more reasonable than those found using the DC power-flow constraints (as observed in previous work).
- The previous findings hold when VAr (that is, reactive power) compensation is considered, including the case where each bus has an unlimited reactive power injection capacity.
- By using the LPAC (linear) power-flow constraints, the TNEP constrained optimisation problem can be formulated as a joint optimisation of circuit expansion and VAr compensation (these steps are typically separated in previous work). The resulting expansion plans exhibit no violations on the operational constraints; that is, they are AC feasible.
- LPAC-TNEP becomes computationally impractical when including in its model the reliability constraints as per the $n-1$ reliability criterion (even with state-of-the-art industrial-strength tools, the MIP algorithms seem unable to optimise for our study-case networks).
- It is viable to find AC feasible expansion plans that satisfy the $n-1$ reliability criterion by using a genetic algorithm for optimisation and a parallel LPAC-based meta-model for simulation.

\(^1\)The LDC constraints reduce to one single and small linear constraint per bus after all.

\(^2\)Unfortunately, we could not identify during this study any additional valid constraints that we could add into the constrained optimisation model to improve performance.
When solving with the genetic algorithm, it was observed that:

- Finding feasible solutions is computationally difficult; it is hard to get \( n - 1 \) LPAC-TNEP solution configurations by evolution from configurations created completely at random; if the initial population has no expansion plans that lead to \( n - 1 \) LPAC-TNEP solution configurations, the algorithm may struggle and never find one. The effects:
  - A large and diverse \( n - 1 \) LPAC-TNEP feasible configurations in the initial generation is paramount for the algorithm to produce AC feasible solutions (and consequently lower the cost). These could be created by testing for feasibility configurations generated at random.
  - It seems necessary to find a way of filtering out (or avoiding) bad configurations from the population as it evolves; otherwise, bad genes become dominant. Furthermore, the right approach seems to be maintain meta-model feasibility in all generations.
- Evaluating the potential feasibility of the sub-networks against the meta-model, concurrently with an underlying map-reduce technique, gives a (expected) speedup; but it may be too ambitious to aim to handle all of the sub-networks at the same time,\(^3\) given current computational limits.
- The two point crossover operation behaves much better than one point or multi-point or none crossover operations for this problem.
- Alternative mutation operations did not show any significant improvement for this problem in the case-study networks.

Directions for future work include:

- Since the MIP approach seems to behave better than the genetic algorithm (GA) approach, one may attempt to, instead of running LPs in parallel, run MIPs in parallel, incorporating a technique that allows shared expansion option decision variables, and optimise in such a way that by the end, all of the networks in the scenarios set are feasible under the same expansion plan. For example, there is a technique called *progressive hedging* (see, for instance, the references in Rendl’s ‘Stochastic MiniZinc’ paper \(^{28}\)).
- The GA seems to be a good approach, nevertheless, for seeking solutions that comply with the \( n - 1 \) reliability constraints. A variation of the GA, called CBGA, has been used before for TNEP with security constraints \(^{33}\), and it may worth its application to the LPAC power-flow constraints.\(^4\)
- An optimisation technique that seems popular in the power systems domain is particle swarm optimisation. One may attempt to explore such a technique (or other bio-inspired techniques) in this context.
- Future work may also incorporate other expansion options.

\(^3\)In this case, a single machine with 12 cores was not enough in our setup to allow a couple of generations in the evolution for the 46-bus network. Consequently, a distributed approach may be needed requiring a good extra deal of software engineering.

\(^4\)A parallel implementation of the GA may also worth to be tried (currently only the simulation component is parallelised).
• Another path is to abandon the LPAC approximation (and in a sense discrete optimisation) and explore convex relaxations. See, for a good start, Low’s ‘Convex Relaxation of Optimal Power Flow, Part I: Formulations and Equivalence’ paper [22].
Bibliography


Appendices
APPENDIX A

Constraints derivations

In this appendix, for reference, we first derive the non-linear AC power-flow constraints, and then derive both the LDC and the LPAC (linear) power-flow constraints. In other words, we first derive, from Ohm’s law, the non-linear AC power-flow constraints expressions that we aim to satisfy in the experiments, and then obtain from there (by using approximations) the linear expressions for both the LDC and the LPAC constraints.

1. Non-linear AC power-flow constraints

As per Ohm's law, the non-linear non-convex AC power-flow constraints ask that, for all bus \( n \in B \):

\[
S_n = E_n \cdot I_n^* \tag{72}
\]

However, since each bus can have more than one circuit connecting to it, such succinct expression necessarily considers the sum of the power entering or leaving all incident circuits to \( n \). A circuit \( n-m \) (that is, either a transmission line or a transformer connecting two buses \( n \) and \( m \)), has the following main properties (see Figure 2.2):

- the voltages at points \( n, m, p, \) and \( q \), respectively, \( E_n = \tilde{E}_n \angle \theta_n, \ E_m, \ E_p, \) and \( E_q \),
- the currents as seen at buses \( n \) and \( m \), respectively \( I_{nm} \) and \( I_{mn} \),
- the impedance \( Z_{nm} = r_{nm} + j \cdot x_{nm} \) that represents resistance and reactance,
- the shunt admittances \( Y_{nm}^s \) and \( Y_{mn}^s = \frac{1}{Z_{mn}^s} = g_{mn}^s + j \cdot b_{mn}^s \), expressed in terms of shunt conductance and susceptance, and
- the tap ratio for transformers at both sides of the circuit, respectively, \( T_{nm} = a_{nm} \angle \varphi_{nm} \) and \( T_{mn} \).

Furthermore, sometimes shunt elements are present not only in the circuits, but in the buses themselves. With all of these considerations (taking the conjugates for convenience) the power-flow constraints (in what is called their nodal formulation) ask that:

\[
\forall n \in B : S_n^* = E_n^* \cdot I_n = E_n^* \cdot \left( -I_n^{sh} + \sum_{n \sim m} I_{n \sim m} \right) \tag{73}
\]
\[ \begin{align*}
\ &= -E_n^* \cdot I_n^h + \sum_{n \sim m} E_n^* \cdot I_{n \sim m} \\
\ &= -E_n^* \cdot Y_n^h \cdot E_n^* + \sum_{n \sim m} S_{n \sim m}^*
\ &= E_n^* \cdot E_n \cdot Y_n^h + \sum_{n \sim m} (\bar{p}_{n \sim m} - j \cdot q_{n \sim m})
\ &= \tilde{e}_n^2 \cdot (g_n^h + j \cdot b_n^h) + \sum_{n \sim m} (\bar{p}_{n \sim m} - j \cdot q_{n \sim m})
\end{align*} \]

in other words, the constraints ask that:

\[(74) \quad \forall n \in B: S_n = (\tilde{e}_n^2 \cdot g_n^h - j \cdot \tilde{e}_n^2 \cdot b_n^h) + \sum_{n \sim m} (\bar{p}_{n \sim m} + j \cdot q_{n \sim m}) \]

or, equivalently, by separating real and imaginary components, the constraints ask both:

\[(75) \quad \bar{p}_n^d - \bar{p}_n^l = \tilde{e}_n^2 \cdot g_n^h + \sum_{n \sim m} \bar{p}_{n \sim m} \quad \text{and} \quad q_n^d - q_n^l = -\tilde{e}_n^2 \cdot b_n^h + \sum_{n \sim m} q_{n \sim m} \]

where \( n \sim m \) indexes a circuit in the set of all circuits \( n \sim m \) connected to bus \( n \),

- \( \bar{p}_{n \sim m} \) represents the active power in the circuit at the \( n \) end (similarly for \( \bar{p}_{mn} \)), and
- \( (\bar{p}_n^d - \bar{p}_n^l) \) represents the difference between active power generation \( \bar{p}_n^d \)
  and active power demand (or load) \( \bar{p}_n^l \) at bus \( n \), and

analogous for the reactive power \( q \).

In order to get the expressions for \( \bar{p}_{n \sim m} \) and \( q_{n \sim m} \) (that is to say, in order to get the components of \( S_n = E_n \cdot I_n^* \)) it is necessary to analyse both \( E_n \) and \( I_n \). We analyse first the voltage. When a transformer \( T_{nm} = a_{nm} \cdot e^{j \varphi_{nm}} \) is present (the transformer part of Figure 2.2 is made more explicitly in Figure A.2), it affects both the phase and magnitude components of both \( E_p \) and \( E_n \). The ratio is:

\[(76) \quad \frac{E_p}{E_n} = T_{nm} = a_{nm} \cdot e^{j \varphi_{nm}}; \therefore \theta_p = \theta_n + \varphi_{nm}, \text{ and } \tilde{e}_p = a_{nm} \cdot \tilde{e}_n. \]

For an in-phase transformer (that is to say, when \( \varphi_{nm} = 0 \)) the ratio \( \frac{E_p}{E_n} \) is \( a_{nm} \), as expected; furthermore, \( \frac{\tilde{e}_p}{\tilde{e}_n} = \frac{E_p}{E_n} \) since \( \theta_p = \theta_n \). Analysed the voltage, we analyse now the current. Since there are no power losses in an (ideal) transformer:

\[(77) \quad E_n \cdot I_{nm}^* + E_p \cdot I_{qp}^* = 0; \]

therefore:

\[(78) \quad \frac{I_{nm}^*}{I_{qp}^*} = -\frac{E_p}{E_n} = -T_{nm} = \left( \frac{I_{nm}}{I_{qp}} \right)^* \Rightarrow \frac{I_{nm}}{I_{qp}} = -T_{nm}^* = -a_{nm} \cdot e^{-j \varphi_{nm}} \]

Ignoring for a moment the shunt elements, we get:

\[(79) \quad I_{nm} = -T_{nm}^* \cdot I_{qp} = -T_{nm}^* \cdot (E_q - E_p) \cdot Y_{nm} \]

\[= T_{nm}^* \cdot (T_{nm} \cdot E_n - T_{nm} \cdot E_m) \cdot Y_{nm} \]
which yield the following symmetric expressions:

\[ I_{nm} = (a_{nm}^2 \cdot E_n - T_{nm}^* \cdot T_{mn} \cdot E_m) \cdot Y_{nm} \]

(81)

\[ I_{mn} = (a_{mn}^2 \cdot E_m - T_{mn}^* \cdot T_{nm} \cdot E_n) \cdot Y_{nm} \]

(82)

Adding similar terms for the shunt elements, we get the expressions for the currents \( I_{nm} \) and \( I_{mn} \) in Figure 2.2:

\[ I_{nm} = (a_{nm}^2 \cdot E_n - T_{nm}^* \cdot T_{mn} \cdot E_m) \cdot Y_{nm} + a_{nm}^2 \cdot E_n \cdot Y_{nm}^{sh} \]

(83)

\[ I_{mn} = (a_{mn}^2 \cdot E_m - T_{mn}^* \cdot T_{nm} \cdot E_n) \cdot Y_{nm} + a_{mn}^2 \cdot E_m \cdot Y_{nm}^{sh} \]

(84)

Knowing these general expressions for the currents, it is straightforward to derive the power for a circuit:

\[ S_{nm}^* = E_n^* \cdot I_{nm} \]

\[ = E_n^* \cdot [(a_{nm}^2 \cdot E_n - T_{nm}^* \cdot T_{mn} \cdot E_m) \cdot Y_{nm} + a_{nm}^2 \cdot E_n \cdot Y_{nm}^{sh}] \]

\[ = (a_{nm}^2 \cdot \tilde{E}_n - T_{nm}^* \cdot T_{mn} \cdot E_m) \cdot Y_{nm} + a_{nm}^2 \cdot \tilde{E}_n \cdot Y_{nm}^{sh} \]

\[ = (a_{nm} \cdot \tilde{E}_n)^2 \cdot (g_{nm} + j \cdot b_{nm}) + (g_{nm}^h + j \cdot b_{nm}^h) \]

\[ - (a_{nm} \cdot \tilde{E}_n) \cdot (a_{mn} \cdot \tilde{E}_m) \cdot \cos (\theta_{nm} + \varphi_{nm} - \varphi_{mn}) \]

\[ + j \cdot (b_{nm} \cdot \cos (\theta_{nm} + \varphi_{nm} - \varphi_{mn})) \]

\[ - j \cdot \sin (\theta_{nm} + \varphi_{nm} - \varphi_{mn}) \]

\[ - g_{nm} \cdot \sin (\theta_{nm} + \varphi_{nm} - \varphi_{mn}) \]

where \( \theta_{nm} = \theta_n - \theta_m \). Separating the real and imaginary parts, we finally get the expressions for the remaining terms of the non-linear power-flow constraints; that is, the expressions for the active and reactive power in a circuit \( n-m \):

\[ \tilde{P}_{nm} = (a_{nm} \cdot \tilde{E}_n)^2 \cdot g_{nm} \]

\[ - (a_{nm} \cdot \tilde{E}_n) \cdot (a_{mn} \cdot \tilde{E}_m) \cdot g_{nm} \cdot \cos (\theta_{nm} + \varphi_{nm} - \varphi_{mn}) \]

\[ - (a_{nm} \cdot \tilde{E}_n) \cdot (a_{mn} \cdot \tilde{E}_m) \cdot b_{nm} \cdot \sin (\theta_{nm} + \varphi_{nm} - \varphi_{mn}) \]

\[ q_{nm} = - (a_{nm} \cdot \tilde{E}_n)^2 \cdot (b_{nm} + b_{nm}^h) \]

\[ + (a_{nm} \cdot \tilde{E}_n) \cdot (a_{mn} \cdot \tilde{E}_m) \cdot b_{nm} \cdot \cos (\theta_{nm} + \varphi_{nm} - \varphi_{mn}) \]

\[ - (a_{nm} \cdot \tilde{E}_n) \cdot (a_{mn} \cdot \tilde{E}_m) \cdot g_{nm} \cdot \sin (\theta_{nm} + \varphi_{nm} - \varphi_{mn}) \]
that are in its general form.

For example, if the circuit is a transmission line (as shown in Figure A.1); that is, if $a_{nm} = a_{mn} = 1$ and $\varphi_{nm} = \varphi_{mn} = 0$, we get:

\[
\begin{align*}
(88) & \quad p_{nm} = g_{nm} \cdot \tilde{e}_n^2 - g_{nm} \cdot \tilde{e}_n \cdot \tilde{e}_m \cdot \cos \theta_{nm} - b_{nm} \cdot \tilde{e}_n \cdot \tilde{e}_m \cdot \sin \theta_{nm} \\
(89) & \quad q_{nm} = - (b_{nm} + b_{nm}^{sh}) \cdot \tilde{e}_n^2 - g_{nm} \cdot \tilde{e}_n \cdot \tilde{e}_m \cdot \sin \theta_{nm} + b_{nm} \cdot \tilde{e}_n \cdot \tilde{e}_m \cdot \cos \theta_{nm}
\end{align*}
\]

and expressions for $p_{mn}$ and $q_{mn}$ can be obtained in the same way:

\[
\begin{align*}
(90) & \quad p_{mn} = g_{mn} \cdot \tilde{e}_m^2 - g_{mn} \cdot \tilde{e}_n \cdot \tilde{e}_m \cdot \cos \theta_{nm} + b_{mn} \cdot \tilde{e}_n \cdot \tilde{e}_m \cdot \sin \theta_{nm} \\
(91) & \quad q_{mn} = - (b_{mn} + b_{mn}^{sh}) \cdot \tilde{e}_m^2 + g_{mn} \cdot \tilde{e}_n \cdot \tilde{e}_m \cdot \sin \theta_{nm} + b_{mn} \cdot \tilde{e}_n \cdot \tilde{e}_m \cdot \cos \theta_{nm}
\end{align*}
\]

For a transmission line, the active-power losses in a line are given by $\overline{p}_{nm} + \overline{p}_{mn}$. Therefore, from the previous:

\[
\begin{align*}
(92) & \quad \overline{p}_{nm} + \overline{p}_{mn} = g_{nm} \cdot (\tilde{e}_n^2 - 2 \cdot \tilde{e}_n \cdot \tilde{e}_m \cdot \cos \theta_{nm} + \tilde{e}_m^2) = g_{nm} \cdot |E_n - E_m|^2 \\
\end{align*}
\]

since:

\[
\begin{align*}
(93) & \quad |E_n - E_m|^2 = \Re(E_n - E_m)^2 + \Im(E_n - E_m)^2 \\
& \quad = \tilde{e}_n^2 - 2 \cdot \tilde{e}_n \cdot \tilde{e}_m \cdot \cos \theta_{nm} + \tilde{e}_m^2
\end{align*}
\]

Similarly, the reactive-power losses are given by:

\[
\begin{align*}
(94) & \quad q_{nm} + q_{mn} = - b_{nm}^{sh} \cdot (\tilde{e}_n^2 + \tilde{e}_m^2) - b_{nm} \cdot (\tilde{e}_n^2 - 2 \cdot \tilde{e}_n \cdot \tilde{e}_m \cdot \cos \theta_{nm} + \tilde{e}_m^2) \\
& = - b_{nm}^{sh} \cdot (\tilde{e}_n^2 + \tilde{e}_m^2) - b_{nm} \cdot |E_n - E_m|^2
\end{align*}
\]
The power-flow constraints are also straightforward to specialise from its general form to those for an in-phase transformer circuit (that is, for example, when \( a_{mn} = 1 \), \( \varphi_{mn} = \phi_{mn} = 0 \), and \( Y_{sh}^{hn} = Y_{sh}^{mn} = 0 \), or to those for a phase-shifting transformer circuit (that is, for example, when \( a_{mn} = 1 \), \( \varphi_{mn} = 0 \), and \( Y_{sh}^{hn} = Y_{sh}^{mn} = 0 \)). See Figure A.2.

Finally, getting back to the power-flow constraints for a circuit, for convenience, we define \( b_{sh}^{hn} \) as \( \frac{b_{cm}}{2} \) and \( a_{nm} \) as \( \frac{1}{t_{nm}} \), and, furthermore, consider one tap ratio at one side of the circuit only (that is, \( \varphi_{mn} = \varphi_{nm} = 0 \) and \( a_{mn} = 1 \)), so that the general power-flow constraints reduce to:

\[
\begin{align*}
\tilde{p}_{nm} &= \frac{\tilde{e}_n^2}{t_{nm}} \cdot g_{nm} - \frac{\tilde{e}_n \cdot \tilde{e}_m}{t_{nm}} \cdot (g_{nm} \cdot \cos \theta_{nm} + b_{nm} \cdot \sin \theta_{nm}) \\
q_{nm} &= -\frac{\tilde{e}_n^2}{t_{nm}} \cdot \left( b_{nm} + \frac{b_{cm}}{2} \right) + \frac{\tilde{e}_m \cdot \tilde{e}_n}{t_{nm}} \cdot (g_{nm} \cdot \sin \theta_{nm} - b_{nm} \cdot \cos \theta_{nm})
\end{align*}
\]

which are the realistic \(^1\) asymmetric non-linear non-convex power-flow constraints for large scale power systems that we aim to satisfy in the experiments.

2. Linear AC power-flow constraints

The linear AC power-flow constraints that we present in this section are linear approximations derived from the non-linear AC power-flow constraints obtained in the previous section. They are attractive due to their complexity.\(^2\) The two approximations derived herein are: the popular Linearised DC (or LDC) power-flow constraints\(^3\) and the more accurate LPAC power-flow constraints.

2.1. LDC power-flow constraints. The simplest (and most popular) variant of the Linearized DC (LDC) power-flow constraints is derived from the non-linear AC power-flow constraints by using several approximations justified by the following assumptions:

1. The susceptance of the circuits is large with respect to their conductance.
2. The phase angle difference across a circuit is small.
3. The voltage magnitude root mean square values, assumed close to unity in all buses, do not vary significantly.
4. Ignore transformers. Ignore line charging.

\(^1\)Note the bibliography is located before the appendices part. We kindly ask the reader to look back to find the citations (since there are quite few references in the appendices, we did not replicate them at the end of the appendices part).

\(^2\)Solutions to linear models can many times be computed efficiently with non-iterative and absolutely convergent algorithms.

\(^3\)Please, do not get confused with its name since the ‘DC’ power-flow constraints, regardless of its name, approximates the ‘AC’ power-flow constraints (notice it involves phase angles).
A. CONSTRAINTS DERIVATIONS

Under these assumptions, by using the following approximations:

1. Take all $g_{nm} \approx 0$, since $|g_{nm}| \ll |b_{nm}|$.
2. Take all $\sin \theta_{nm} \approx \theta_{nm}$ and $\cos \theta_{nm} \approx 1.0$, since $\theta_n - \theta_m = \theta_{nm} \approx 0$.
3. Take $\tilde{\theta} \approx 1.0$, at all buses.
4. All $t_{nm} = 1$ and $b_{nm} = 0$.

The non-linear AC power-flow constraints derived in the previous section:

(97) \[ \bar{p}_{nm} = \frac{\tilde{e}^2_m}{t_{nm}^2} \cdot g_{nm} - \frac{\tilde{e}_n \cdot \tilde{e}_m}{t_{nm}} \cdot (g_{nm} \cdot \cos \theta_{nm} + b_{nm} \cdot \sin \theta_{nm}) \]
(98) \[ q_{nm} = -\frac{\tilde{e}^2_n}{t_{nm}^2} \cdot \left( b_{nm} + \frac{b_{nm}^2}{2} \right) - \frac{\tilde{e}_n \cdot \tilde{e}_m}{t_{nm}} \cdot (g_{nm} \cdot \sin \theta_{nm} - b_{nm} \cdot \cos \theta_{nm}) \]
(99) \[ \bar{p}_{mn} = \tilde{e}_m^2 \cdot g_{nm} - \frac{\tilde{e}_m \cdot \tilde{e}_n}{t_{nm}} \cdot (g_{nm} \cdot \cos \theta_{mn} + b_{nm} \cdot \sin \theta_{mn}) \]
(100) \[ q_{mn} = -\tilde{e}_m^2 \cdot \left( b_{mn} + \frac{b_{mn}^2}{2} \right) + \frac{\tilde{e}_m \cdot \tilde{e}_n}{t_{nm}} \cdot (b_{nm} \cdot \cos \theta_{mn} - g_{nm} \cdot \sin \theta_{mn}) \]

become, respectively:

(101) \[ \bar{p}_{nm} = -b_{nm} \cdot \theta_{nm} \]
(102) \[ q_{nm} = -b_{nm} + b_{nm} = 0 \]
(103) \[ \bar{p}_{mn} = -b_{nm} \cdot \theta_{mn} = b_{nm} \cdot \theta_{nm} \]
(104) \[ q_{mn} = -b_{nm} + b_{nm} = 0 \]

which means $\bar{p}_{nm} = -\bar{p}_{mn}$ and $q_{nm} = q_{mn} = 0$. The symmetric formulation:

(105) \[ \bar{p}_{nm} = -b_{nm} \cdot \theta_{nm} \quad \text{where} \quad \theta_{nm} = \theta_n - \theta_m \]

is used in many frameworks for decision support in power systems.

2.2. LPAC power-flow constraints. The LPAC power-flow constraints [9] is a more accurate (yet linear) approximation to the non-linear AC power-flow constraints. It relies on a polyhedral relaxation to approximate the cos function (see [9] for details), denoted by $\cos$, and assumes that the phase angle difference across a circuit is small enough to approximate the sin function as $\sin \theta_{nm} \approx \theta_{nm}$. It can take target root mean square voltage values, denoted by $\tilde{V}$ (which, for a ‘cold start’, can be set to 1.0 mimicking what is done for the LDC approximation), and defines $\tilde{V} = \tilde{v} + \phi$ where $\phi$ is a deviation in the target voltage magnitude. Observe:

(106) \[ \tilde{V}_n^2 = \left( \tilde{v}_n + \phi_n \right)^2 = \tilde{v}_n^2 + 2 \cdot \tilde{v}_n \cdot \phi_n + \phi_n^2 \quad \text{and} \]
(107) \[ \tilde{v}_n \cdot \tilde{v}_m = \left( \tilde{v}_n + \phi_n \right) \cdot \left( \tilde{v}_m + \phi_m \right) = \tilde{v}_n^2 + \tilde{v}_n \cdot \phi_m + \tilde{v}_m \cdot \phi_n + \phi_n \cdot \phi_m. \]

The AC active power non-linear constraints:

(108) \[ \bar{p}_{nm} = \frac{\tilde{e}_m^2}{t_{nm}^2} \cdot g_{nm} - \frac{\tilde{e}_m \cdot \tilde{e}_n}{t_{nm}} \cdot (g_{nm} \cdot \cos \theta_{nm} + b_{nm} \cdot \sin \theta_{nm}) \]
(109) \[ \bar{p}_{mn} = \tilde{e}_m^2 \cdot g_{nm} - \frac{\tilde{e}_m \cdot \tilde{e}_n}{t_{nm}} \cdot (g_{nm} \cdot \cos \theta_{mn} + b_{nm} \cdot \sin \theta_{mn}) \]
are approximated using target voltages, so they become:

\[\tilde{p}_{nm} = \frac{\tilde{e}_n^2}{t_{nm}} \cdot g_{nm} - \frac{\tilde{e}_n \cdot \tilde{e}_m}{t_{nm}} \cdot (g_{nm} \cdot \cos \theta_{nm} + b_{nm} \cdot \theta_{nm})\]

\[\tilde{q}_{nm} = \frac{\tilde{e}_m^2}{t_{nm}} \cdot g_{nm} - \frac{\tilde{e}_m \cdot \tilde{e}_n}{t_{nm}} \cdot (g_{nm} \cdot \cos \theta_{nm} + b_{nm} \cdot \theta_{mn})\]

The AC reactive power non-linear constraints:

\[q_{nm} = -\frac{\tilde{e}_m^2}{t_{nm}} \cdot \left(b_{nm} + \frac{b_{nm}^c}{2}\right) - \frac{\tilde{e}_m \cdot \tilde{e}_n}{t_{nm}} \cdot (g_{nm} \cdot \sin \theta_{nm} - b_{nm} \cdot \cos \theta_{nm})\]

\[q_{mn} = -\frac{\tilde{e}_n^2}{t_{nm}} \cdot \left(b_{nm} + \frac{b_{nm}^c}{2}\right) + \frac{\tilde{e}_n \cdot \tilde{e}_m}{t_{nm}} \cdot (b_{nm} \cdot \cos \theta_{mn} - g_{nm} \cdot \sin \theta_{mn})\]

are approximated by using both target voltage magnitudes and target voltage magnitudes deviations since voltage magnitudes are the primary factor in reactive power-flows,\(^4\) so they become:

\[q_{nm} = -\frac{\tilde{e}_n^2}{t_{nm}} + \frac{2 \cdot \tilde{e}_n \cdot \phi_n + \phi_n^2}{t_{nm}^2} \cdot \left(b_{nm} + \frac{b_{nm}^c}{2}\right) - \frac{\tilde{e}_n \cdot \tilde{e}_m + \tilde{e}_m \cdot \phi_n + \phi_n \cdot \phi_m}{t_{nm}} \cdot (g_{nm} \cdot \theta_{nm} - b_{nm} \cdot \cos \theta_{nm})\]

\[q_{mn} = -\frac{\tilde{e}_m^2}{t_{nm}} + \frac{2 \cdot \tilde{e}_m \cdot \phi_m + \phi_m^2}{t_{nm}^2} \cdot \left(b_{nm} + \frac{b_{nm}^c}{2}\right) + \frac{\tilde{e}_m \cdot \tilde{e}_n + \tilde{e}_n \cdot \phi_m + \phi_m \cdot \phi_n}{t_{nm}} \cdot (b_{nm} \cdot \cos \theta_{mn} - g_{nm} \cdot \theta_{mn})\]

Defining \(q_{nm}^\Delta\) and \(q_{mn}^\Delta\) (so that \(q_{nm} = q_{nm}^t + q_{nm}^\Delta\) and \(q_{mn} = q_{mn}^t + q_{mn}^\Delta\)) as:

\[q_{nm}^\Delta = -\frac{2 \cdot \tilde{e}_n \cdot \phi_n + \phi_n^2}{t_{nm}^2} \cdot \left(b_{nm} + \frac{b_{nm}^c}{2}\right) - \frac{\tilde{e}_n \cdot \tilde{e}_m + \tilde{e}_m \cdot \phi_n + \phi_n \cdot \phi_m}{t_{nm}} \cdot (g_{nm} \cdot \theta_{nm} - b_{nm} \cdot \cos \theta_{nm})\]

\[q_{mn}^\Delta = -\frac{2 \cdot \tilde{e}_m \cdot \phi_m + \phi_m^2}{t_{nm}^2} \cdot \left(b_{nm} + \frac{b_{nm}^c}{2}\right) + \frac{\tilde{e}_m \cdot \tilde{e}_n + \tilde{e}_n \cdot \phi_m + \phi_m \cdot \phi_n}{t_{nm}} \cdot (b_{nm} \cdot \cos \theta_{mn} - g_{nm} \cdot \theta_{mn})\]

and approximating these by using the linear terms of the Taylor series, they become:\(^4\)

\[q_{nm}^\Delta = -\frac{2 \cdot \tilde{e}_n \cdot \phi_n}{t_{nm}^2} \cdot \left(b_{nm} + \frac{b_{nm}^c}{2}\right) + \frac{\tilde{e}_n \cdot \phi_m + \phi_m \cdot \phi_n}{t_{nm}} \cdot b_{nm}\]

\[q_{mn}^\Delta = -\frac{2 \cdot \tilde{e}_m \cdot \phi_m}{t_{nm}^2} \cdot \left(b_{nm} + \frac{b_{nm}^c}{2}\right) + \frac{\tilde{e}_m \cdot \phi_n + \phi_n \cdot \phi_m}{t_{nm}} \cdot b_{nm}\]

which are linear. Finally, as per the cold-start LPAC model \(^9\), by setting \(\tilde{e}_n\) and \(\tilde{e}_m\) to unity\(^5\) we get the following asymmetric and linear constraints:

\(^4\)For more about this (and all in this section), see the LPAC paper \(^9\).

\(^5\)Note that by ignoring transformers and shunt elements the LPAC constraints reduce to:

\[\tilde{p}_{nm} = g_{nm} - g_{nm} \cdot \cos \theta_{nm} - b_{nm} \cdot \theta_{nm}\]

\[q_{nm} = -b_{nm} + b_{nm} \cdot \cos \theta_{mn} - g_{nm} \cdot \theta_{nm} - b_{nm} \cdot (\phi_n - \phi_m)\]

as displayed in Subsection III.C of our PSCC 2014 proceedings paper \(^2\).
Figure A.3. Non-linear and LPAC (linear) power-flow constraints correlation plots for the MatPower 14-bus OPF benchmark.

\[
\begin{align*}
\mathbf{p}_{nm} &= \frac{1}{2} \cdot \frac{1}{t_{nm}} \cdot g_{nm} - \frac{1}{2} \cdot \frac{1}{t_{nm}} \cdot g_{nm} \cdot \cos \theta_{nm} - \frac{1}{2} \cdot \frac{1}{t_{nm}} \cdot b_{nm} \cdot \theta_{nm} \\
\mathbf{q}_{nm} &= g_{nm} - \frac{1}{2} \cdot \frac{1}{t_{nm}} \cdot g_{nm} \cdot \cos \theta_{nm} - \frac{1}{2} \cdot \frac{1}{t_{nm}} \cdot b_{nm} \cdot \theta_{nm} \\
\mathbf{q}_{nm}^+ &= \frac{1}{2} \cdot \frac{1}{t_{nm}} \cdot \left( b_{nm} + \frac{b_{cm}}{2} \right) - \frac{1}{2} \cdot \frac{1}{t_{nm}} \cdot g_{nm} \cdot \theta_{nm} + \frac{1}{2} \cdot \frac{1}{t_{nm}} \cdot b_{nm} \cdot \cos \theta_{nm} \\
\mathbf{q}_{nm}^- &= \frac{2 \cdot \phi_n}{2} \cdot \left( b_{nm} + \frac{b_{cm}}{2} \right) + \frac{\phi_m + \phi_n}{t_{nm}} \cdot b_{nm} \\
\mathbf{q}_{nm} \Delta &= \frac{2 \cdot \phi_n}{2} \cdot \left( b_{nm} + \frac{b_{cm}}{2} \right) + \frac{\phi_m + \phi_n}{t_{nm}} \cdot b_{nm} \\
\mathbf{q}_{n}^+ &= \frac{1}{2} \cdot \frac{1}{t_{nm}} \cdot \left( b_{nm} + \frac{b_{cm}}{2} \right) - \frac{1}{2} \cdot \frac{1}{t_{nm}} \cdot g_{nm} \cdot \theta_{nm} \\
\mathbf{q}_{n}^- &= -\frac{2 \cdot \phi_m}{2} \cdot \left( b_{nm} + \frac{b_{cm}}{2} \right) + \frac{\phi_m + \phi_n}{t_{nm}} \cdot b_{nm} \\
\mathbf{q}_{n} \Delta &= -\frac{2 \cdot \phi_m}{2} \cdot \left( b_{nm} + \frac{b_{cm}}{2} \right) + \frac{\phi_m + \phi_n}{t_{nm}} \cdot b_{nm}
\end{align*}
\]
However, to get the best approximation the linear program needs to maximise the sum of the cosine relaxation decision variables; therefore, the linear formulation of the LPAC power-flow constraints that we use in the experiments become:

\[
\begin{align*}
\text{(126)} & \quad \max \sum_{n \in B} \hat{c} \cos \theta_{nm} \\
\text{subject to}, \forall n & \in B : \\
\text{(127)} & \quad \bar{P}_n = (1 + 2 \cdot \phi_n) \cdot g_n^b + \sum_{n \sim m} \bar{P}_{n \sim m} \\
\text{(128)} & \quad q_n = -(1 + 2 \cdot \phi_n) \cdot b_n^b + \sum_{n \sim m} (q_{n \sim m}^\Delta + q_{n \sim m}^\Delta)
\end{align*}
\]
2.2.1. Correlation plots. For reference, Figures A.3 and A.4 (together with Figure 5.2) give correlation plots that correlate the power and voltage values obtained with our implementation of the LPAC (linear) power-flow constraints vs. those obtained with PyPower’s implementation of the (non-linear) AC power-flow constraints (that is based on Newton’s method)\(^6\). The mentioned figures show correlation plots for the 14 and 57-bus networks (together with the 24-bus network) MatPower OPF benchmarks, respectively.

\(^6\)Newton’s method is also known as the ‘Newton-Raphson method’. The method was first named after Isaac Newton (Lincolnshire 1643 - 1727 London), but it was later discovered that Joseph Raphson (Middlesex 1648 - 1715) independently explained Newton’s method in his book ‘Analysis Aequationum Universalis’ published nearly 50 years before Newton’s ‘Method of Fluxions’ book.
APPENDIX B

Study and test cases

The case-study networks are the three most popular TNEP cases from the literature: networks with 6, 24, and 46 buses that can be found, e.g., in papers [13], [10], and [14] respectively. Since there are some variations of these networks in the literature, this appendix reviews our versions in detail. Our 6-bus case is the network from paper [29] with bus 6 as the slack bus and $0.95 \leq \tilde{e}_n \leq 1.05$. Our 24-bus case is the network from MatPower’s distribution case files with investment cost data from paper [29], $b_{c6-10}/100$, and the load increasing strategy discussed in the next paragraph. Our 46-bus case is the network from paper [14] with: bus 16 as the slack bus, generators’ reactive injection capacity set to half of the active capacity, and circuit resistances assigned to one fifth of the respective reactances.

To understand the TNEP problem on a wider collection of networks, we designed the following procedure to generate TNEP instances from any MatPower OPF test case. Both the loads and the capacity of generating units are scaled by a factor of 3 (except synchronous condensers), and the reactive power injection capacity of all generators is set to half of the active power injection capacity (that is, $q_n^g = -0.5 \cdot \tilde{p}_n$ and $\tilde{q}_n^g = 0.5 \cdot \tilde{p}_n$). For circuits with a thermal limit set to 9900 [MVA], we use the value of the circuit loaded at a phase angle difference of 15 degrees instead. Finally, the investment cost of adding each circuit is set to 1 (i.e., the goal becomes to minimize the number of circuits added to the expansion plan). This procedure is used to build the 9, 14, 30, 39, 57, and 118-bus cases from the MatPower networks (that is to say, to build the test cases or test-set). Finally, since the voltage constraints dominate the 57-bus network, we widen the bounds to $0.9 \leq \tilde{e}_n \leq 1.1$ in order to make the optimisation task interesting. In all of our experiments, expansions can select up to 6 circuits per corridor.

For the reliability problem, we further expand these cases by creating a set of $n$ scenarios for each of them that includes both the $n-1$ sub-networks (which are obtained by eliminating one single circuit from the base network) and the (projected) base network. Furthermore, we widen in all of the sub-networks the voltage magnitude bounds by 10% of its original value.

Finally, note that each case usually specifies (for a circuit) both the resistance and reactance values (that is, both $r_{nm}$ and $x_{nm}$) from which both susceptance and admittance can be determined. For the shunt admittances, what is usually specified is a value $b^{sh}$ (that we call circuit charging), and we assume $b^{sh} = b_{nm} = b_{mn}^{sh}$. For

†The case-study and test-case networks reviewed in this appendix are reviewed in a similar fashion in Section V of our PSCC 2014 proceedings paper [2].
a transformer, usually the tap ratio is given in the form \( a = a_{nm}^{-1} \angle -\varphi_{nm} \), and its conversion to \( t \) is straightforward since \( t = \frac{1}{a} \) (taking care when \( a \) is zero).

The \( \text{gx} \) test cases are networks taken from paper [10].
APPENDIX C

Papers and a few more tables

In this appendix we attach two papers:


(2) “Reliable power transmission networks” – presented at The University of Melbourne CIS 2014 Doctoral Colloquium

In addition, for the curious reader, we attach a few more tables showing the impact of the constraint tightening procedure (in the thermal limit constraints) on the operational-constraints violations of the solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP. Tables C.1, C.2, C.3, C.4, and C.5 evaluate the produced solutions (by all of these methods) with constraint tightening at 0%, 5%, 10%, 15%, and 20% respectively.

1. Papers


The remainder of this page is intentionally left blank.
Transmission Network Expansion Planning: 
Bridging the Gap between AC Heuristics and DC Approximations

Russell Bent*, Carleton Coffrin†, Rodrigo R. Gumucio E., and Pascal Van Hentenryck†

*Los Alamos National Laboratory, Los Alamos, NM, U.S.A.
†NICTA & The University of Melbourne, Melbourne, VIC, Australia
‡NICTA & The Australian National University, Canberra, ACT, Australia

Abstract—It was recently observed that a significant gap exists between the costs of Transmission Network Expansion Planning (TNEP) solutions produced by the DC power flow approximation and AC power flow heuristics. This paper confirms the existence of that gap and shows that DC-based TNEP solutions exhibit significant constraint violations when converted into AC power flows. The paper then demonstrates how to bridge this gap with the LPAC power flow model, a linear-programming approximation of the power flow that captures reactive power and voltage magnitudes. Indeed, LPAC-based TNEP solutions have minimal violations in AC power flows and provide high-quality solutions. The strength of the LPAC formulation is further demonstrated on the joint optimization of line expansion and VAR compensation, as well as a competitive market study. These studies suggest that the underlying TNEP formulation has significant impacts on the proposed expansion plans.


I. INTRODUCTION

Transmission Network Expansion Planning (TNEP) is a well-studied optimization problem which consists of finding the least expensive way of increasing the capacity of a transmission network to meet some projected future energy delivery requirements. Due to its computational complexity, TNEP problems have been traditionally studied with active-power-only approximations of the transmission system [1]–[5]. Such approximations are appealing as they yield Mixed Integer Linear Programs (MIPs), which exploit decades of research in network design optimization and existing commercial tools [6].

Many TNEP studies have adopted the widely-used DC power flow model. However, its applicability for power flow applications is a point of significant discussion: Some papers take an optimistic outlook, (e.g., [7], [8]) while others are more pessimistic (e.g., [9], [10]). For TNEP applications, some major shortcomings of DC model were identified in [11]. Recognizing these potential limitations, recent work has started to consider the TNEP problem with the complete AC power flow equations [11]–[16].

This paper explores the use of the LPAC model [17] for transmission planning. The LPAC model was proposed recently for approximating the AC power flow equations. Contrary to the DC model, the LPAC approximation captures both reactive power and voltage magnitudes; Yet it is linear program which is highly desirable computationally. The LPAC model thus provides an appealing tradeoff between computational efficiency and solution accuracy: It was instrumental in identifying new best solutions on existing test cases and bridging the significant gap between DC-based models and AC heuristics on transmission planning. The key findings in this paper are summarized as follows:

1) DC-based transmission planning significantly underestimates the cost of expansion; The resulting plans, when converted into an AC plan, exhibits large violations of line capacities and voltage bounds.
2) LPAC-based transmission planning provides AC plans with little or no violations and more reasonable expansion costs; Yet the use of the LPAC model may significantly improve the quality of AC-based heuristics.
3) The results in (1) and (2) still hold when VAR compensation is considered, including in the case where each bus has unlimited reactive power injection.
4) The LPAC model can be used for the joint optimization of line and VAR compensation costs, steps which are typically separated in prior work.
The AC-TNEP formulation is given by: 

\[
\begin{align*}
\min \quad & \sum_{n,m \in L} c_{mn} \left( x_{mn} - 1 \right) + \sum_{n,m \in L} c_{mn} \bar{x}_{mn} \\
\text{s.t.} \quad & \left[ \begin{array}{l}
V_{mn} \\
V_{mn}^* \\
\phi_{mn} \\
\phi_{mn}^* \\
p_{mn} \\
p_{mn}^* \\
q_{mn} \\
q_{mn}^* \\
p_{mn}^{out} \\
p_{mn}^{out}^* \\
q_{mn}^{out} \\
q_{mn}^{out}^*
\end{array} \right] \in N \times N \\
\left| V_{mn} \right| \leq \left| V_{mn}^* \right|, \quad \forall n \in N \\
\left| V_{mn}^* \right| = 0 \\
\phi_{mn} - \phi_{mn}^* = \sum_{n \in N} \varpi_{mn} \phi_{mn}, \quad \forall n \in N \\
\phi_{mn}^* - \phi_{mn} = \sum_{n \in N} \varpi_{mn} \phi_{mn}, \quad \forall n \in N \\
1 \leq \bar{x}_{mn}, \quad \forall (n,m) \in L \\
\left| \bar{V}_{mn} \right| = 0, \quad \forall (n,m) \in L \\
\left| \bar{V}_{mn} \right| \leq \left| \bar{V}_{mn}^* \right| \\
p_{mn}^{out} = g_{mn} \left( \bar{V}_{mn} - g_{mn} \bar{V}_{mn}^* \right) - h_{mn} \bar{V}_{mn} \\
q_{mn}^{out} = -h_{mn} \left( \bar{V}_{mn}^* - g_{mn} \bar{V}_{mn} \right) - h_{mn} \bar{V}_{mn} \\
p_{mn}^{out} + q_{mn}^{out} \leq \bar{x}_{mn}
\end{align*}
\]

Additionally, the paper also suggests that, given the LPAC models accuracy, it may be informative to revisit previous TNEP studies. This point is illustrated on a simple case study about the implications of competitive markets for the TNEP.

The rest of the paper is organized as follows. Section II introduces the TNEP problem and the notations. Section III discusses potential solution methods, including prior art, and uses a 3-bus example to illustrate the tradeoffs of the approaches. The next three sections conduct the case studies: Section V establishes the baseline by considering the classic TNEP formulation; Section VI considers the implications of Var compensation; and Section VII studies the implications of a competitive market. Section VIII concludes the paper.

The AC-TNEP problem presented in Model 1 (and its extensions) are challenging Mixed Integer Non-Convex Non-Linear Programs (MINLP) and are outside the scope of current global optimization solvers. As a result, one must resort to solving alternative, computationally tractable versions of the problem. There are three main approaches to making the AC-TNEP more tractable: (1) using heuristics [11], [14], [15]; (2) approximating the power flow equations [1], [17]; (3) relaxing the power flow equations to a convex set [12], [13]. Heuristics are often very fast to compute, but they provide no quality guarantees. Approximations can alter the computational complexity (e.g., moving from a MINLP to a MILP) and provide quality guarantees within the confines of the approximation. In practice, approximations may be sufficiently accurate to provide feasible solutions to the original problem, but they provide no such guarantees. In the context
of the AC-TNEP problem, it was recently demonstrated that feasibility can be quite challenging when using the DC power flow approximation [11]. Relaxations provide provable dual bounds to the original problem. In the context of the AC-TNEP, relaxations compute an optimistic value for the number of lines required to meet the future demands. The rest of this section introduces various heuristics, relaxations, and approximations of the AC-TNEP problem, and evaluates them on a simple 3-bus example to illustrate their properties. All the models presented below rely on a continuous relaxation of the discrete variable $z_{bin}$ and use the binarization of $z_{bin}$ from [13].

### A. Heuristics

Several heuristics have been proposed [11], [14], [15] for solving the AC-TNEP. In contrast to these constructive heuristics, we consider a destructive heuristic HAC-TNEP which maintains feasibility in each iteration. HAC-TNEP out-performs the constructive heuristic proposed in [11], produces high-quality primal solutions, and thus provides a good basis for comparison. HAC-TNEP starts with all of the lines in the network (there are sufficiently many lines to guarantee feasibility). On each iteration, HAC-TNEP attempts to remove one line from the network and tests if the network is still feasible in an AC Optimal Power Flow model (AC-OPF). Lines are selected in increasing order of relative loads. HAC-TNEP completes when no line can be removed.

### B. DC Power Flow

The DC model is a popular approximation of the AC power flow model motivated by design and operational considerations. It uses the polar voltage formulation of the AC power flow equations, i.e., $V = |V| e^{j\theta}$, and makes the following simplifications: (1) the susceptance is large relative to the conductance $|g| \ll |b|$; (2) the phase angle difference is small enough to ensure $\sin(\theta_n - \theta_m) \approx \theta_n - \theta_m$ and $\cos(\theta_n - \theta_m) \approx 1.0$; and (3) the voltage magnitudes $|V|$ are close to 1.0 and do not vary significantly. Under these assumptions, equations (M1.7-M1.8) reduce to

$$ p_{bin} = -b_{nm}(\theta_n - \theta_m) $$

(1)

and the resulting model is called the DC-TNEP formulation. Due to the voltage approximation, this formulation cannot capture any of the extensions presented in Section II-A. From a computational standpoint, the DC model is much more appealing than the AC power flow: It forms a system of linear equations and can be naturally embedded in Mixed-Integer Linear Programs (MIPs). For this reason, many TNEP models have focused on the DC-TNEP variant [1]-[4]. The DC-TNEP model however has no notion of line losses, reactive power flow, or bus voltage magnitudes. As noticed in [11], these inaccuracies may make it difficult to convert DC-TNEP solutions to AC-TNEP solutions.

### C. LPAC Power Flow

The LPAC power flow model [17] bridges the gap between the DC power flow and the AC power flow model without an increase in computational complexity (i.e., it can still be embedded in MIPs). The LPAC model approximates the AC power flow equations (also in the polar voltage formulation) with the following modifications: (1) $\sin(\theta_n - \theta_m) \approx \theta_n - \theta_m$; (2) the voltage magnitude at each bus is based on the deviation form a nominal voltage $|V| = 1.0 + \phi$; (3) the non-convex cosine function is replaced with a polyhedral relaxation ($g_{nm}$); (4) the remaining non-linear terms are factored and approximated with a first order Taylor expansion. These modifications yield the following power flow equations:

$$ p_{bin} = g_{nm} - b_{nm} g_{nm} - h_{nm}(\theta_n - \theta_m) $$

$$ q_{bin} = -g_{nm} + b_{nm} g_{nm} - h_{nm}(\theta_n - \theta_m) - b_{nm}(\phi_n - \phi_m) $$

and the resulting model is called the LPAC-TNEP formulation. The LPAC model captures line losses, reactive power flows, and an approximation of bus voltage magnitudes and is significantly more accurate than the DC power flow model. The LPAC-TNEP model can also incorporate the extensions discussed in Section II-A. The line charging constraint (M1.9) can be embedded in the LPAC-TNEP through a polyhedral outer approximation.

### D. SOCP Power Flow

A Second Order Cone Problem (SOCP) relaxation of the rectangular power flow equations was proposed in [19]. This was used in [13], [16] to build a SOCP-TNEP formulation. This formulation is appealing for two reasons: (1) High-quality industrial solvers exist for SOCPs; and (2) it is a relaxation and can be used for bounding the AC-TNEP problem. The primary disadvantage of the SOCP formulation is that it assumes the network has a sufficient number of virtual phase-shifting transformers (at least one for every cycle in the network). This assumption means that the SOCP formulation degenerates into a transportation model at the limit. In addition to extending [19] to TNEP, the formulation in [13] also adds an $\epsilon = \pi/720$ term for limiting the effects of the virtual phase-shifting transformers. It is not clear, however, whether the resulting formulation is still a relaxation. In this paper, we refer to this modified SOCP model as SOCP*-TNEP. The SOCP*-TNEP formulation in [13] included the line charging extension, but bus shunts and transformers were not mentioned.

### E. Comparing Expansion Plans

An expansion plan produced by any approximation or relaxation of the AC-TNEP problem can be infeasible for the original AC-TNEP problem (i.e., Model 1). To compare expansion plans, their constraint violations can be measured by computing an AC power flow in the expanded network for the generator dispatches and the voltage magnitudes obtained in the solution (the voltage magnitudes are fixed to 1.0 in the DC-TNEP). Two types of constraint violations are used to characterize the quality of an expansion plan: Line capacity violations and voltage magnitude violations. Line capacity violations measure how overloaded a line is. The relative capacity violation of a line $(n, m)$ is the maximum relative violation at either side of the line:

$$ \frac{\max \left(\frac{\sqrt{p_{bin}^2 + q_{bin}^2}}{S_{nm}}, \frac{\sqrt{p_{bin}^2 + q_{bin}^2}}{S_{nm}}\right)}{S_{nm}} $$

where $p_{bin}$ and $q_{bin}$ denote the real and reactive power on the line in the AC power flow. Obviously the line is overloaded...
when this value is greater than 100%. Voltage magnitude violations are defined as
\[
\text{max} \left(0, |V_n| - |\bar{V}_n|, |\bar{V}_n| - |V_n|\right)
\]
which captures violations above or below the desired limits.

IV. A CASE STUDY ON TNNEP SOLUTION METHODS

This section studies the behavior of each TNEP solution method on a simple 3-bus example. The network and its parameters are presented in Figure 1: It is designed so that the power flows on the paths 1–3 and 1–2–3 are roughly the same. This is illustrated on the AC load flow study presented in Table I. To understand the properties of the methods, two variants of this simple 3-bus system are considered: One with a low capacity line (SMVA thermal limit on 2–3) and one with high capacity limits (SMVA thermal limit on 1–2–3 and 1–3–2). The results of each TNEP solution method on these two cases are compared to our best-known AC feasible solution (AC∗-TNEP) in Tables II and III. The quality of the solutions produced by the approximation and relaxation methods are evaluated using the constraint violations metrics discussed previously. We now review the results of each method in detail.

In the low capacity case (Table II), AC∗-TNEP features an expansion plan with 27 additional lines. Because the approximate methods effectively capture line congestion caused by cycles in the network, both LPAC-TNEP and DC-TNEP correctly identify expansions with 27 lines. In contrast, the SOCP model provides a better solution by adding lines but still exhibits significant violations. Since similar trends were observed on the SOCP∗-TNEP solutions to the other benchmarks studied in this paper, this solution method is not discussed further.

The results on the tight voltage magnitude bound case in Table III are entirely different. AC∗-TNEP is an expansion plan requiring 5 expansions. DC-TNEP has no notion of the voltage magnitudes; Hence it adds no lines and exhibits significant voltage violations. The SOCP model proposes only 2 expansions, producing a network with less severe voltage violations. The remaining methods (HAC-TNEP, SOCP∗-TNEP, and LPAC-TNEP) all suggest adding 5 lines and have no violations.

In summary, on this 3-bus network, LPAC-TNEP provides an appealing tradeoff of accuracy and computational complexity. It finds the best-known AC solution in both cases. Similar observations also hold for the other benchmarks discussed in this paper. Hence, although there are many options for solving the AC-TNEP problem, we selected LPAC-TNEP as a reasonable approximation of the challenging AC-TNEP MINLP for the remainder of the paper. LPAC-TNEP will be compared with HAC-TNEP and DC-TNEP.

V. EVALUATION ON CLASSIC TEST SYSTEMS

The three most popular TNEP test cases are networks with 6, 24, and 46 buses (from [1], [3], [5] respectively). Since there are some variations on these test cases in the literature, we review our versions in detail. Our 6-bus benchmark corresponds to the one from [14] with bus 6 as the slack bus and 0.95 ≤ |\bar{V}_n| ≤ 1.05. Our 24-bus case is from [5] with bus 16 as the slack bus. The reactive injection capacity of generators is set to half of the active capacity, and line resistances are assigned one fifth of the reactance.

To understand the TNEP problem on a wide collection of networks, we design the following procedure for generating TNEP instances from any MATPOWER OPF test case. The loads and the capacity of generating units are scaled by a factor of 3 (except for synchronous condensers). The reactive injection capacity of each generator is set to half of the active capacity (i.e., \(q_g = -0.5 \, p_g\), \(\pi_q = 0.5 \, p_g\)). For lines with a capacity set to 9900 MVA, we use the value of the line loaded at a phase angle difference of 15 degrees instead. Finally, the cost of adding each line is set to 1 (i.e., the goal is to minimize the number of lines added). This procedure is used to build the 9, 14, 30, 39, 57, and 118 bus TNEP problems from the MATPOWER cases.1 In all of our benchmarks, expansions can select up to 6 lines per corridor (\(\tau = 6\)) and their quality

1The voltage constraints on case 57 dominate this network and we widen the bounds to 0.9 ≤ |\bar{V}_n| ≤ 1.1 to make the optimization task interesting.
is evaluated using an AC power flow as discussed in Section III-E. Each model was executed on a 2 x 2.00GHz Intel Quad Core Xeon E5405 with 2x6MB Cache and 16 GB RAM using GUROBI. 5.5 on 4 cores.

The HAC-TNEP heuristic was run until completion without a time limit, while the DC-TNEP and LPAC-TNEP algorithms were terminated after 2 hours. Our analysis considers the following metrics to characterize the quality of a TNEP solution (see Table IV): (1) the expansion cost (i.e., the objective value) with the number of lines added in parenthesis; (2) for the DC-TNEP and the LPAC-TNEP the maximum and average line violations with the number of lines with violations in parenthesis, and (3) the maximum voltage bound violations. Note that the AC power flow algorithm (AC-PF) used to convert the DC-TNEP and the LPAC-TNEP plans into AC feasible solutions is not guaranteed to converge to a solution. This is noted when it occurs.

Table IV shows that there is a significant gap in the cost between the TNEP solution found by the HAC-TNEP heuristic and the DC-TNEP models, confirming the results of [11]. The table also shows that DC-TNEP solutions have significant violations to the original AC-TNEP constraints. In three cases, the DC-TNEP solutions cannot be converted into a AC-feasible plan. In contrast to the DC-TNEP, the LPAC-TNEP solutions have significantly higher costs but also significantly less violations in both maximum and average values. The LPAC-TNEP model also produces significant improvements over the HAC-TNEP heuristic.

When using an approximate TNEP solution method, it is common to use a second corrective stage to eliminate violations [16]. Here we propose an alternative approach called constraint tightening, where the model constraints are modified before the solution process. For example, reducing the line capacity by 10% may mitigate small line loading violations and lead to AC-feasible solutions. Table V evaluates the quality of the DC-TNEP and LPAC-TNEP solutions when the line capacities are reduced by 10%. In the DC-TNEP model, the constraint-tightening procedure makes marginal improvements in the violations and the solutions still have major issues with AC-PF feasibility. In contrast, the constraint-tightening procedure has a great impact on the LPAC-TNEP solutions. It reduces the line violations in all cases and eliminates the violations in most cases. The violations occurring on the 30 and 118 cases can be reduced further by a more aggressive tightening than 10%.

Tables IV and V highlight three key points: (1) the DC-TNEP may significantly underestimate the expansion cost; (2) the LPAC-TNEP provides a nice compromise between accuracy and computational complexity; (3) constraint tightening is effective for eliminating line-loading violations in LPAC-TNEP. It is thus reasonable to conclude that LPAC-TNEP is an excellent vehicle for studies in transmission planning.

VI. TNEP WITH VAR COMPENSATION

The classic TNEP formulation presented in Section II assumes that transmission lines are the only components to be added to the network. Section IV indicated that tight voltage magnitude bounds may necessitate the addition of a significant number of lines. Although the AC-TNEP with voltage bounds is a common AC-TNEP formulation [11], [12], [16], tight voltage magnitude constraints may be unrealistic. As noted in [20], VAr compensation equipment is much cheaper than transmission lines and may be installed throughout a network to satisfy voltage magnitude bounds. Inspired by [20], this section investigates the transmission models assuming unlimited VAr compensation at every bus, which can be modeled by transforming every bus into a synchronous condenser with unlimited reactive injection capacity and a voltage set-point of 1.0. We call this model the perfect voltage profile AC power flow (AC-PVP) and our goal is to study how the DC-TNEP and LPAC-TNEP behave, and how much cheaper a TNEP solution might be in this context. The DC-PVP-TNEP has the same first step as the DC-TNEP. It is only when converting the resulting expansion plan that the VAr compensation plays a role. In contrast, the LPAC-PVP-TNEP exploits VAr compensation in
the first stage as well, since it models reactive power and voltage magnitudes.

Table VI revisits the evaluation of Section V with the AC-PVP-TNEP model. Since the AC-PVP model integrates the nominal voltage assumption of the DC power flow, it is no surprise that the gap between the HAC-PVP-TNEP solution is much smaller than in Section V. However, despite the gap reduction, there are still significant violations in the DC-PVP-TNEP solutions. In contrast, the LPAC-PVP-TNEP does amazingly well and has almost no violations without constraint tightening. Although the cost of line expansion is greatly reduced in these PVP solutions, the amount of required reactive injection capacity is roughly half the total active injection. Depending on the cost of VaR compensation devices, it may be advantageous to jointly optimize of expansion lines and VaR compensation as in [13], [15]. The main message however is that LPAC model is an excellent vehicle for these studies, given its accuracy and computational advantages.

The classic TNEP formulation and the PVP-TNEP formulation are special cases of a multi-objective optimization problem. The former assumes lines are cheap and VaR compensation is very expensive, while the later assumes the reverse. It is possible to enumerate some of the solutions along the Pareto Frontier to understand the tradeoff between line expansions and VaR compensation. Table VIII uses a scaling parameter $\lambda$ to balance the tradeoff between expansion costs an VaR compensation. The classic TNEP formulation is captured by $\lambda = 1$ while an extreme VaR minimization model is captured by $\lambda = 0$. The Table illustrates that there is a huge range of network design possibilities spanning from as few as 13 new lines to an amazing 200 new lines. However, in all of these design possibilities, the LPAC-TNEP is producing high-quality solutions with no violations.

To illustrate the tradeoff of joint line and VaR expansion further, Figure 2 shows part of the Pareto frontier for the 6 and 24 bus cases as found using the LPAC-TNEP model. The non-linear shape of both plots suggests that the tradeoff of line expansions and VaR compensation is non-trivial and the resulting network expansion plan may be heavily influenced by the cost models of line expansion and VaR compensation. Hence, it is best to optimize these two quantities jointly (e.g. [13], [15]), possibly under several cost models.

VII. POWER MARKET CONSIDERATIONS

Since the LPAC-TNEP provides an appealing tradeoff between solution quality and computational efficiency, it can be used for studies of more complex models of transmission expansion, producing results that can be trusted in contrast to the DC-TNEP. This section provides such an illustration. As observed in [1], the classic TNEP formulation presented in Section II is meaningful for a power system based on a regulated monopoly. However, its solutions may not be suitable for emerging competitive power markets, as physical transmission limitations may prevent competitive power generators from entering the market. In [1], an economic model was used to build four different competitive generation scenarios (called g1, g2, g3, and g4). Each scenario indicates the relative contribution of each generator to supplying the required demand (called contribution factors). These contribution factors amount to
TABLE IX. LPAC-TNEP SOLUTIONS WITH
CAPACITIES REDUCED BY 10%.

<table>
<thead>
<tr>
<th>Case</th>
<th>Cost</th>
<th>LPAC-TNEP</th>
<th>Volume</th>
<th>V violation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1171 (29)</td>
<td>0 (0)</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>1007 (21)</td>
<td>0 (0)</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

This paper revisited the gap between the Transmission Network Expansion Planning (TNEP) solutions produced by the DC power flow approximation and AC power flow heuristics. It was shown that the TNEP solutions produced by the DC power flow approximation significantly underestimate the expansion costs and have significant violations in the AC power flow model. The recent LPAC power flow model was proposed to bridge the gap between infeasible DC-TNEP solutions and AC heuristics. It was demonstrated that the LPAC-TNEP solutions have minimal constraint violations and, with a constraint tightening procedure, these violations can often be eliminated entirely.

The strength of the LPAC formulation was further demonstrated on additional studies on the joint optimization of line expansion and VAR compensation, as well as a competitive market study. The VAR planning study showed that the gap between the DC approximation and AC heuristics may be reduced through two-stage VAR planning, but significant AC violations still remain. In contrast, the LPAC-TNEP solutions have no violations in this two-stage approach. A study on the co-optimization of line expansions and VAR planning enabled by the LPAC-TNEP formulation illustrated the delicate balance between line and VAR costs models on the resulting expansion plans. Finally, a competitive market study indicated that the cost of the TNEP solutions may increase significantly to accommodate emerging energy markets.

Overall, this study indicates that the LPAC power flow model provides a good tradeoff between computational benefits and model accuracy for solving TNEP problems. Furthermore, it indicated that AC feasible expansion plans can be heavily influenced by the underlying TNEP formulation (e.g., adding VAR planning or market considerations). Great care should be taken in selecting an appropriate TNEP formulation for the study at hand. Future work utilizing the LPAC TNEP model should consider extending the models proposed here by incorporating network faults with recourse to ensure N–1 reliability as well as incorporating multiple generation and loading scenarios to produce more flexible and robust expansion plans.

ACKNOWLEDGMENT

This work was conducted in part at NICTA and is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

REFERENCES


Reliable power transmission networks

(Extended Abstract)

Rodrigo R. Gumucio E.
The University of Melbourne & NICTA
RGumucio@student.unimelb.edu.au

Carleton Coffrin
NICTA & The University of Melbourne
Carleton.Coffrin@nicta.com.au

Pascal Van Hentenryck
The Australian National University & NICTA
Pascal.VanHentenryck@anu.edu.au

ABSTRACT

In the context of the Transmission Network Expansion Planning (TNEP) problem, recent studies show that there exists a gap, between high cost expansion plans produced by AC-heuristics and potentially infeasible expansion plans produced by DC-approximations, that can be bridged by using the recently proposed LPAC power flow approximation.

The work in progress for this paper aims to incorporate the $n - 1$ reliability criterion to the mentioned expansion planning study by proposing a genetic algorithm which may scale to large networks.

Categories and Subject Descriptors
I.2 [Artificial Intelligence]: Miscellaneous; G.1.6 [Optimisation]: Constrained optimisation

Keywords
Power transmission network planning, $n - 1$ reliability, linear ac power flow, non-linear non-convex optimisation, genetic algorithm, ac feasible expansion plans

1. INTRODUCTION

Power transmission networks need to be expanded, for example, to satisfy a projected future demand or to make them more robust against potential threats like natural disasters.

Expanding a network entails to add a set of new circuits to it which is typically very expensive. We strive then to choose a set of circuits, called an expansion plan, such that the investment cost of adding them to the network is minimum.

The minimisation problem is hard because, due to the nature of the AC power flow model, it is subject to complex non-convex and non-linear constraints which not only grow with the number of nodes in the network but also involve, among a huge number of model variables, both discrete and continuous decision variables.

The intractability of this problem is, in most studies, worked around by using a heuristic to the full AC model or by relying on an approximation based on the popular DC linear power flow model. However, due to the limitations of DC-approximations which ignore reactive power and voltage magnitudes, recent work in the community started to consider the TNEP problem with the full AC power flow equations (AC-TNEP), as for example [2].

We alternatively take the new LPAC[3] power flow model, which attempts to provide a good balance between the tractability of the DC model and the accuracy of the AC model, and propose a LPAC-TNEP formulation which proves to be a good mechanism to find expansion plans of reasonable cost with no network violations.

2. PRIOR WORK AND RESULTS

The Transmission Network Expansion Planning problem asks to find a lowest cost expansion plan to satisfy a projected future energy demand. Our study in [1] found that an LPAC-TNEP formulation, which can benefit of industrial Mixed Integer Program (MIP) solvers, can be used to find AC feasible expansion plans of reasonable cost for several benchmarks. For illustrative purposes, results are shown in the left side of Table 1 for the 6 and 24 bus benchmarks. Evidence shows, however, that even using state-of-the-art industry standard tools, the MIP will struggle and even fail to find feasible solutions for larger benchmarks or even for the same benchmarks used in [1] when the reliability constraints are included into the model.

3. WORK IN PROGRESS AND RESULTS

When deciding which circuits to add to an expansion plan, usually a reliability criteria is desired to be taken into account in the analysis in order to make the network more robust. We are studying now a genetic algorithm (GA) for optimisation which relies on an LPAC-based simulator to check for $n - 1$ feasibility. The main characteristics of the GA are a fitness function which drives towards both feasibility and optimality; a two point crossover operation; and a mutation operation with both an increasing mutation probability and a filter based on the metropolis probability.

Current results, as shown in the right side of Table 1, give some evidence that it may be possible to find $n - 1$ compliant AC feasible expansion plans of reasonable low cost with an acceptable number of not so expensive simulations.

4. REFERENCES


Table 1: Solutions Produced by LPAC-TNEP

<table>
<thead>
<tr>
<th>Case</th>
<th>$\text{Cost ($)}$</th>
<th>Violations</th>
<th>$\text{Cost ($)}$</th>
<th>Violations</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>130 (15)</td>
<td>0.00</td>
<td>200 (8)</td>
<td>0.00</td>
</tr>
<tr>
<td>24</td>
<td>685 (15)</td>
<td>0.00</td>
<td>1557 (36)</td>
<td>0.00</td>
</tr>
</tbody>
</table>
2. A few more tables

### Table C.1. Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP with constraint tightening at 0%.

<table>
<thead>
<tr>
<th>Case</th>
<th>DC-TEP</th>
<th>LPAC-TEP</th>
<th>HAC-TEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>Cost Max. [%]</td>
<td>Cost Mean [%]</td>
<td>Cost Max. [%]</td>
</tr>
<tr>
<td>6</td>
<td>160 (6)</td>
<td>57.32 (5)</td>
<td>21.90</td>
</tr>
<tr>
<td>24</td>
<td>2310 (43)</td>
<td>52.66 (26)</td>
<td>8.21</td>
</tr>
<tr>
<td>46</td>
<td>5698 (47)</td>
<td>13.46 (8)</td>
<td>8.21</td>
</tr>
</tbody>
</table>

### Table C.2. Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP with constraint tightening at 5%.

<table>
<thead>
<tr>
<th>Case</th>
<th>DC-TEP</th>
<th>LPAC-TEP</th>
<th>HAC-TEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>Cost Max. [%]</td>
<td>Cost Mean [%]</td>
<td>Cost Max. [%]</td>
</tr>
<tr>
<td>6</td>
<td>160 (6)</td>
<td>57.32 (5)</td>
<td>21.90</td>
</tr>
<tr>
<td>24</td>
<td>2310 (43)</td>
<td>52.66 (26)</td>
<td>8.21</td>
</tr>
<tr>
<td>46</td>
<td>5698 (47)</td>
<td>13.46 (8)</td>
<td>8.21</td>
</tr>
</tbody>
</table>

### Table C.3. Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP with constraint tightening at 10%.

<table>
<thead>
<tr>
<th>Case</th>
<th>DC-TEP</th>
<th>LPAC-TEP</th>
<th>HAC-TEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>Cost Max. [%]</td>
<td>Cost Mean [%]</td>
<td>Cost Max. [%]</td>
</tr>
<tr>
<td>6</td>
<td>160 (6)</td>
<td>57.32 (5)</td>
<td>21.90</td>
</tr>
<tr>
<td>24</td>
<td>2310 (43)</td>
<td>52.66 (26)</td>
<td>8.21</td>
</tr>
<tr>
<td>46</td>
<td>5698 (47)</td>
<td>13.46 (8)</td>
<td>8.21</td>
</tr>
</tbody>
</table>
### Table C.4. Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP with constraint tightening at 15%.

<table>
<thead>
<tr>
<th>Case</th>
<th>DC-TEP</th>
<th>LPAC-TEP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Thermal limits violations</td>
<td>Voltage vio.</td>
</tr>
<tr>
<td>6</td>
<td>160 (6)</td>
<td>130 (5)</td>
</tr>
<tr>
<td>24</td>
<td>2524 (45)</td>
<td>316 (10)</td>
</tr>
<tr>
<td>46</td>
<td>569810 (47)</td>
<td>143865 (12)</td>
</tr>
<tr>
<td>9</td>
<td>5 (5)</td>
<td>2 (2)</td>
</tr>
<tr>
<td>14</td>
<td>15 (15)</td>
<td>1 (1)</td>
</tr>
<tr>
<td>30</td>
<td>22 (22)</td>
<td>8 (8)</td>
</tr>
<tr>
<td>39</td>
<td>50 (50)</td>
<td>35 (35)</td>
</tr>
<tr>
<td>57</td>
<td>49 (40)</td>
<td>2 (2)</td>
</tr>
<tr>
<td>118</td>
<td>40 (40)</td>
<td>8 (8)</td>
</tr>
</tbody>
</table>

### Table C.5. Evaluation of solutions produced by HAC-TNEP, DC-TNEP, and LPAC-TNEP with constraint tightening at 20%.

<table>
<thead>
<tr>
<th>Case</th>
<th>DC-TEP</th>
<th>LPAC-TEP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Thermal limits violations</td>
<td>Voltage vio.</td>
</tr>
<tr>
<td>6</td>
<td>160 (6)</td>
<td>130 (5)</td>
</tr>
<tr>
<td>24</td>
<td>2479 (45)</td>
<td>383 (12)</td>
</tr>
<tr>
<td>46</td>
<td>56956 (49)</td>
<td>162775 (15)</td>
</tr>
<tr>
<td>9</td>
<td>5 (5)</td>
<td>3 (3)</td>
</tr>
<tr>
<td>14</td>
<td>15 (15)</td>
<td>1 (1)</td>
</tr>
<tr>
<td>30</td>
<td>29 (29)</td>
<td>11 (11)</td>
</tr>
<tr>
<td>39</td>
<td>57 (57)</td>
<td>42 (42)</td>
</tr>
<tr>
<td>57</td>
<td>49 (49)</td>
<td>2 (2)</td>
</tr>
<tr>
<td>118</td>
<td>44 (44)</td>
<td>10 (10)</td>
</tr>
</tbody>
</table>
Afterword

It has been quite an adventure, in more than one sense, coming to Australia and living and studying in Melbourne. I thank again both The University of Melbourne and National ICT Australia for this research experience. I am happy at this point that, by working on a problem that is more related to engineering and the natural sciences than to computer science, I have widened my view about constraint satisfaction and optimisation problems, and, moreover, made clearer my ideas about what applied research is about.

Australia is a great country, and Melbourne an extraordinary city to live in. Every four years, I find myself abroad for study purposes: 2006, 2010, 2014. I hope by 2018 I find myself involved in a PhD programme at a prestigious university.

Despite the necessity to have been expatriate for learning purposes, I have been following the transformational process going on in Bolivia the last years, and I feel quite proud of all of the positive changes. I expect we will soon be hearing good news about our legitimate right of sovereign access to the sea.
Index

AC-OPF, see AC optimal power flow
AC-TNEP, 72
active power, 44, 51, 126, 127
demand, 126
generation, 50, 126
load, 50
losses, 51, 128
please, see also average power
AC optimal power flow, 53, 68
AC power-flow solution, 66
admittance, 48, 135
shunt, 49, 125, 135
apparent power, 44
average power, 44, 45
bad configuration, 105, 113
blackout, 29
bound operation, 80–82
branch-and-bound
algorithm, 79–82
framework, 80, 84
search, 79
tree, 83
branch operation, 80–82
bus, 24, 25, 65, 125
operational constraints, see
voltage bounds constraints
shunt, 49
case-study networks, 91
case-study networks, 135
circuit, 24, 25, 49, 65, 125
unified π model, 49
charging, 135
operational constraints, see
thermal-limit constraints
power, 127
shunt, 50
complete-search, 79

complex power, 47
please, see also power
conclusion, 116
conductance, 48
please, see also admittance
congestion, 73
constraint tightening, 102, 115
convex programming, 33
corridor, 24
crossover operation, 62, 86–88, 99
current, 39–42, 47, 49, 125, 127
alternating, 42, 43
direct, 41
rms, 43
DC-TNEP, 72
gx networks solutions, 75
decision variables, 93
dispatch, 68
distribution network, 22
please, see also electric power system
electric charge, 39
electric potential energy, 40
please, see also voltage
electric power, 39
please, see also power
electric power system, 22
AC, 42
DC, 43
electron, 39
evaluation on classic test systems
experiment, 101
results, 102
evidence, 93, 105
evolutionary search, 79, 85
expansion plan, 30, 54, 55
investment cost gap, 105
underestimated cost, 115
expansion problem, 31
exponential growth, 68
findings, 116
fitness function, 87, 88, 98, 99
future work, 117
GA, see genetic algorithm
generating-unit, 22, 65
    operational constrains, 68
    please, see voltage controlled bus
genetic algorithm, 61, 79, 85, 86, 98
gx networks, 102, 136
HAC-TNEP, 72, 97
    algorithm, 98
heuristic, 60
impedance, 41, 47–49, 125
individual, 86
intractable, 31, 80
joint expansion
    circuit and VAr, 110, 115
    co-optimisation, 111, 115
    tradeoff, 112
linear program, 32
load, 22, 25, 65
load-flow, 28
    please, see power-flow analysis
local search, 61, 79, 85
LP, see linear program
LPAC power-flow constraints, 93, 115, 125,
    130, 133
cos polyhedral relaxation, 93, 130
assumptions, 130
constraints, 94
correlation plots, 74, 134
decision variables, 93
objective function, 94
thermal limits polyhedral approximation, 95
machines, 103
measures, 93, 105
method, 92
metropolis probability, 100
MILP, see mixed-integer linear program
mixed-integer linear program, 32
mixed-integer non-linear program, 54
mutation operation, 62, 86–88, 100
n-1 reliability
    experiment, 102
    results, 113
Newton-Raphson method, 66, 134
non-linear program, 54
Ohm's law, 41, 47, 48, 125
    please, see also power-flow constraints
operational constraints, 68
    violation, 26, 115
optimisation problem, 32, 81
    instance, 32
papers, 137
Pareto frontier, 56
    joint expansion, 110
    participation factors, 112
perfect voltage profile, 101, 108, 111
    reactive power injection, 111
phase shift, 42, 44, 45, 47
physical constraints, see
    power-flow constraints
population, 99
power, 22, 26, 42, 43, 47
    active, see active power
    instantaneous, 43
    losses, 126
    reactive, see reactive power
power-flow analysis, 28
power-flow constraints, 26, 48, 66, 67, 129
    AC, 125
    LDC, 57, 66, 125, 129
LPAC, see LPAC power-flow constraints
    linear, 125, 131
    non-linear, 48, 51, 52, 65, 66, 125, 127
power factor, 44, 47
power market considerations
    experiment, 102
    results, 112
proton, 39
reactance, 45, 48, 135
    capacitive, 45
    inductive, 45
    please, see also impedance
reactive power, 45, 47, 51, 126, 127
    generation, 50
    load, 50
    losses, 51, 128
shunt generation, 51
real power, 44, 45
  please, see also average power
reliability criterion, 30
  \( n = 1, 31, 57, 58, 70 \)
research question, 76
resistance, 25, 41, 45, 48, 135
  please, see also impedance
resistive heating, 42, 43
results, 105
rms please, see root mean square
root mean square, 43
scenario, 30
select operation, 86, 87, 89, 99
shunt, 24, 47, 125
simplex algorithm, 79
simulation-optimisation, 61, 84
  framework, 80, 85
slack bus, 27
SOCP-TNEP, 72
susceptance, 48, 135
  please, see also admittance
synchronous condenser, 24, 101
  please, see also generator
systematic-search, 60
tension, 40
  please, see also voltage
termination criterion for GA, 87, 100
test-case networks, 92
test-case networks, 135
test-set networks, see test-case networks
thermal-limit constraints, 25, 67, 68
  violations measure, 92
TNEP, see
  transmission network expansion planning
TNEP with VAr compensation
  experiment, 101
  results, 108
transformer, 24, 43, 49, 50, 65, 126, 128,
  136
  model, 128
in-phase, 50, 51, 129
phase-shifting, 51, 129
tap, 49, 125, 136
  please, see also circuit
transmission line, 24, 49, 51, 65
  drop, 41, 51
  impedance, 47
  model, 128
voltage, 40
  please, see also circuit
transmission network, 22, 25, 65
  graph abstraction, 25
  operating normally, 26, 28, 66, 67
transmission network expansion planning,
  31, 54, 55, 65
  configuration, 68, 93
decision variables, 66
  model, 69
  solution configuration, 69
ture power, 44
  please, see also average power
voltage, 39–42, 47, 49, 125
  alternating, 42, 43
  magnitude deviation, 130
  rms, 43
  unipolar, 41
voltage bounds constraints, 68
  violations measure, 92
voltage controlled bus, 27
work, 40
  please, see also electric potential energy
Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
GUMUCIO-ESCOBAR, RODRIGO

Title:
Reliable power transmission networks

Date:
2015

Persistent Link:
http://hdl.handle.net/11343/55465

File Description:
MPhil thesis: Reliable Power Transmission Networks