Expansion Planning of Low-Voltage Grids Using Ant Colony Optimization

Ausbauplanung von Niederspannungsnetzen mithilfe eines Ameisenalgorithmus

Master's Thesis

Lukas Gebhard
University of Freiburg
Germany

August 2021

Keywords

Distribution Grid • Network Expansion • Grid Reinforcement • Energy Transition • Automated Planning • Swarm Intelligence

Examiners

Prof. Dr. Hannah Bast
Chair of Algorithms and Data Structures
University of Freiburg

Prof. Dr. Christof Wittwer
Director of Business Division Power Electronics, Grids and Smart Systems
Fraunhofer Institute for Solar Energy Systems ISE

Supervisors

Dipl.-Ing. Wolfgang Biener, M.Sc. Matthias Hertel

Contact

Lukas Gebhard
Freiburg, Germany
research@lukasgebhard.de

Version

1.0

Declaration of Authorship

I hereby declare that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work. I also hereby declare that my thesis has not been prepared for another examination or assignment, either in its entirety or excerpts thereof.

Freiburg, August 2021

Lukas Gebhard

Abstract

The decarbonization of the energy sector necessitates a large-scale expansion of low-voltage (LV) grids. Yet, such expansion is expensive and increasingly difficult to plan, particularly in face of the rising number of distributed energy generation facilities and the electrification of the transportation sector. To assist utilities in the planning of electricity grids, scholars have proposed a variety of tools that automatically search for optimal grid expansion strategies. Tools that implement *ant colony optimization* (ACO), a heuristic framework for the solution of combinatorial optimization problems, rank among the best. Nevertheless, only few researchers have studied the application of ACO to LV grid expansion planning; and existing research neglects the cost savings potential of combining conventional power line expansion with a reconfiguration of power switches. To fill this gap, I present *AntPower*, a tool that searches for a minimum cost strategy to expand a given LV grid, subject to topological and electrical requirements that are common in Europe. Unlike existing tools, AntPower integrates the installation, reinforcement, and dismantling of power lines with an optimization of power switch settings. For evaluation, I consider the case of expanding a heavily overloaded grid that powers an 800-inhabitant village in rural Germany. The expansion plan generated by AntPower is by 60% cheaper than an expansion plan obtained through conventional, manual planning based on expert knowledge, and is by 64% cheaper than the expansion plan generated using a local search algorithm. Finally, a sensitivity analysis indicates that AntPower is robust against changes in most of its parameters; and for the sensitive parameters, good default values exist.

Zusammenfassung

Die Dekarbonisierung des Energiesektors erfordert einen umfassenden Ausbau von Niederspannungsnetzen. Doch ein solcher Ausbau ist teuer und zunehmend schwer zu planen, besonders in Hinblick auf die steigende Zahl verteilter Energieerzeugunganlagen und die Elekrifizierung des Verkehrssektors. Um Energieversorgungsunternehmen in der Planung von Stromnetzen zu unterstützen, haben Wissenschaftler:innen eine Vielzahl verschiedener Tools entwickelt, die automatisiert nach optimalen Netzausbaustrategien suchen. Zu den Tools, die am besten abschneiden, gehören Implementierungen von *Ant Colony Optimization* (ACO) - ein heuristisches Verfahren zur Lösung kombinatorischer Optimierungsprobleme. Doch die Anwendung von ACO auf die Ausbauplanung von Niederspannungsnetzen wurde bisher wenig erforscht. Zudem vernachlässigt die Forschung das Kosteneinsparungspotential, das mit einer Kombination von konventionellem Leitungsausbau und einer Trennstellenoptimierung (d.h. einer Neukonfigurierung von Leistungsschaltern) einhergeht. Um diese Lücke zu schließen, wird in dieser Arbeit *AntPower* vorgestellt, ein Software-Tool, das nach einer kostenminimalen Ausbaustrategie für ein gegebenes Niederspannungsnetz sucht und dabei in Europa übliche topologische und elektrische Rahmenbedingungen berücksichtigt. Im Gegensatz zu bestehenden Netzplanungstools vereint AntPower die Installation, Verstärkung und den Rückbau von Stomleitungen mit einer simultanen Trennstellenoptimierung. Zur Evaluierung wird der Ausbau eines stark überlasteten Netzes betrachtet, das ein Dorf mit 800 Einwohner:innen versorgt. Der von AntPower generierte Ausbauplan ist um 60% günstiger als ein Ausbauplan, der mit einer konventionellen, händischen, auf Expertenwissen beruhenden Planungsmethode erstellt wurde, und ist um 64% günstiger als der Ausbauplan, der von einem lokalen Suchalgorithmus generiert wurde. Im Rahmen einer Sensitivitätsanalyse zeigt sich AntPower robust gegenüber von Veränderungen der meisten Parameterwerte und für die übrigen Parameter sind gute Standardwerte vorhanden.

Contents

1 Introduction

Before the energy transition, the structure of electricity grids was relatively simple. Large power plants would feed electricity into the *transmission grid*, which would transmit the electricity over long distances to large factories and *distribution grids*. The distribution grids would then distribute the electricity among local consumers such as small factories, supermarkets, and residential buildings. Meanwhile, the decarbonization of the energy sector has triggered a large-scale transformation of distribution grids, involving the integration of generation facilities (e.g., photovoltaic appliances), energy storage (e.g., electric vehicles), and additional electrical loads (e.g., heat pumps). This transformation pushes distribution grids to their limits, thereby forcing many grid operators to expand their grids [1, 2].

However, planning the expansion of distribution grids is becoming increasingly challenging for several reasons. First, distribution grids are becoming more complex due to the integration of new technologies (e.g., load-scheduling controllers, on-load tap changers). Second, as distribution grids have not been designed to accommodate distributed electricity generation facilities [3, 4], the integration of such facilities can necessitate unusually large-scale restructuring and reinforcement measures. Third, as the energy transition accelerates, future supply and demand of electricity is becoming harder to predict (e.g., due to the volatility of renewable energy sources, the electrification of the transportation sector, and the liberation of electricity markets) [5–7].

Distribution grids often consist of sub-grids operating under low voltage (LV) or medium voltage (MV), and the energy transition affects both the LV and the MV grids; yet, in two respects, the expansion of LV grids poses a particular challenge. First, LV grids typically make up the largest share of a country's electricity grid in terms of line length; for example, Germany's LV grids constitute two thirds of the German grid [8]. Second, many LV grids must cope with large amounts of fluctuating and dispersed electricity feed-ins (e.g., 37% of Italy's and 54% of Germany's photovoltaic capacity is installed in LV grids [9, 10], and 98% of Italy's and 97% of Germany's photovoltaic plants are installed in LV grids [9, 11]). The resulting need for grid expansion incurs high costs; for example, the German Federal Ministry for Economic Affairs and Energy estimated in 2014 that—given conventional planning principles—the expansion of Germany's LV grids in the period from 2012 to 2032 requires investments of 4.6B € to 9.8B € [2].

To facilitate the planning of distribution grids, many scholars have developed tools to automate (parts of) the planning process [12–15]. The proposed approaches vary strongly, both in the considered planning conditions (e.g., constraints, inputs) and the applied methods (e.g., tabu search, genetic algorithms). While it is not known which method performs best under any given set of planning conditions, the application of *ant colony optimization* (ACO) [16] to grid planning is a promising research direction, because ACO-based grid planning methods outperformed other methods in several studies [17–23]. Although various ACO-based grid planning tools exist [17–27], only few support the expansion planning of LV grids; and those that do have only been evaluated on simplistic grids, that is, unrealistically small grids [21], grids without switches [17, 18, 21, 24], or grids with only one feed-in point [21, 24].

To fill this gap, I present *AntPower*, an implementation of ACO that facilitates efficient planning of LV grids, particularly grids that face heavy load or are expected to face heavy load in the near future. AntPower searches for a cost-optimal strategy of expanding a given LV grid, adhering to topological and electrical norms that are common in Europe. When generating an expansion plan, AntPower regards power lines as sequences of line segments, each of which may be installed, reinforced, and dismantled individually to keep down costs. Simultaneously, AntPower searches for an optimal routing of electricity by evaluating different configurations of power switches. Such re-routing of electricity can reduce load on parts of the grid, thus making expensive construction projects redundant. For evaluation, I consider a 20 km-long LV grid, which serves an 800-inhabitant village in rural Germany. There, the rising penetration of solar energy necessitates large-scale grid expansion. The expansion plan generated by AntPower meets all topological and electrical requirements. Its total cost (i.e., the cost of digging up roads, acquiring cables, and operating switches) is by 60% lower than the cost of an expansion plan obtained through careful manual planning, and is by 64% lower than the cost of an expansion plan derived using a local search algorithm. Additionally, a systematic sensitivity analysis indicates AntPower's robustness against changes in most of its parameters; and for the few sensitive parameters, good default values exist.

Overall, this thesis makes the following contributions to previous research on grid planning:

1. It presents AntPower, the first ACO-based tool that allows to plan the expansion of LV grids using a combination of line expansion (i.e., installation, reinforcement, and dismantling of power line segments) and switch reconfiguration (i.e., the optimization of switch settings).
2. It shows how planning the expansion of a large LV grid using ACO can cut expansion costs by 60% as compared to using a conventional planning method, and can cut costs by 64% as compared to using a local search algorithm.
3. It documents a straight-forward way of applying ACO to grid expansion planning in a limited context, disregarding advanced concepts such as load scheduling, multi-stage planning, and multi-objective optimization.
4. It analyses the sensitivity of ACO's parameters in the context of LV grid expansion planning, and reports the ranges of parameter settings that yield best results.
5. It gives a concise and easy-to-follow introduction to grid planning, making grid planning research accessible to scholars without background in power engineering.

This thesis has the following structure. Chapter § 2 lays the technical foundations that are necessary to understand the problem AntPower seeks to solve. Chapter § 3 then formulates this problem. Chapter § 4 documents how AntPower implements ACO to solve the given problem. Chapter § 5 evaluates AntPower's performance, and chapter § 6 concludes.

2 Technical Foundations

To begin with, this chapter gives a short introduction to the planning of electricity grids. The chapter is intended to be easy to follow for scholars with background in mathematics, computer science, physics, or engineering. I also recommend experts in power engineering to read this chapter, as it establishes common ground for subsequent chapters.

This chapter has the following structure. First, subsection § 2.1 introduces the building blocks and architecture of electricity grids, and outlines the requirements that grids must meet. Then, subsection § 2.2 gives an overview of grid planning approaches by characterizing the underlying conditions, models, and methods.

2.1 Grids

A *grid* is a network delivering electricity from feed-in points to components consuming electricity. Grids highly vary in size, powering small villages just like whole continents.

2.1.1 Components

The components of a grid are the following:

Sources and Loads

Sources feed electricity into the grid. Examples include nuclear power plants, wind farms, domestic photovoltaic appliances, and diesel generators. In turn, *loads* - such as households, public facilities, and factories - consume electricity.

Storage Systems

Storage systems draw electricity from the grid, store the electricity, and feed it back to the grid at a later time. That is, storage systems act as loads during charge periods, and play the role of sources during discharge periods. Examples of storage systems are batteries and pumped-storage hydropower plants.

Controllers

Some sources and loads are equipped with *controllers*, which allow them to react to state changes in the grid. For example, when a photovoltaic plant generates more electricity than the grid can handle, the plant's controller may detect a voltage spike and curtail the generation of electricity. Conversely, controllers that control loads (e.g., charging stations) can schedule the load's consumption of electricity. From an economic perspective, the use of controllers is a service of electricity suppliers and consumers in that it provides grid operators with flexibility.

Lines

Lines transmit electricity from sources to loads. High voltage lines are typically overhead lines, except in highly populated areas [28]. For lower voltages, some countries prefer overhead lines, while others prefer underground lines [28]. A large number of cable types exists, varying in features such as transmission capacity, impedance[1], and cost [28].

Buses

Buses connect lines. A bus is essentially a bar of conductive metal [29], to which two or more lines attach. In terms of reliability and stability (§ 2.1.2), buses are critical points [29].

Transformers

A *transformer* transforms an input voltage to some output voltage. Transformers with *tap-changers* allow adjusting the ratio of input and output voltage. There are two types of tap-changers [28]: *no-load* and *on-load* tap changers. A transformer that is equipped with a no-load tap-changer must be de-energized before the voltage ratio is adjusted. In contrast, on-load tap-changers allow to adjust the voltage ratio during normal operation, either manually or automatically in response to undesired voltage deviations at the transformer's output.

Switches

Switches allow opening and closing connections between grid components. Types of switches differ both in application and technical specification [28]. Some switches operate automatically, such as *circuit breakers*: To protect other components, a circuit breaker opens when current flow exceeds a critical value [29]. Other types of switches serve to reroute electricity, or to de-energize grid segments for maintenance and in cases of congestion and failures [29].

2.1.2 Requirements

When managing grids, utilities must meet several, partly competing goals:

Reliability measures the degree at which electricity demand is met over time [29]. In a perfectly reliable grid, every load's electricity demand is always met. Obviously, this requires sources of electricity to feed in enough electricity to satisfy all loads at all times. However, the grid must also be able to transmit the feed-ins to loads. In fact, most cases of supply shortfall are caused by grid failure rather than generation shortfall [29].

Security describes a grid's ability to withstand component failures [29]. A common criterion is *N-1 redundancy* [29, 30]. A grid is N-1 redundant if it remains functional when a single, arbitrary component fails. For that, the grid must have backup components such as spare transformers and additional power lines.

[1] Impedance is a measure of a component's opposition to the flow of power through the component.

Stability describes the tendency of an alternating current (AC) grid to maintain a constant voltage level and to synchronize the phase of voltage oscillations across the grid [29]. Both are crucial conditions for the grid to remain in a balanced operating state.

Power Quality measures how close various electrical quantities are to nominal values [29]. Measurements are taken at the points where loads attach to the grid [29]. For power quality to be high, voltage and AC frequency must be steady and close to the nominal values, and the waveform of voltage must resemble a sine wave [29].

Efficiency measures the cost that arises in the process of meeting the above requirements. Grid planners traditionally define this cost as the sum of *investment costs* and *operational costs* [31]. Investment costs are costs for buying and installing grid components. Operational costs include costs for maintenance, for the use of load flexibility (§ 2.1.1), and for energy losses[2] [31]. In addition to economic costs, other types of costs increasingly gain attention in grid planning [31, 32]. For example, power lines incur high ecological costs if they pass through natural reserves, and may cause high social costs if they are built close to popular tourist attractions.

2.1.3 Structure

Large grids consist of multiple layers, each layer being operated at a different voltage level.[3] A grid's top layer typically spans the widest area, and is operated at the highest voltage level. With decreasing position in the layer stack, the sub-grids' supply areas get smaller. For example, Germany's national grid has six main layers [28]. The lowest layer operates at 400 V, also known as *low voltage* (LV). It is superimposed by *medium voltage* (MV) layers, operating at 10 kV and 20 kV and covering distances up to about 100 km. The highest layers are the *high voltage* (HV) layer (110 kV) and the *extra high voltage* (EHV) layers (220 kV, 380 kV).

Transformers connect layers with each other. Figure 2.1 visualizes the connections between the two EHV layers of Germany's national grid, disregarding geographical distances. Transformers (green) connect the upper layer (purple) with the lower layer (blue). On an abstract level, the transformers constitute the interface between the two layers; from the lower layer's perspective, the transformers feed in electricity to the grid, whereas from the upper layer's perspective, the transformers draw electricity from the grid.[4] This abstraction allows viewing each layer as a standalone grid, disregarding the other layers.

Grid layers not only have different voltage levels but also different topologies. Each topology makes a different trade-off between economy and security (§ 2.1.2) [27]. In upper layers of a grid, security is particularly important, because failures likely propagate down to subordinate

[2] Energy loss is the difference of produced and consumed electricity. This difference is always greater than zero, because part of the produced electricity gets converted to heat during transmission.

[3] While higher voltage implies higher risk of electric shocks and fires and thus requires better insulation, it also reduces the fraction of energy converted to heat in the wires [29]. The longer the wires, the more significant is the loss of energy [29]. Grids transmitting energy over far distances are therefore operated at high voltages. Smaller (sub-)grids, however, are operated at lower voltages, because insulation costs and safety concerns outweigh costs for energy losses [29].

[4] Usually, electricity flows from high to low grid layers, but it can also flow in the opposite direction.

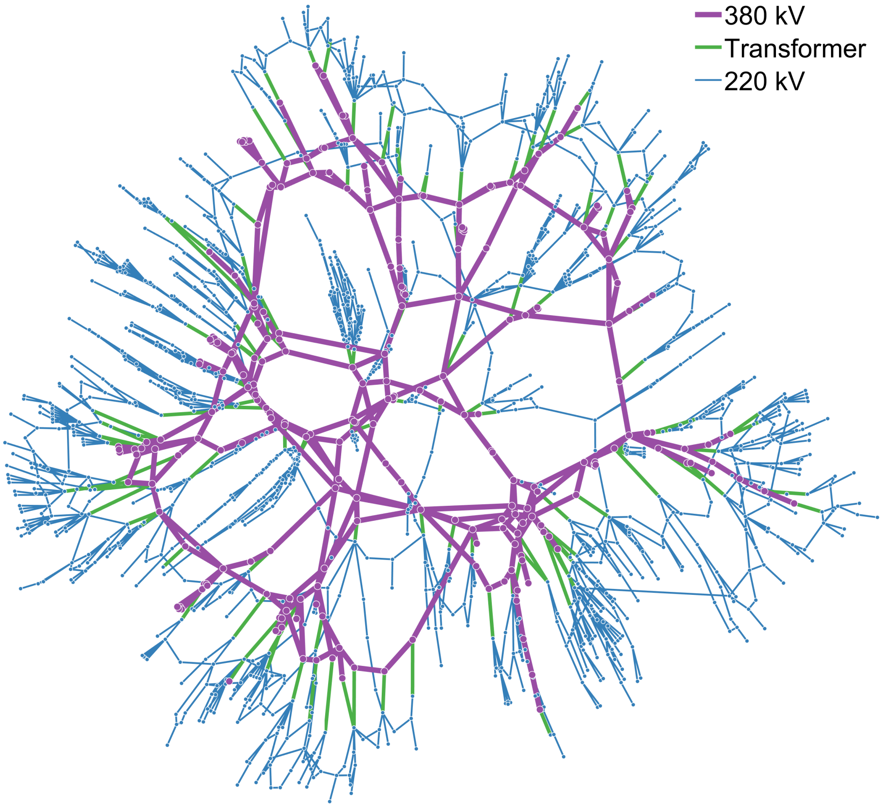


Figure 2.1: Graph representation of a real-world grid structure [33]. Nodes are buses; edges represent lines and transformers. Image by Cuffe [34] (license: https://creativecommons.org/licenses/by-sa/4.0/)

layers. Therefore, a grid's top layers are characterized by high interconnectivity, which implies high redundancy and thus high security. In lower grid layers, however, costs of additional components outweigh an increase in security.

Figure 2.2 shows topologies that are common in Germany. *Radial* topology is typical of LV grids, which power small villages and city blocks [28]. In radial topology, lines called *feeders* depart at the upstream transformers (i.e., at the transformers that connect the grid to upper layers). Each feeder offers a set of *connection points*, that is, buses for the connection of sources, loads, storage systems, and downstream transformers. Topologies that are more secure than radial topology emerge when both ends of each feeder attach to upstream transformers [27]; connecting both ends to the same transformer yields a *ring*, whereas connecting the ends to different transformers yields a *thread*. Each ring and thread contains an open switch, which is only closed on demand [30]. Therefore, during normal operation, a grid with ring or thread topology can be seen as a set of radially structured sub-grids, which are separated by open switches. Ring and thread topologies incur higher installation and operation costs than radial topology, and are typically used as a reference for MV grids [28]. Finally, *mesh* topology - the most expensive and most secure topology - is predominant in HV and EHV grids. Mesh grids are characterized by multiple feed-in points and cross-connections forming loops [30]. The majority of switches is

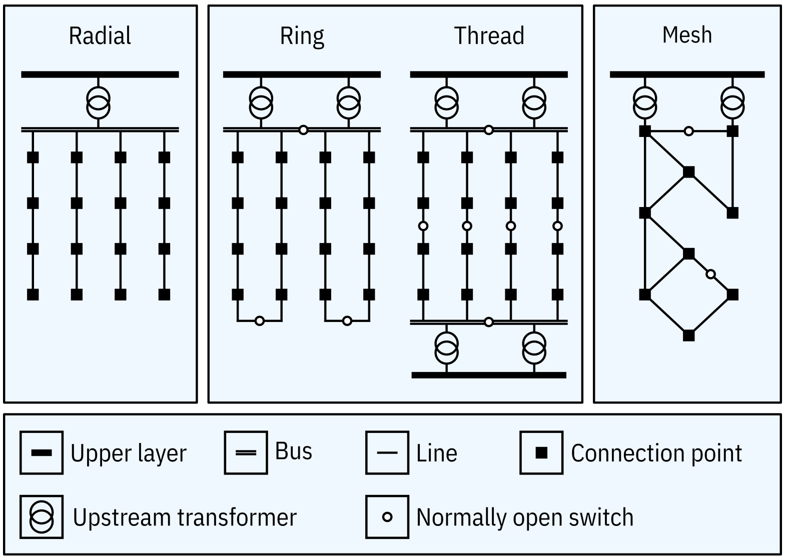


Figure 2.2: Predominant grid topologies in Germany. For simplicity, the figure does not show normally closed switches. Adapted from Verheggen [27]

closed by default [30], so that loads and downstream transformers receive electricity from multiple directions simultaneously.

Although grids are generally planned with specific topologies in mind, they tend to have peculiarities diverging from ideal topology. For example, remote loads (e.g., farms, mountain shelters) are rarely connected to an LV grid via new feeders, but rather via lines that attach to existing feeders. Such lines are known as *stitches* [30].

2.2 Grid Planning

Future-proofing grids is challenging. On one hand, the development of grids involves expensive construction projects with long lead times. As the lifespan of grid components typically lies in the range of several decades, each new component is a long-time investment. On the other hand, grids must be adapted to constantly changing needs of both producers and consumers of electricity. However, supply and demand of electricity is hard to predict, even more its temporal and regional distribution. Consequently, the development of grids must be planned carefully.

2.2.1 Planning Conditions

The choice or implementation of a suitable grid planning method depends on the conditions that underlie the planning process. This subsection gives a short overview of these planning conditions.

Desired Output

First, a grid planner must know what type of output the planning process should yield. Often, the desired output is a *target grid* [26, 27, 35, 36], that is, a specification of how a grid would ideally look like under certain conditions. Another example of a planning output is the optimal timing of construction projects on the basis of triggering events such as a component's end of life [25].

Planning Horizon

Another prerequisite of every grid planning process is the *planning horizon* θ, defined as the period that the grid planner will consider in the planning process. In Germany, for example, grid planning usually involves a long-term and a short-term planning process [27]: First, long-term planning yields a series of target grids, one of which is to be realized within 40-50 years (i.e., $40\,\mathrm{a} \leq \theta \leq 50\,\mathrm{a}$). In a second stage, grid planners consider a much shorter period. The goal of this second stage is to make short-term decisions on how to push forward the transition to the target grid selected in the first planning stage.

Constraints

Furthermore, grid planning methods vary in constraints imposed on planning outcomes. These constraints ensure that all requirements (§ 2.1.2) are met and laws are observed. Table 2.1 shows exemplary constraints and their purposes; in the next chapter, I will specify the actual constraints that I focus on in this thesis.

Table 2.1: Exemplary grid planning constraints and their main purposes

Constrained aspects	Exemplary constraints	Main purposes
Voltage deviation	Voltage must not deviate from the grid's nominal voltage by more than 5% at any bus and at any time.	Stability, power quality
Line loading	The power that flows through a line must never exceed 60% of the line's transmission capacity.	Reliability
Supply shortfall	Blackouts must not last longer than three seconds.	Reliability
Topology	The grid must have ring topology.	Security
Curtailment[1]	The annual curtailment of renewable energy sources must not exceed 1% of generated electricity.	Legality

Degrees of Freedom

In grid planning, many decisions must be taken, such as whether to install a new transformer at a certain site or whether to reinforce a certain line. The number of decisions that must be taken

[1] Curtailment is a ”reduction in the output of a generator from what it could otherwise produce” [37].

corresponds to the planning problem's *degrees of freedom*. A problem with many degrees of freedom is characterized by high complexity, but offers much flexibility for reaching the planning goal. In grid planning, the degrees of freedom depend on the considered options for installing, replacing, reinforcing, reconfiguring, and dismantling grid components. Options that have often been studied in previous research are the installation of new lines, transformers, sources, and controllers; the reinforcement of existing lines and transformers; and the reconfiguration of switches [12, 13].

Input Data

Finally, planning methods differ in the data they require as inputs. Table 2.2 distinguishes six categories of inputs: (1) *Technical* inputs include component specifications and constants needed to define technical constraints. (2) *Structural* inputs consist of topological and geographical data. (3) *Feed-in and consumption* profiles of sources and loads are specified either explicitly (as time series), or are generated using parameterizable component models. (4) A *financial* evaluation of planning outcomes requires prices and fees as inputs. (5) Finally, some planning methods depend on the *timing* of events that occur within the planning horizon.

Table 2.2: Exemplary inputs of grid planning methods, by category

	Categories	Exemplary data points
(1)	Technical	Transmission capacity of line 7; maximum allowed deviation of voltages from the grid's nominal voltage
(2)	Structural	Location of transformer 2; number of installed photovoltaic appliances; components attaching to bus 1
(3)	Feed-in and consumption	Electricity consumption of load 1 between 0:00 a.m. and 0:15 a.m. on January 1, 2030; average solar radiation in August; power of wind farm 2 at wind speed of 50 km/h
(4)	Financial	Market price of transformers of type X; annual maintenance cost per transformer; market price of 1 kWh of electricity on January 1, 2030 at 3:00 a.m.; Compensation for delaying the consumption of 1 kWh by three hours
(5)	Timing	Time of connecting load 9 to the grid; end of life of transformer 3

Except for the last category, data may either describe the status quo (e.g., the current grid structure) or future conditions (e.g., the anticipated number of installed photovoltaic appliances). Some planning methods [27, 36] entirely disregard the status quo, that is, design grids from scratch. For these methods, all inputs describe future conditions. Other methods [25, 35] aim at adapting the existing grid to future conditions, and thus require inputs to describe both the status quo and future conditions.

2.2.2 Power Flow Analysis

Unlike in routable networks (e.g., packet-switched communication networks), where the transmitted objects (e.g., data packets) can be arbitrarily routed over the network, in grids, the flow of electricity follows physical laws. Simulating the flow of electricity in a grid is possible using a technique known as *power flow analysis* (PFA). PFA is an essential part of most—if not all—grid planning tools, because it allows to inspect the state of a grid under a given scenario of grid usage.

Intuitively, PFA considers how much electricity each source feeds into the grid and how much electricity each load draws from the grid in a given moment, and, using this information, determines how the electricity flows through the grid in the given moment. Simulating the flow of electricity for a whole period (e.g., a whole year) requires discretizing the period, that is, modeling the period as a series of time steps (e.g., one time step per hour), and then performing PFA for each time step separately. For each time step, PFA yields the electrical quantities that are required to assess if a grid meets given electrical constraints (e.g., the voltage and line loading constraints given in Table 2.1).

For AC grids, state-of-the-art PFA amounts to solving a system of quadratic-trigonometric equations, whose variables are electrical quantities [38]. Such non-linear equation systems are commonly solved using numerical methods such as the *Newton-Raphson* method or the *Gauss-Seidel* method [38]. As inputs, PFA requires the impedances of lines and transformers, the voltages at sources, and the powers at sources, loads, and transformers [39]. Because the electricity drawn from the grid plus the electricity that gets lost during transmission may surpass the electricity fed into the grid, one source or transformer is chosen as a ”slack component”, which compensates for the difference [30, 39]. The power at this slack component is taken to be unconstrained, and is not an input but an output of PFA. Apart from the power at the slack component, the outputs include the voltages at the grid's buses [38, 39]. These voltages in turn allow to analytically determine the flow of electricity through the grid [38, 39]. Eventually, the loading of each line can be calculated as the power that passes through the line divided by the maximum power that may pass through the line (i.e., the line's transmission capacity).

In the remainder of this thesis, I will abstract from the technical underpinnings of PFA to ease reading—especially for readers without background in power engineering. From now on, PFA will appear as a black box, which receives electrical quantities as inputs and outputs additional electrical quantities. The only thing needed to remember is that the outputs of PFA are needed to check if voltages or line loadings exceed critical values.

2.2.3 Optimization Models and Methods

Most scholars approach grid planning as an optimization problem [12, 13]. Adopting the notation of Dorigo and Stützle [40], an optimization problem Π is a triple (S, f, Ω) where S is a set of candidate solutions, f is a *cost function* assigning a cost $f(s)$ to each $s \in S$, and Ω is a set of *constraints* on S. If $s \in S$ satisfies all constraints in Ω, s is called *feasible*. Denoting the set of feasible solutions by $\tilde{S}$, the goal is to find solutions $s^* \in \tilde{S}$ with $f(s^*) \leq f(s)$ for all $s \in \tilde{S}$.

The task of a grid planner is to derive Π from given planning conditions (§ 2.2.1). This task involves a trade-off between model accuracy and optimization performance [35, 41]. If the planner makes many simplifying assumptions, optimization tends to be fast, and tends to find optimal or close-to-optimal solutions with respect to the model. However, the model may oversimplify the planning problem. Conversely, if the given planning conditions are modeled in detail, the model better reflects the planning case, but optimization methods tend to become computationally infeasible, or fail to find close-to-optimal solutions. A variety of modeling decisions mirror this trade-off [41], including the following:

Static ↔ dynamic The simplest possible model disregards any dynamics within the planning horizon. In more accurate models, not only the set of actions to perform is subject to optimization but also their timing. In the most realistic case, time is modeled as a continuous period representing the full planning horizon.

Deterministic ↔ stochastic Various methods are available to model uncertainty of future conditions (e.g., Bayesian methods, Monte Carlo simulation, fuzzy set theory). In grid planning, the traditional method is to specify multiple scenarios and assign to each scenario the probability that it will become reality [25, 27, 41]. Each scenario describes conditions at some time within the planning horizon, and the probabilities of scenarios describing the same time must add up to one. In the simplest case, only a single scenario is considered, that is, uncertainty is not modeled at all.

Single-objective ↔ multi-objective In the early years of grid planning, cost function f was always a scalar function, usually measuring financial cost [41]. Today, *multi-objective* optimization methods exist (e.g., Pareto-optimal methods, outranking methods), which allow to consider multiple objectives simultaneously. In multi-objective optimization, $f(s)$ is a vector for $s \in S$, and each vector component measures a different quantity. For example, f_1 would measure financial cost and f_2 ecological cost.

Linear ↔ non-linear Many aspects of grid planning models can introduce non-linearity. Typical sources of non-linearity are the cost function [26, 41], boolean variables (e.g., to model the state of switches), and PFA (§ 2.2.2) [26, 42].

For sufficiently simple models or small problem sizes, exact optimization methods such as *cutting plane* and *branch-and-bound* methods are efficient solution methods. Exact optimization methods find minimum-cost solutions or solutions whose cost is within some band around the global minimum [36]. However, real-world problem instances are generally too large or the underlying models too complex for exact methods to be feasible [35, 36, 41]. Instead, heuristic methods have become state of the art in grid planning, particularly *genetic algorithms* and *particle swarm optimization* [12, 13].

3 Problem Formulation

Having established technical foundations in the previous chapter, I now specify the problem that I focus on in this thesis. First, I introduce a formal model of grids (§ 3.1). Then, using this model, I formalize the objective (§ 3.2), the degrees of freedom (§ 3.3), the constraints (§ 3.4), and the required inputs (§ 3.5). Finally, I summarize the problem's characteristics (§ 3.6).

3.1 Grid Model

I begin by introducing a simple way of modeling grids, inspired by the data model of the software package *PyPSA* [39]. I model a grid as an undirected graph $G = (B, E)$, whose nodes B are the grid's buses and whose edges $E := L \cup W$ are the grid's line segments L and switches W. For ease of implementation (§ 4), I represent each switch by a pseudo line segment, which is only present if the represented switch is closed. Furthermore, I do not explicitly model sources, loads, storage systems, and transformers. Instead, I model these components implicitly in terms of their aggregated electricity feed-in and consumption at each bus $b \in B$; the next paragraph will give the details. Figure 3.1 visualizes the graph $G = (B, E)$ of an exemplary LV grid. The set B contains the grid's buses $b_1, \dots, b_{16}$, and the set $E = L \cup W$ the line segments $l_1, \dots, l_9 \in L$ and closed switches $w_1, \dots, w_5 \in W$. Each switch links an MV-LV transformer (i.e., a transformer that connects an MV grid to an LV grid) with a feeder $F \in \{\{l_1, l_2, l_3\}, \{l_5\}, \{l_6\}, \{l_7, l_8\}, \{l_9\}\}$. With that, the grid has radial topology (§ 2.1.3), except that it contains a stitch, which is represented by $\{l_4\}$.

The nodes and edges of G have several attributes, which fall into two categories: time-invariant and time-dependent. The time-invariant attributes are necessary to model characteristics of line segments and switches. Each line segment $l \in L$ has two time-invariant attributes: a length $|l|$ and a type $z \in Z$, where Z is the set of all considered types. The type in turn specifies a line segment's impedance (i.e., the electrical opposition to the flow of power through the line segment) and transmission capacity (i.e., the maximum power that may pass through the line segment). A (closed) switch $w \in W$ is a pseudo line segment of zero length, negligible impedance,[1] and infinite transmission capacity. Unlike time-invariant attributes, time-dependent attributes model the temporal state of the grid. I model time as a series of moments $(t_1, t_2, \dots, t_m)$, $m \geq 1$. The only time-dependent attribute of line segments $l \in L$ is the *power flow* $p_l = (p_l(t_1), p_l(t_2), \dots, p_l(t_m))$, where $p_l(t_i)$ denotes the power that flows through l at time t_i. Buses $b \in B$ have two time-dependent attributes. The first is the *bus voltage* $v_b = (v_b(t_1), v_b(t_2), \dots, v_b(t_m))$, where $v_b(t_i)$ denotes the voltage at b and t_i. The second attribute is a *feed-in and consumption profile* $p_b = (p_b(t_1), p_b(t_2), \dots, p_b(t_m))$, where $p_b(t_i)$ denotes the power at b and t_i. If $p_b(t_i)$ is positive, the components attaching to b in sum feed in more

[1] I do not model switches as components of zero impedance, because zero-impedance components can cause numerical problems in PFA. Section § 4.5.1 gives further information.

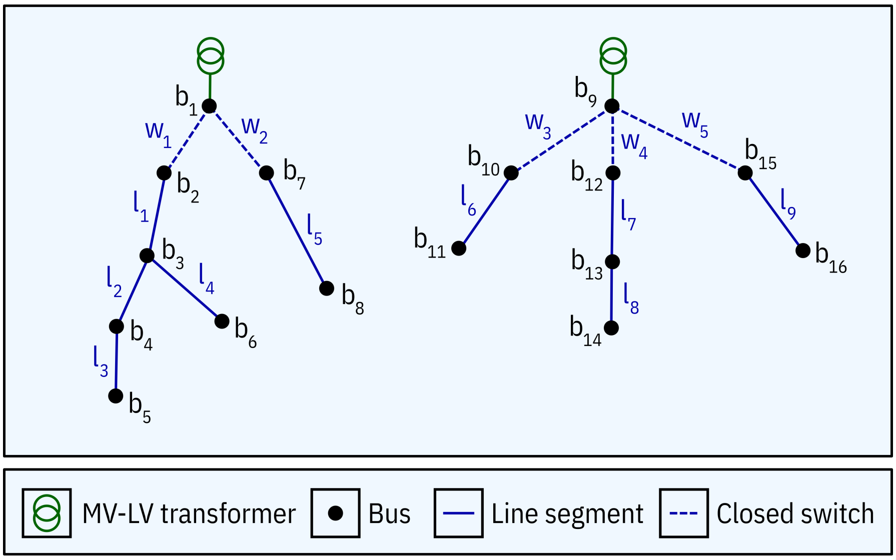


Figure 3.1: Model of an LV grid

electricity than they consume at t_i, and vice versa. With that, p_b captures the aggregated impact of all sources, loads, storage systems, controllers, and transformers that attach to b.

3.2 Objective

The goal of the problem that I consider in this thesis is to expand a given LV grid such that the expanded grid satisfies a given set of constraints at minimum cost. Formally, given an LV grid $G_{\text{now}} = (B, E_{\text{now}})$ and a set E_{add} of options for expanding the grid, the desired outcome is a target grid $G^* = (B, E^*)$, $E^* \subseteq E_{\text{now}} \cup E_{\text{add}}$, and the cost $c_{\text{exp}}(G_{\text{now}}, G^*)$ of transitioning from G_{now} to G^*. G^* represents the target structure of G_{now} at time θ (with reasonable values for θ ranging from one year to about 20 years), and ideally, $c_{\text{exp}}(G_{\text{now}}, G^*) \leq c_{\text{exp}}(G_{\text{now}}, G)$ for all candidate target grids G.

The expansion cost $c_{\text{exp}}(G_{\text{now}}, G^*)$ sums up the costs of all grid modifications that are involved in the transition from G_{now} to G^*. I define the costs of grid modifications as follows, denoting by $c_{\text{cab}}(z)$ the per-unit market price of cable type $z \in Z$ and by $c_{\text{ins}}(z)$ the per-unit installation cost for lines of type $z \in Z$:

1. Installing or replacing a line segment l of type z and length $|l|$ costs $|l| \cdot [c_{\text{ins}}(z) + c_{\text{cab}}(z)]$.
2. Dismantling a line segment l of type z and length $|l|$ costs $|l| \cdot c_{\text{ins}}(z)$.
3. Operating a switch costs c_{swi}.

Throughout this thesis, I specify costs in terms of one-time expenses rather than annuity repayment, because—given the long lifespan of grid components—most components will reach their end of life long after the planning horizon θ.

3.3 Degrees of Freedom

Next, I specify the options that I consider for modifying the given grid $G_{\mathrm{now}} = (B, L_{\mathrm{now}} \cup W_{\mathrm{now}})$ to yield an expanded grid $G_{\mathrm{exp}} = (B, L_{\mathrm{exp}} \cup W_{\mathrm{exp}})$ with $L_{\mathrm{exp}} \subseteq L_{\mathrm{now}} \cup L_{\mathrm{add}}$ and $W_{\mathrm{exp}} \subseteq W_{\mathrm{now}} \cup W_{\mathrm{add}}$. I allow five types of grid modifications:

1. The installation of a line segment $l_{\mathrm{add}} \in L_{\mathrm{add}}$ at a site where no line is installed yet (l_{add} connects two buses that are not connected by any $l_{\mathrm{now}} \in L_{\mathrm{now}}$)
2. The replacement of a line segment $l_{\mathrm{now}} \in L_{\mathrm{now}}$ by a line segment $l_{\mathrm{add}} \in L_{\mathrm{add}}$ with different characteristics (l_{add} connects the same buses as l_{now})
3. The dismantling of a line segment $l_{\mathrm{now}} \in L_{\mathrm{now}}$
4. The opening of a closed switch $w_{\mathrm{now}} \in W_{\mathrm{now}}$, that is, the removal of the pseudo line w_{now} representing the closed switch
5. The closing of an open switch, that is, the construction of the pseudo line $w_{\mathrm{add}} \in W_{\mathrm{add}}$ representing the closed switch

Each of these modifications forces electricity into different routes, and can thus help to balance load across the grid. Additionally, the installation and the reinforcement of lines locally increases the amount of electricity that can pass through the grid.

3.4 Constraints

After applying modifications to grid G_{now}, the resulting expanded grid G_{exp} must satisfy several constraints, which fall into two categories: electrical and topological.

3.4.1 Electrical Constraints

The electrical constraints impose limits on bus voltages (i.e., the voltages at the grid's buses) and power flows (i.e., the powers that flow through the grid's line segments). By doing so, the constraints ensure that the grid is able to cope with the strain that grid usage is expected to put on the grid at the end of the planning horizon θ.

To simulate this strain, I require an estimation of the power $p_b(t_i)$ at each of the grid's connection points $b \in \{b_1, \dots, b_r\} \subset B$ for moments $t_i \approx \theta, i = 1, \dots, m, m \geq 1$. Notably, I do not require $p_b(t_i)$ for buses to which upstream transformers attach, because I consider the amount of electricity drawn from or fed into upper grid layers to be unconstrained. In analogy to the terminology of the PyPSA software library, I call $p(t_i) := (p_{b_1}(t_i), \dots, p_{b_r}(t_i))$ a *snapshot*. The more snapshots are available, the better they collectively estimate the strain put on the grid around time θ.

The snapshots allow to simulate the flow of electricity through the grid using PFA (§ 2.2.2). For each moment $t_i, i \in \{1, \dots, m\}$, each bus $b \in B$, and each line segment $l \in L_{\mathrm{exp}}$, PFA outputs the bus voltage $v_b(t_i)$ and power flow $p_l(t_i)$.

The bus voltages and power flows are subject to the following constraints:

$$\begin{aligned} &\omega_b(t_i): \; v_{\min} \leq v_b(t_i) \leq v_{\max} \\ &\psi_l(t_i): \; p_l(t_i) \leq \lambda \hat{p}_l \end{aligned}$$

where $v_{\min} \in \mathbb{R}^{\geq 0}$, $v_{\max} \in \mathbb{R}^{\geq 0}$, and $\lambda \in [0, 1]$ are parameters. For $\omega_b(t_i)$ to hold true, the bus voltage $v_b(t_i)$ must be in the interval $[v_{\min}, v_{\max}]$. For $\psi_l(t_i)$ to hold true, the power flow $p_l(t_i)$ must not exceed a fraction λ of line segment l's transmission capacity $\hat{p}_l$; that is, the loading $p_l(t_i) \;/\; \hat{p}_l$ of l must not exceed λ. In short, for all electrical constraints to be satisfied, the bus voltages and line loadings must always stay within their given limits during PFA.

3.4.2 Topological Constraints

The expanded grid must not only satisfy voltage and line loading constraints, but must also have a certain topology. In this thesis, I demand that the grid has radial topology (§ 2.1.3)—a common requirement for LV grids. However, I make two exceptions. First, the grid may contain rings and threads (for increased security); and second, it may contain stitches of arbitrary length and branching (to reduce investment costs).

In the following, I formalize this topological constraint using basic graph theory. To recall, in radial grids, all lines branch out from the upstream transformers, and the lines neither form loops nor do they connect upstream transformers with each other. In graph theoretical terms, the buses to which the upstream transformers attach can be seen as the roots of trees, and each of the trees' branches represents a line. In other words, the graph of a radial grid is a forest, where each tree contains exactly one upstream transformer.

This characterization is still valid when the grid contains rings, threads, or stitches. To recall, each ring and thread contains an open switch; and open switches disappear in graph representation. Consequently, the graph of a radial grid with rings or threads still does not contain loops or connections between upstream transformers. The same holds if the grid contains stitches of arbitrary length and branching, because in graph representation, such stitches are just sub-trees of other trees.

3.5 Required Inputs

The previous sections imply which input data are needed to solve the presented planning problem. To summarize, the required inputs are:

1. The graph $G_{\text{now}} = (B, L_{\text{now}} \cup W_{\text{now}})$ of the grid that is to be expanded, including the length $|l|$ and type z_l of each line segment $l \in L_{\text{now}}$
2. A set $E_{\text{add}} = L_{\text{add}} \cup W_{\text{add}}$ of grid expansion options, including the length $|l|$ and type z_l of each line segment $l \in L_{\text{add}}$

3. The installation cost $c_{\mathrm{ins}}(z)$ and cable cost $c_{\mathrm{cab}}(z)$ for each line type $z \in Z$, where Z contains the types of all line segments in $L_{\mathrm{now}} \cup L_{\mathrm{add}}$; and the cost c_{swi} of operating a switch
4. Snapshots $p(t_i)$ for moments $t_i \approx \theta, i = 1, \dots, m, m \geq 1$, where θ is the planning horizon
5. Constants v_{min} and v_{max}, which limit the allowed range of bus voltages; and a constant λ, which specifies the maximum allowed line loading

3.6 Problem Characteristics

To conclude, the planning conditions specified in the previous sections induce an optimization model with the following properties:

1. It is static, because it does not consider dynamics within the planning horizon.[2]
2. It is deterministic, as it does not model uncertainty.[3]
3. It is single-objective, as the only objective is to minimize the financial cost of grid expansion.
4. It is non-linear, because state-of-the-art PFA involves solving equations that contain quadratic and trigonometric terms [38].

[2] The optimization model being static does not contradict the fact that it takes time series as inputs; this is because the time series do not describe the period from now until planning horizon θ but the time *right at the end* of the planning horizon, that is, the time around θ.

[3] Implicitly, however, the optimization model accounts for uncertainty of the amount of strain resting on the grid around time θ if the planner provides multiple scenarios (in form of snapshots) of grid usage around time θ.

4 Implementation

In this chapter, I present *AntPower*, a software tool that solves the grid expansion problem specified in the previous chapter.

4.1 Basic Software Characteristics

AntPower is a tested, easy-to-use, and cross-platform software tool that facilitates the expansion planning of LV grids. The tool runs on any system on which a Docker environment is installed. AntPower is a Python application, and its algorithmic core (i.e., everything excluding preprocessing, logging, and visualization) comprises 506 statements, 95% of which are covered by unit tests. Reaching a significantly higher test coverage is difficult due to the tool's probabilistic nature.

4.2 Ant Colony Optimization

AntPower implements *ant colony optimization* (ACO), a method that outperformed other methods in previous grid planning research [17–22]. ACO is a meta-heuristic, that is, ”a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems” [40]. It can be applied to optimization problem $\Pi = (S, f, \Omega)$ if [40]:

1. $S \subseteq X$, where X is a set of finite sequences over a finite set of *components* C. The elements of X are called *states* of Π.

2. A problem-dependent test defines a set of *viable* states $\tilde{X}$ with $\tilde{X} \subseteq X$.

3. Let $\tilde{S}$ be the set of feasible solutions (defined in terms of the set of constraints Ω). For the set of optimal solutions S^*, it holds that $S^* \neq \emptyset$, $S^* \subseteq \tilde{S}$, and $S^* \subseteq \tilde{X}$.

4. f is defined (not only on $S \subseteq X$ but) on X.

In ACO, agents called *ants* construct candidate solutions $s \in S$ in parallel. The construction procedure is a randomized walk on the *construction graph* $G_C = (C, C \times C)$. This procedure works as follows. Consider an ant is in state $x_r = \langle x_{r-1}, c_i \rangle \in \tilde{X}$, that is, it resides on $c_i \in C$. If no termination condition holds, the ant moves to some $c_j \in C$ in its *neighborhood* $N(x_r)$ according to a probabilistic *transition rule*. Afterwards, the ant's state is $\langle x_r, c_j \rangle \in \tilde{X}$. The only way for ants to communicate is through *pheromones*. Pheromones are global and dynamic values attached to the nodes of G_C. After an ant constructs a solution $s \in S$, it may update pheromones as a function of $f(s)$. The transition rule in turn is a function of the pheromones and, optionally, static heuristics. Typically, an ant deposits pheromones on visited components after it constructed a low-cost solution, thereby increasing the probability that ants will visit these components in future walks. Ideally, this mechanism leads to a situation where ants initially explore large areas of G_C (because the distribution of pheromones is roughly uniform) but increasingly exploit the collective knowledge stored as pheromones to converge to optimal solutions. Dorigo and Stützle [40] and Blum [43] give detailed introductions to ACO and its many variants.

4.3 Selection of Libraries

Central to AntPower is PyPSA [39], a free and open-source Python library maintained by researchers of the Karlsruhe Institute of Technology, Germany. PyPSA contributes to AntPower in two ways. First, it provides a data structure for accurate grid modeling. Second, it implements state-of-the-art PFA algorithms, one of which AntPower uses to determine if candidate target grids observe voltage and line loading limits.[1]

In addition, AntPower depends on the following libraries, which are well-known in the field of scientific computing: *NumPy*, *pandas*, and *NetworkX* for data representation and manipulation; and *matplotlib* and *plotly* for data visualization.

Although several Python libraries exist that implement ACO[2], I use none of them; instead, I implement ACO from scratch. The reason is that the ACO libraries significantly lack generality; probably for historical reasons, all of the ACO libraries implement ACO specifically for the *traveling salesman problem* (TSP), a well-studied problem in operations research and computer science [40]. As a result, the ACO libraries are strongly tied to the TSP, which strongly differs from the grid expansion problem AntPower seeks to solve.

4.4 Algorithm

The algorithm AntPower implements is an adaptation of the ACO variant *Ant Colony System* (ACS) [44]. The reason for basing AntPower on ACS (instead of basing it on ”pure” ACO) is the following. ACO, being a meta-heuristic, is extremely flexible; and the downside of this flexibility is that implementation involves making many difficult decisions. To limit the number of decisions that must be taken, scholars developed variants of ACO whose optimization frameworks are less generic than that of pure ACO. One of these ACO variants is ACS. I base the implementation of AntPower on ACS, because ACS performed well on grid planning problems in previous research [26, 27]. I leave for future work the application of other ACO variants to grid planning problems.

[1] The PFA algorithm that is used in AntPower employs the Newton-Raphson method to numerically solve a system of quadratic-trigonometric equations, whose variables are electrical quantities. PyPSA’s documentation formally describes the PFA algorithm at: https://pypsa.readthedocs.io/en/latest/power_flow.html#full-non-linear-power-flow

[2] On October 15, 2020, a keyword search for libraries implementing ACO (keywords: ”ant”, ”ant colony optimization”, ”aco”; websites: google.com, pypi.org, anaconda.org, and github.com) yielded eight results: ”ACOpy” (https://github.com/rhgrant10/acopy); ”swarmlib” (https://github.com/HaaLeo/swarmlib); ”PYaco” (https://github.com/Ganariya/PyACO); ”ant-colony” (https://github.com/jurekpawlikowski/ant-colony); ”randomized-tsp” (https://github.com/akshatkarani/randomized_tsp); ”scikit-opt” (https://github.com/guofei9987/scikit-opt); ”AntColonyOptimization” (https://github.com/Akavall/AntColonyOptimization); ”ant-colony-optimization” (https://github.com/pjmattingly/ant-colony-optimization).

Algorithm 1 shows how AntPower works on an abstract level. In lines 2-6, k_{col} groups of ants, called *colonies*, independently search for the cheapest solutions they can find. Eventually, in line 7, AntPower returns the cheapest of all found solutions along with this solution's cost. While $k_{\mathrm{col}} = 1$ in ACS, $k_{\mathrm{col}} \geq 1$ in AntPower. Each colony runs in a separate process, so that setting $k_{\mathrm{col}} > 1$ allows to run multiple colonies in parallel. The colonies are isolated from each other; therefore, running AntPower with $k_{\mathrm{col}} = 2$ yields the same results as running AntPower with $k_{\mathrm{col}} = 1$ twice (except for differences caused by random effects). A promising direction for future work is to implement communication between colonies. For example, the colonies could from time to time exchange the pheromone trails they have produced [45].

Algorithm 1 High-level implementation of AntPower

```
1: **procedure** runAntPower
2:     bestFoundSolutions ← ∅
3:     **repeat** $k_{\mathrm{col}}$ **times**
4:         bestFoundSolutions ← bestFoundSolutions ∪ searchForBestSolution()
5:     **end repeat**
6:     bestFoundSolution ← argmin($f$, bestFoundSolutions)
7:     **return** bestFoundSolution, $f$(bestFoundSolution)
8: **end procedure**
```

Algorithm 2 outlines the implementation of colonies in AntPower. First, in line 2, the procedure assigns to each solution component $c_i \in C$ a pheromone value $\tau_i \in \mathbb{R}$, and sets each τ_i to τ_0, where τ_0 is a parameter of AntPower. Then, the colony iteratively searches for solutions (lines 3-16). In each of the k_{itr} iterations, the colony's ants construct solutions $s_i \in S$, $i = 1, \dots, k_{\mathrm{ant}}$ (lines 5-9). In the process of constructing solutions (which I will elaborate later) the ants reduce the pheromone values of the solution components they visit (line 7). The purpose of this pheromone reduction is to discourage ants from visiting the same components over and over again [40]. After all ants constructed solutions, the colony selects from the constructed solutions a minimum cost solution (line 10). If the cost of this *iteration-best solution* is below that of the best solution found so far (line 11), the colony remembers the iteration-best solution as the *best-so-far solution* (lines 12-13). Next, it increases the pheromone values of all components contained in the best-so-far solution (line 15), thus encouraging ants to construct solutions that share components with the best-so-far solution. After the final iteration, the procedure returns the best-so-far solution (line 17). In sum, this implementation of colonies equals that of ACS, except for a minor simplification; whereas in ACS, ants construct solutions in parallel, in AntPower, they construct solutions sequentially (lines 6-9).

Algorithm 3 zooms in on the implementation of ants. Starting with state $x \leftarrow \langle\rangle \in X$, an ant iteratively selects components from its neighborhood $N(x)$ and appends them to x, until $x \in S$ (lines 2-7). The decision of which component to select next depends on the (dynamic) vector of pheromone values $\tau := (\tau_1, \dots, \tau_n)$, a (static) vector of *heuristic values* $\eta := (\eta_1, \dots, \eta_n)$, and chance; the higher τ_i and the higher η_i, the higher the probability that an ant selects $c_i \in N(x)$. Each time an ant visits a component c_i (i.e., adds c_i to x), it decreases τ_i (line 6); thereby it incentivizes all ants to visit components $c \in C \backslash \{c_i\}$. Eventually, the procedure of constructing solutions returns the constructed solution $x \in S$ and the updated vector of pheromone values τ (line 8).

Algorithm 2 Implementation of colonies in AntPower

```
procedure searchForBestSolution()
    $\tau \leftarrow (\tau_1, \tau_2, \dots, \tau_n) \leftarrow (\tau_0, \tau_0, \dots, \tau_0)$
    lowestCostSoFar $\leftarrow \infty$
    repeat $k_{\mathrm{itr}}$ times
        solutions $\leftarrow \varnothing$
        repeat $k_{\mathrm{ant}}$ times
            solution, $\tau \leftarrow$ constructSolution($\tau$)
            solutions $\leftarrow$ solutions $\cup$ solution
        end repeat
        iterationBestSolution $\leftarrow$ argmin($f$, solutions)
        if $f$(iterationBestSolution) $<$ lowestCostSoFar then
            lowestCostSoFar $\leftarrow f$(iterationBestSolution)
            bestSolutionSoFar $\leftarrow$ iterationBestSolution
        end if
        $\tau \leftarrow$ updatePheromones($\tau$, bestSolutionSoFar)
    end repeat
    return bestSolutionSoFar
end procedure
```

Algorithm 3 Implementation of ants in AntPower

```
procedure constructSolution($\tau$)
    $x \leftarrow \langle\rangle$
    while $\neg$ isSolution($x$) do
        $c \leftarrow$ selectBestComponent($N(x)$, $\tau$, $\eta$)
        $x \leftarrow \langle x, c \rangle$
        $\tau \leftarrow$ updatePheromones($\tau$, $c$)
    end while
    return $x$, $\tau$
end procedure
```

Although, eventually, AntPower is to return a feasible solution, neither Algorithm 1, Algorithm 2, nor Algorithm 3 assesses the feasibility of a found solution. The reason is that I will later define cost function f such that $f(s)$ reflects whether $s \in S$ is feasible or not. Concretely, I will penalize infeasible solutions to incentivize the construction of feasible solutions. Chapter § 5 will show that even a simple penalty term can efficiently guide ants toward feasible solutions.

In terms of computational complexity, two parts of AntPower are significant: the underlying PFA algorithm (§ 4.3) and the solution construction algorithm (Algorithm 3). The complexity of the PFA algorithm results from the need to solve a system of equations $Ax = 0$, where A is a square matrix whose number of rows is in the order of $|B|$ (i.e., the number of buses). The (worst-case) time complexity of solving $Ax = 0$ is $O(|B|^3)$; and in each of AntPower's k_{itr} iterations, each of the $k_{\mathrm{col}} \cdot k_{\mathrm{ant}}$ ants performs PFA for each of the m snapshots. Therefore, PFA contributes $O(k_{\mathrm{itr}}k_{\mathrm{col}}k_{\mathrm{ant}}m|B|^3)$ to AntPower's time complexity. Additionally, the solution construction algorithm—that is, the implementation of ants—adds a significant amount of complexity. In the worst case, a constructed solution comprises all n solution components, in which case the respective ant must make n decisions on which component to visit next. Each of these decisions involves iterating over up to $|B|$ buses and n components; thus the time complexity of the solution construction algorithm is $O\left(n(|B| + n)\right)$. As this algorithm runs $k_{\mathrm{itr}} \cdot k_{\mathrm{col}} \cdot k_{\mathrm{ant}}$ times, the construction of solutions contributes $O(k_{\mathrm{itr}}k_{\mathrm{col}}k_{\mathrm{ant}}n(|B| + n))$ to AntPower's time complexity. With that, the overall time complexity of AntPower is $O(k_{\mathrm{itr}}k_{\mathrm{col}}k_{\mathrm{ant}}(m|B|^3 + n|B| + n^2))$. Regarding space complexity, AntPower needs to store $|B|$ buses, up to n solution components for each of the $k_{\mathrm{col}} \cdot k_{\mathrm{ant}}$ ants, n pheromone values for each of the k_{col} colonies, and n heuristic values. In sum, these requirements imply a space complexity of $O(|B| + k_{\mathrm{col}}k_{\mathrm{ant}}n)$.

4.5 Design Decisions

Implementing the algorithm presented in the previous section involves three decisions. First, I must decide for a data model that enables an accurate representation of grids. Second, I must decide how I translate the specification of the given planning problem to the methodological framework of ACO; more precisely, I must define C, X, $\tilde{X}$, S, $\tilde{S}$, Ω, and f. Third, I must decide for a search strategy; more precisely, I must define the ants' neighborhood, transition rule, pheromone update mechanism, and heuristic values. The following three subsections document the decisions taken in the process of implementing AntPower: section § 4.5.1 discusses AntPower's data model; section § 4.5.2 shows how AntPower translates the planning conditions to the ACO framework; and section § 4.5.3 presents the search strategy implemented by AntPower.

4.5.1 Data Model

AntPower's data model must be able to accurately represent the components, structure, and states of a grid. PyPSA's data model[3] fulfills these criteria: it features parameterizable models of grid components; it allows to model grids of arbitrary structure; and, for each grid component, it stores values of electric quantities that together constitute the state of the modeled grid. Therefore, I adopt PyPSA's data model for the implementation of AntPower.

A limitation of PyPSA's current version (0.17.1) is that it lacks a model of switches; thus it does not allow specifying the locations and states of a grid's switches. I circumvent this limitation by modeling each switch as a low-impedance line segment that is present only if the switch is in closed state. Typically, grid planners regard switches as components of zero impedance [28, 30]. However, zero-impedance components can cause numerical problems in PFA. Therefore, I set the impedance of line segments that represent switches to 0.1 µΩ, which is the lowest value that did not cause numerical problems in my experiments.

[3] PyPSA's documentation comprehensively describes the data model at: https://pypsa.readthedocs.io/en/latest/components.html

4.5.2 Translation of Planning Conditions to ACO

The next step in implementing AntPower is the translation of planning conditions to ACO's methodological framework. This translation amounts to stating optimization problem Π, which in turn amounts to defining C, X, $\tilde{X}$, S, $\tilde{S}$, Ω, and f.

Solution Components (C)

I begin by defining the set of solution components C. Considering that the planning objective is to find E^*, where E^* is a subset of $E_{\text{now}} \cup E_{\text{add}}$, a straight-forward approach is to set $C = E_{\text{now}} \cup E_{\text{add}}$. AntPower takes this approach.

Before I continue to define the other parts of Π, I try an approach for reducing the size of C, that is, reducing the number of components. Reducing the number of components dramatically reduces the number of solutions, considering that I will later define solutions as sequences over C. With a reduced number of solutions, ants can construct a larger fraction of solutions, which in turn increases the probability of finding high-quality solutions.

The presented approach for reducing the number of components is based on the observation that in LV grids, lines often run along streets, and connect multiple buildings in each street. In graph representation (§ 3.1), a line running along a street corresponds to a series of consecutive edges (line segments), which are connected by nodes (buses) of degree two, and the buses are the connection points of the buildings along the street. Instead of considering each segment as an individual component, the idea is to de-segment the lines, that is, to aggregate a line's segments into a single solution component.

In the exemplary problem shown in Figure 4.1, such de-segmentation yields six components ($c_1, \dots, c_6$), and thus reduces the number of components by 40%, as compared to the approach of considering each segment to be an individual component. Notably, the three segments in the upper-right corner form a cycle; therefore, grids that contain all of the three segments have invalid topology. Consequently, as solution components are the atomic building blocks of solutions, the three segments must constitute at least two components (c_4 and c_5 in the example) for ants to be able to create feasible solutions.

Although, at first glance, de-segmentation of lines is a promising preprocessing step, the approach turns out to be incompatible with the requirement for radial topology, given how ants are implemented in AntPower. To recall, ants iteratively select components $c \in C$ from their neighborhood $N(x) \subseteq \tilde{X}$, and append them to their state $x \in \tilde{X}$. I will later define the set of viable states $\tilde{X}$ to be the set of states that represent topologically valid grids. To adhere to this definition, ants must begin the process of constructing a solution at the upstream transformers; for example, in Figure 4.1, ants must begin by selecting either c_1 or c_4. If an ant selects c_1 (i.e., $x \leftarrow \langle c_1 \rangle$), it must then select c_2, c_3, or c_4 (i.e., $N(x) = \{c_2, c_3, c_4\}$), because these components are the only components that preserve valid topology. An issue arises after an ant selects all components except c_3 and c_5 (i.e., $x \leftarrow \langle c_1, c_2, c_4, c_6 \rangle$). At this point, the ant's neighborhood is empty (i.e., $N(x) = \emptyset$), because the ant has already selected all components except c_3 and c_5, and selecting c_3 or c_5 would yield a topologically invalid grid (because c_3 would connect the two

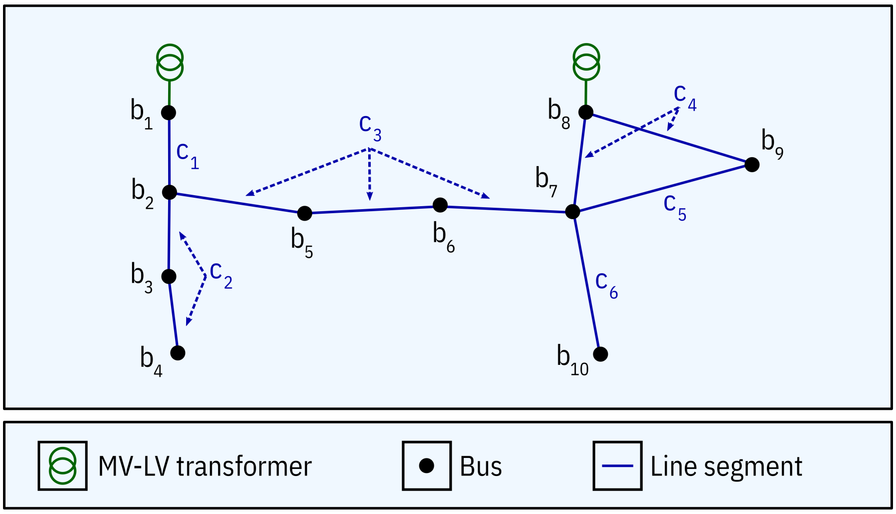


Figure 4.1: De-segmentation of line segments

MV-LV transformers, and c_5 would form a cycle with c_4). At this stage, the ant has not yet found a solution (i.e., $x \notin S$) if we suppose that, for x to be a solution, the components in x must connect each of the buses $b_1, \dots, b_{10}$ with an MV-LV transformer. In this situation, all an ant can do is to start from scratch (i.e., $x \leftarrow \langle\rangle$). In real-world problems, this issue can be such prevalent that it prevents ants from finding any solution at all. However, discarding the de-segmentation step resolves the issue.

As the de-segmentation approach fails, I stick to the original approach of considering each line segment to be a separate solution component. That is, I set $C = E_{\mathrm{now}} \cup E_{\mathrm{add}}$, so that a solution component is either a line segment $l \in L_{\mathrm{now}} \cup L_{\mathrm{add}}$ or a closed switch $w \in W_{\mathrm{now}} \cup W_{\mathrm{add}}$.

States (X, $\tilde{X}$)

Next, I define the set of states X and the set of viable states $\tilde{X}$. First, I define X to be the set of all sequences over C that contain each $c \in C$ at most once. I exclude sequences that contain any component more than once to ensure that ants visit each component at most once. Second, I define $\tilde{X}$ as the set of all states that represent topologically valid grids. Considering that ants can only reach states $x \in \tilde{X}$, the given definition prevents ants from visiting states that represent topologically invalid grids. To recall from section § 3.4.2, the graph G of a topologically valid grid is a forest, where each tree contains exactly one upstream transformer. Accordingly, denoting by E the set of components contained in $x \in X$, x is viable ($x \in \tilde{X}$) if and only if $G = (B, E)$ is a forest where each tree contains exactly one upstream transformer.

Solutions (S, $\tilde{S}$)

Having defined X, I can now specify the set of solutions $S \subseteq X$. I define S to be the set of all states $x \in X$ that have the following property P: *If E denotes the set of components appearing in x, $G = (B, E)$ connects each connection point with an upstream transformer.* The rationale is the following. If x does not have property P, at least one connection point is isolated from the grid. Therefore, the ant whose state is x should append components to x until x has property P, that is, until all connection points are connected to the grid. Because the ant stops the process of appending components to x as soon as $x \in S$, and because the ant should stop this process as soon as x has property P, I set $S := \{x \in X \mid x \text{ has property } P\}$.

In addition, I must define the set of feasible solutions $\tilde{S}$. To recall, a solution is feasible if it satisfies all constraints $\omega \in \Omega$. Consequently, the definition of $\tilde{S}$ follows from Ω, which in turn is defined as follows.

Constraints (Ω)

The set of constraints Ω simply consists of the voltage and line loading constraints (§ 3.4.1), which are part of the problem specification. The problem specification additionally includes a topological constraint (which asserts that grids have radial topology, except that they may contain rings, threads, and stitches); yet, this constraint is not part of Ω, because it would be redundant; given the definition of $\tilde{X}$, ants can only construct topologically valid grids anyway.

Costs (f)

To complete the definition of $\Pi = (S, f, \Omega)$, the only thing left to do is to specify cost function $f : X \to \mathbb{R}_{\geq 0}$. Because the optimization objective is to find a grid G^* such that the cost of transitioning from G_{now} to G^* is minimal, $f(x)$ should reflect the cost of transitioning from G_{now} to $G(x)$, where $G(x)$ is the grid that x represents. I approximate this cost by $c_{\text{exp}}(G_{\text{now}}, G(x))$, which sums up the costs of all grid modifications that are involved in the transition from G_{now} to $G(x)$. The formal definition of c_{exp} is part of the problem specification (§ 3.2).

To guide ants toward feasible solutions, I add to the definition of f a term P that penalizes infeasible solutions depending on the degree of their infeasibility. For simplicity, I set this penalty to be proportional to the number of violated constraints:

$$P(x) = \hat{f}(s^*)\, k_{\text{vio}}(x)$$

where $\hat{f}(s^*) \in \mathbb{R}_{\geq 0}$ is a parameter estimating the cost of an optimal solution, and $k_{\text{vio}}(x)$ denotes the number of constraints $\omega \in \Omega$ violated by x. Eventually, I obtain the following definition of f:

$$f(x) = c_{\text{exp}}\left(G_{\text{now}}, G(x)\right) + P(x)$$

4.5.3 Search Strategy

The final step in implementing AntPower is to implement the ants' strategy of searching for optimal solutions. A search strategy consists of a neighborhood function, a transition rule, a pheromone update mechanism, and heuristic values.

Neighborhood

Intuitively, an ant's neighborhood comprises all states that the ant can reach with one step. Formally, neighborhood function N assigns to each state $x \in X$ all components $c \in C$ for which $\langle x, c\rangle \in \tilde{X}$. Consequently, to define N, I only need to define $\tilde{X}$, which I have already done in section § 4.5.2. To recall, $\tilde{X}$ is the set of states that represent topologically valid grids. Consequently, $c \in N(x)$ if and only if $\langle x, c\rangle$ represents a topologically valid grid.

Transition Rule

When an ant is in state $x \in \tilde{X}$, it moves to some state $\langle x, c\rangle \in \tilde{X}$ according to a *transition rule*. A transition rule selects a component $c \in N(x)$ as a function of pheromone values $\tau_1, \dots, \tau_n$ and heuristic values $\eta_1, \dots, \eta_n$. AntPower adopts the transition rule of ACS, which probabilistically selects $c \in N(x)$ such that:

$$c = \begin{cases} \text{argmax}_{c_i \in N(x)} \left(\tau_i {\eta_i}^{\beta}\right) & \text{if } q \leq q_0 \\ c' & \text{if } q > q_0 \end{cases}$$

where $q \in [0, 1]$ is a uniformly distributed random variable, $q_0 \in [0, 1]$ and $\beta \in \mathbb{R}_{\geq 0}$ are parameters, and $c' \in C$ is a random variable with the following probability distribution p:

$$p(c_i) = \begin{cases} \frac{\tau_i {\eta_i}^{\beta}}{\Sigma_{c_j \in N(x)}\left(\tau_j {\eta_j}^{\beta}\right)} & \text{if } c_i \in N(x) \\ 0 & \text{if } c_i \notin N(x) \end{cases}$$

Parameter q_0 modulates the degree to which ants exploit their collective knowledge stored as pheromones [40]. If $q \leq q_0$, an ant selects from its neighborhood that component which maximizes $\tau_i {\eta_i}^{\beta}$ (e.g., for $\beta = 0$, the ant selects the component with the highest pheromone value). In contrast, if $q > q_0$, the ant probabilistically selects $c_i \in N(x)$, with the probability of selecting c_i being proportional to $\tau_i {\eta_i}^{\beta}$. Thus for high values of q_0, ants tend to select components that were part of high-quality solutions in earlier iterations, whereas for low values of q_0, ants are more likely to explore components that have been part of inferior solutions, or that have not been selected by any ant so far.

Furthermore, parameter β controls the relative weight of pheromone values as compared to heuristic values. The purpose of both pheromone values and heuristic values is to guide ants in the process of constructing solutions. The difference is that pheromone values are dynamic values encoding the ants' empirical knowledge, whereas heuristic values are a priori estimations of the solution components' quality. I will define the heuristic values at the end of this section.

Pheromone Update

ACS specifies two rules for updating pheromones, *local pheromone update* and *global pheromone update*. Local pheromone update occurs each time an ant visits a component $c_i \in C$, and affects only the pheromone value of the visited component:

$$\tau_i \leftarrow (1-\xi)\tau_i + \xi\tau_0$$

where $\xi \in (0,1)$ is a parameter. The initial pheromone value $\tau_0 \in \mathbb{R}_{\geq 0}$ represents a base level of pheromone, and τ_i converges to this base level exponentially with the number of local pheromone updates. If $\tau_i > \tau_0$, local pheromone update decreases τ_i to make c_i less attractive; conversely, if $\tau_i < \tau_0$, local pheromone update increases τ_i to make c_i more attractive.

The second pheromone update rule, *global pheromone update*, is applied at the end of each iteration, and affects only the pheromone values τ_i of components that are part of the best-so-far solution $s \in S$. If c_i is a component of s, its pheromone value is updated as follows:

$$\tau_i \leftarrow (1-\rho)\tau_i + \rho\frac{\hat{f}(s^*)}{f(s)}$$

where $\rho \in (0,1)$ and $\hat{f}(s^*)$ are parameters. Just as the local pheromone update rule, the global pheromone update rule calculates a weighted average of the previous pheromone value and a target value. Instead of τ_0, the target value is now the reciprocal of $f(s)$, normalized by the estimated cost of an optimal solution $s^* \in S^*$. With that, the change in τ_i depends on the quality of s. More precisely, if the reciprocal of the normalized cost of s exceeds τ_i, then τ_i increases, otherwise τ_i decreases.

Heuristic Values

Ants, when deciding which component to visit next, not only consider pheromone values but also heuristic values. Heuristic values are static values that estimate the quality of solution components independent of other components. Without heuristic values, ants would initially have no information about the structure of the problem to solve, and would therefore construct solutions $s \in S$ randomly. In real-world problems, the size of S far exceeds the number of solutions ants can construct; therefore, it is essential to guide ants toward high-quality solutions from the beginning.

Following is the definition of the heuristic values used by AntPower. Denoting the cost of $c \in C$ by $g(c)$ (as specified in subsection § 3.2), the heuristic value $\eta_i \in \mathbb{R}_{\geq 0}$ of component $c_i \in C$ is given by:

$$\eta_i = \frac{1}{g(c_i) - g_{\min} + 1}$$

where $g_{\min}$ is the minimum of $g(c)$ across all $c \in C$. In other words, η_i is the reciprocal of $g(c_i)$, normalized such that the smallest heuristic value equals one. With this definition of η, ants prefer low-cost grid modifications over expensive grid modifications—unless pheromone values point toward expensive grid modifications that have proven valuable in previous iterations.

5 Evaluation

I now evaluate AntPower's performance. For that, I first present a real-world use case (§ 5.1). Then, I develop two simple solution methods, which will serve as baselines for the benchmarking of AntPower (§ 5.2). Next, I present the solution found by AntPower, and compare it to the baseline solutions (§ 5.3). Finally, I study the effect of control parameters on AntPower's performance (§ 5.4).

5.1 Use Case

To begin with, I present a real-world use case of LV expansion planning. This use case will serve as the basis for the benchmarking of AntPower (§ 5.3).

5.1.1 Planning Conditions

The presented use case is the case of expanding the LV grid shown in Figure 5.1. The grid powers a village of 800 residents, and is located in a rural area of Central Germany. Five transformers connect the grid to an MV grid. The grid contains 837 buses and 735 line segments. The lines' total length is 20 km, and—as is common for LV grids in Germany—all lines are underground lines. Furthermore, the grid contains 99 sources (mostly photovoltaic appliances), 357 loads (mostly residential buildings), and 113 switches (most of which are closed).

The grid's operator, having connected many photovoltaic appliances to the grid in recent years, must now extensively expand the grid before the grid can cope with additional strain. As this situation forces the operator to reject requests for the connection of further generators, it is important to expand the grid as soon as possible. Therefore, I assume a narrow planning horizon of three years.

Following are the options that I consider for expanding the grid:

1. The installation of any of five line segments that run along new routes[1]
2. The replacement of any of 369 line segments by high-capacity line segments[2]
3. The dismantling of any of 369 line segments
4. The opening of any of the 102 closed switches
5. The closing of any of the 11 open switches

[1] The reason why I do not consider more options for laying lines along new routes is that the grid operator prefers to lay lines along streets, and, in the given village, lines already run along most of the streets.

[2] I consider "replacement lines" to consist of two parallel cables of type "NAYY 4 x 240 SE".

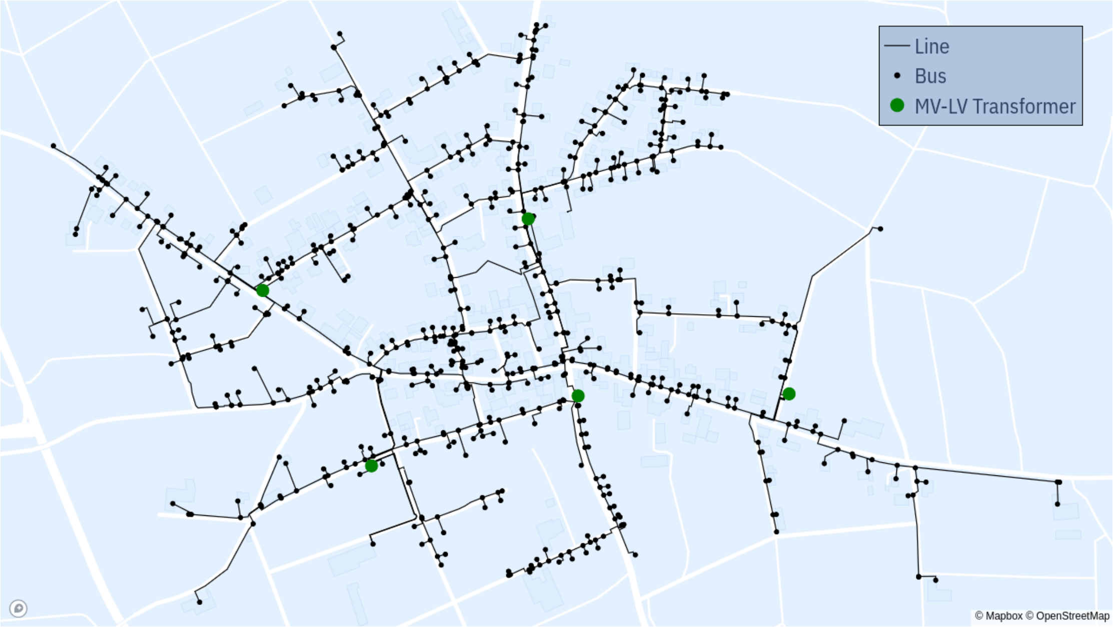


Figure 5.1: Status quo of the grid that is to be expanded. For better clarity, the figure does not show sources, loads, or switches.

Thereby, to reduce complexity, I allow the replacement or dismantling of line segments only for the 369 line segments that are not *private*. I consider a line segment to be private if it attaches to a bus to which also a source or load attaches. In the given LV grid, most private line segments are short segments that connect residential buildings to the grid. Excluding private line segments from the set of allowed modification options greatly reduces the complexity of the planning problem, considering that almost half of all installed line segments are private segments in the given use case. Section § 5.3 will show that the remaining grid modification options suffice to efficiently expand the grid.

After expansion, I demand that the grid is able to cope with the maximum strain that grid usage can place on the grid. Maximum strain emerges from the following worst-case scenarios of grid usage. In the first scenario, called *feed case*, all generators operate at peak power, while none of the loads consumes any electricity. This scenario comes close to the grid usage that is observed during the summer holidays, when photovoltaic generation peaks and electricity consumption stagnates [46]. In the second scenario, called *load case*, none of the sources feeds in any electricity, while all loads operate at peak power. This scenario resembles grid usage on winter days at which photovoltaic generation is low and electricity consumption is high [46]. When performing PFA, I require that none of the line segments gets overloaded, both in the feed case and the load case. Additionally, bus voltages must not deviate from the grid's nominal voltage (0.4 kV) by more than 6% in the feed case and 4% in the load case. Figure 5.2 shows that the grid, given its current structure, violates these constraints at many locations. More precisely, one line segment (located in the village's center) is overloaded, and at 83 buses, voltage exceeds the allowed range in at least one of the two worst-case scenarios.

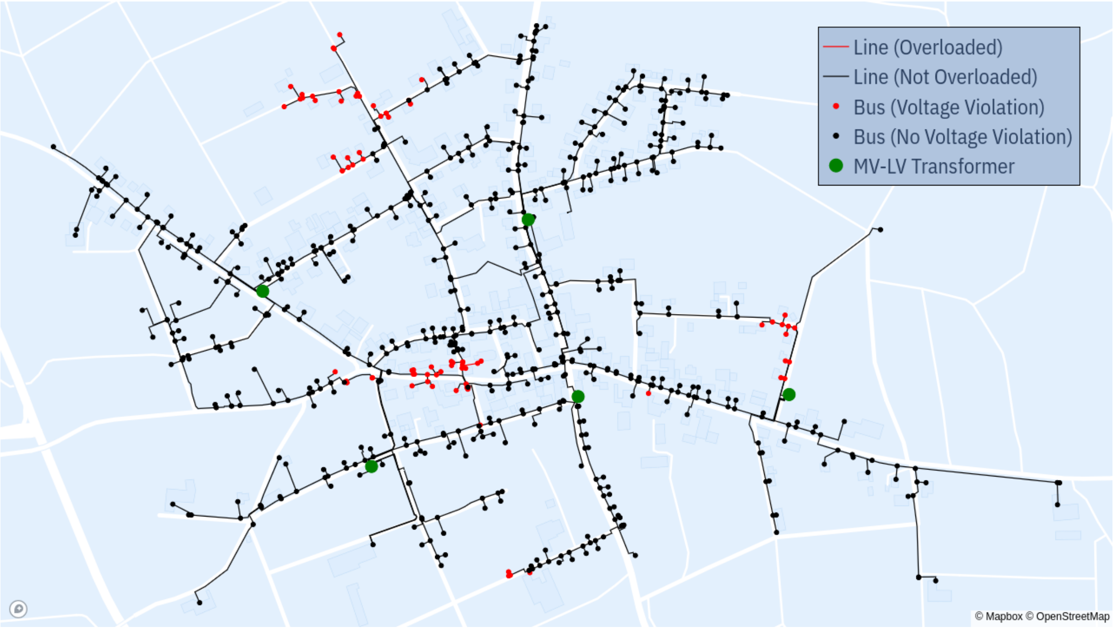


Figure 5.2: The locations at which the grid that is to be expanded will violate voltage or line loading constraints at the end of the planning horizon, given the grid's current structure and worst-case scenarios of grid usage.

To recall from section § 3.4.2, the expanded grid must have radial topology, except that it may contain rings, threads, and stitches. As of now, the grid does not satisfy this constraint, because the grid currently contains several rings and threads in which all switches are closed, although each ring and thread must contain an open switch, as we have seen in section § 2.1.3.

The last planning condition that is left to be specified is the cost of grid expansion measures. As I lack information on the grid operator's expenses, I roughly estimate costs of grid expansion measures as follows. First, I suppose that each switching action incurs an operational cost of 1K €. Second, I assume that installing or replacing one kilometer of line costs 0.1M €. Third, I approximate the acquisition cost of cables by 20K € per kilometer.

In addition to evaluating AntPower using the presented use case, it is essential to consider a wide spectrum of other use cases in future research. The reason is that the grid planning problems AntPower supports vary in many aspects: the structure of the grid that is to be expanded; the scenarios of expected grid usage; the options considered for grid expansion; the constraints imposed on the expanded grid; and the costs of the considered expansion options. An evaluation that covers this wide range of planning conditions requires a diverse set of test grids and planning scenarios.[3]

[3] Generally, a standardized set of publicly available problem instances would facilitate the evaluation of grid planning tools. Such benchmark problem instances could be based on, for example, the recently published and publicly available *SimBench* grids [47].

5.1.2 Input Data

The planning conditions specified in the previous section translate to the following planning inputs:

1. AntPower obviously requires a specification of the grid that is to be expanded; therefore, the planning inputs must specify both the technical properties of the grid's components and the grid's structure $G_{\mathrm{now}} = (B, L_{\mathrm{now}} \cup W_{\mathrm{now}})$. In the given use case, $|B| = 837$ (as the grid contains 837 buses), $|L_{\mathrm{now}}| = 735$ (as the grid contains 735 line segments), and $|W_{\mathrm{now}}| = 102$ (as the grid contains 102 closed switches).

2. AntPower expects to be given the set of expansion options $E_{\mathrm{add}} = L_{\mathrm{add}} \cup W_{\mathrm{add}}$. In the given use case, $|L_{\mathrm{add}}| = 374$ (five lines that run along new routes plus 369 ”replacement lines”) and $|W_{\mathrm{add}}| = 11$ (as the grid contains 11 open switches).

3. The financial inputs are $c_{\mathrm{swi}} = 1$K €, $c_{\mathrm{ins}}(z) = 100$ €/m, and $c_{\mathrm{cab}}(z) = 20$ €/m, where $z \in Z$ is the type of lines that consist of two parallel ”NAYY 4 x 240 SE” cables. I suppose that all newly laid lines are of type z; thus I do not need to specify installation costs and cable costs for any other line types.

4. Further planning inputs are the load case $p(t_1)$ and the feed case $p(t_2)$, where $t_1 \approx t_2 \approx \theta = 3$ a (both the feed case and the load case describe grid usage at the end of the planning horizon, which is three years).

5. The use case's voltage and line loading limits imply $\lambda = 1$ (the maximum power that is allowed to flow through a line equals 100% of the line's transmission capacity), $[v_{\mathrm{min}}, v_{\mathrm{max}}] = [376\,\mathrm{V}, 424\,\mathrm{V}]$ in the feed case (bus voltages may deviate from nominal voltage by at most 6% in the feed case), and $[v_{\mathrm{min}}, v_{\mathrm{max}}] = [384\,\mathrm{V}, 416\,\mathrm{V}]$ in the load case (bus voltages may deviate from nominal voltage by at most 4% in the load case).

5.2 Baselines

In this section, I present two simple methods that solve the problem instance described in the previous section. These methods will serve as baselines when evaluating the solutions found by AntPower.

5.2.1 Manual Method

The first of the two baselines aims to reflect how an expert would solve the given grid expansion problem. To date, most utilities use expert judgment and empirical rules—rather than automated tools—to plan distribution grids [12]. Therefore, supported by an electrical engineer with many years of research experience in grid planning, I developed a manual planning method, which involves no computer aid except a PFA tool and a geographic map of the grid that is to be expanded.

This manual planning method is a heuristic procedure of six steps. As a first step, I reinforce overloaded line segments, that is, I replace each overloaded line segment by a line segment of higher transmission capacity. Second, I open all switches to fix the issue that G_{now} contains rings and threads in which all switches are closed. Third, by closing selected switches, I connect sources and loads to their closest upstream transformers. I preferably close those switches that are also closed in G_{now} to keep down costs. Fourth, I check for each new line segment (i.e., for each line segment that—if built—runs along a route where no line is located so far) if integrating this line segment into the grid significantly reduces the number of voltage limit violations. Integrating a line segment thereby means installing the line segment and then fixing topology by flipping switches and dismantling line segments as needed. Fifth, I shorten overly long feeders if this shortening helps to further reduce the number of voltage band violations. To shorten a feeder, I disconnect several line segments that are located at the end of the feeder, and reconnect these segments to a nearby upstream transformer. Sixth, I fix the remaining voltage limit violations by reinforcing lines as follows. Starting from each upstream transformer, I follow the path to each bus at which voltage exceeds the allowed limits. For each of these paths, I reinforce the line segments along the path, until the voltage limit violation at the end of the path disappears.

5.2.2 One-Opt Local Search

To complement the previously presented human baseline with an automated, algorithmic baseline, I implement a simple variant of *local search*. A local search algorithm is an algorithm that, starting at some initial solution, iteratively moves to a solution in the *neighborhood* of the current solution. The definition of which solutions belong to a solution's neighborhood varies across variants of local search algorithms. In *one-opt local search*, the variant that I implement, the neighborhood of solution $s \in S$ comprises those solutions that result from replacing one component of s by some other component.

In the following, I present an algorithm that results from a straight-forward application of one-opt local search to the grid expansion problem. First, to obtain an initial solution, the algorithm randomly creates a forest $G = (B, E)$, whose roots represent the buses to which the upstream transformers attach and whose edges $E \subseteq L_{\mathrm{now}} \cup L_{\mathrm{add}}$ connect all buses $b \in B$ to the grid. To recall, forests of this type represent topologically valid solutions to the grid expansion problem. Having found an initial solution, the algorithm now iteratively moves from the current solution $s \in S$ to a solution $s' \in S$ in the neighborhood of s; the subsequent paragraph will explain in detail how this transition works. After each transition, the algorithm assigns $s \leftarrow s'$ but only if $f(s') < f(s)$. The overall process terminates after $k_{\mathrm{itr}} \cdot k_{\mathrm{ant}}$ transitions. With that, a local search process generates just as many solutions as a colony of AntPower. Furthermore, just as AntPower runs k_{col} colonies in parallel, the presented local search algorithm runs k_{col} local search processes in parallel. With that, AntPower and the local search algorithm generate the same number of solutions, and use the same number of CPU cores. If $k_{\mathrm{col}} > 1$, an additional advantage of executing k_{col} local search processes (rather than only one) is that, due to the k_{col} random initializations, the quality of the final solution depends less on the quality of the initial solution.

I now focus on the transition from one solution to another. To recall, in one-opt local search, transitioning from one solution to another amounts to replacing one solution component by another. Adopting AntPower's definition of solution components (§ 4.5.2), I define the components of solution $s \in S$ to be the edges $e \in E$ of the graph $G = (B, E)$ that represents s. With that, moving from $s \in S$ to $s' \in S$ amounts to replacing some edge $e \in E$ by some edge $e' \in (L_{\text{now}} \cup L_{\text{add}}) \setminus E$. To implement this replacement, first I randomly select an edge $e' \in (L_{\text{now}} \cup L_{\text{add}}) \setminus E$. Then, I remove an edge e from E such that $E' = (E \cup \{e'\}) \setminus \{e\}$ yields a topologically valid grid $G' = (B, E')$. To determine which edge to remove, I must distinguish three cases. The first case is that the previously added edge e' represents a reinforcement line; that is, e' connects the same buses as some (already existing) line $l \in L \subset E$. In this case, I remove l, so that e' replaces l. In the second case, $E \cup \{e'\}$ contains a cycle; that is, after adding e', the grid contains a ring in which all switches are closed. To fix this, I remove one of the two edges that are adjacent to e' in the cycle. In the third case, $E \cup \{e'\}$ contains a path from one of the forest's roots to another; that is, after adding e', the grid contains a thread in which all switches are closed. Similarly to the previous case, I remove one of the two edges that are adjacent to e' in the path that connects the two roots. Consequently, in each of the three cases, replacing e by e' amounts to transitioning from a topologically valid grid G to a topologically valid grid G'.

5.3 Results

This section presents the results of solving the problem instance introduced in section § 5.1 using the manual method, the local search method, and AntPower. I use the following set of control parameters, which gave best results throughout my experiments: $k_{\text{col}} = 25$; $k_{\text{ant}} = 10$; $k_{\text{itr}} = 2\text{K}$; $\beta = 1$; $q_0 = 0.9$; $\xi = 0.1$; $\rho = 0.1$; $\tau_0 = 0.008$; and $\hat{f}(s^*) = 0.1\text{M}$ €. This section is structured as follows. First, subsection § 5.3.1 gives an overview by comparing the results across the three solution methods. Then, the subsequent subsections examine in detail the results obtained using the manual method (§ 5.3.2), the local search method (§ 5.3.3), and AntPower (§ 5.3.4).

5.3.1 Overview

For a result overview, Table 5.1 compares the best solutions found by the two baseline methods with the best solution found by AntPower. Notably, all of the three methods succeed in finding solutions that observe all constraints. Yet, the solution returned by AntPower saves 60% of costs as compared to the manually created solution, and saves 64% as compared to the solution created by the local search method. With that, AntPower clearly outperforms the two baseline methods.

Table 5.1: Comparison of the best solutions found by the manual method, one-opt local search, and AntPower

Feature of best found solution	Manual method	Local search	AntPower
Topology of the expanded grid	Valid topology	Valid topology	Valid topology
Number of overloaded lines	0	0	0
Number of voltage limit violations	0	0	0
Cost of grid expansion	210K €	231K €	84K €

5.3.2 Manual Method

Figure 5.3 shows the manually created solution $s \in \tilde{S}$, with $c_{\mathrm{exp}}(G_{\mathrm{now}}, G(s)) = 210\mathrm{K}$ €. The expansion plan that s represents involves: (1) the installation of four of the five line segments that run along routes at which no lines are located so far (shown in purple); (2) the reinforcement of 44 line segments, whose total length is 1.0 km (shown in blue); (3) the dismantling of three very short line segments (shown in red but hardly visible due to their short lengths); (4) the opening of eleven switches (shown in orange); and (5) the closing of one switch (also shown in orange).

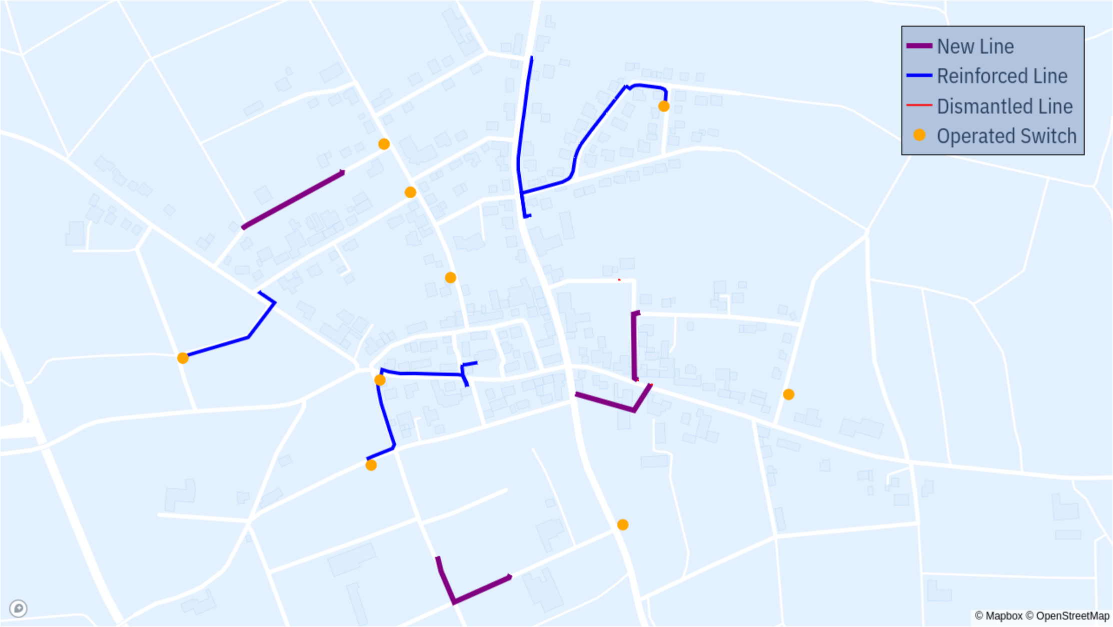


Figure 5.3: The expansion plan obtained using the manual planning method. Although the expansion plan involves the dismantling of three line segments, these line segments are hardly visible in the figure due to their short lengths (the longest dismantled line segment is only 17 meters long). Furthermore, each orange dot can represent several switch operations.

The reason why the three dismantled line segments are very short (with the longest of the three segments being only 17 meters long) is the following. In the context of the manual planning method, the sole purpose of dismantling line segments is to fix violations of the grid's topology. To keep down costs, it is best to fix such violations by dismantling line segments that are as short as possible. In practice, the grid operator may want to install normally open switches rather than dismantling short line segments, because the installation of switches increases flexibility in controlling the grid. Therefore, in future research, regarding the installation of switches as an additional degree of freedom in grid expansion planning is likely worth the resulting increase in problem complexity.

5.3.3 One-Opt Local Search

Figure 5.4 shows the best solution $s \in \tilde{S}$ obtained using the local search method. Strikingly, the local search method selects the same new line segments as the manual method. Additionally, the total length of reinforced line segments is roughly equal across the two methods (manual method: 1.0 km; local search method: 1.1 km). However, whereas the manual method reinforces 44 line segments, which are located throughout the village, the local search method reinforces only 19 line segments, which are located mainly in the village's center. Furthermore, the local search method dismantles 14 line segments (eleven more than the manual method), opens eight switches (three less than the manual method), and closes ten switches (nine more than the manual method). The costs of these expansion measures sum up to 231K €, which is 1.1 times the cost of the manually created expansion plan.

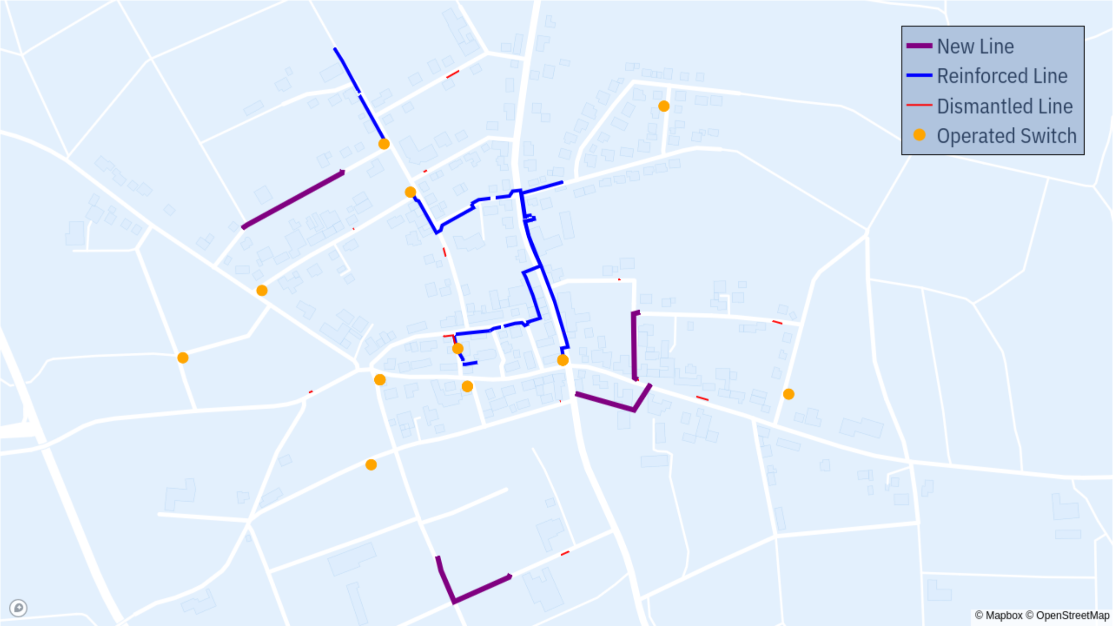


Figure 5.4: The best expansion plan obtained using the local search method. Each orange dot can represent several switch operations.

Next, I examine the convergence of costs across iterations. Figure 5.5 shows the cost $f(s_i)$ of the best found solution s_i after each iteration $i \in \{1, 2, \ldots, k_{\text{itr}} \cdot k_{\text{ant}}\}$ for the first three of the 25 local search processes. Initially, the costs rapidly decrease. However, after 2.5K iterations at the latest, the costs stagnate. This stagnation means that the local search processes get stuck in local optima of f, which they cannot leave due to the greedy nature of one-opt local search. This problem occurs across all processes—even when doubling the number of processes from 25 to 50, in which case the cost of the overall best solution decreases by merely 9%. Furthermore, the costs of the generated solutions vary widely, with a standard deviation as high as 2.6M € (based on a sample of 50 local search processes). To conclude, the grid expansion problem appears to be too complex to be efficiently and reliably solvable using a method as simple as one-opt local search. This observation motivates the use of a more sophisticated method, such as the ACO method that AntPower implements.

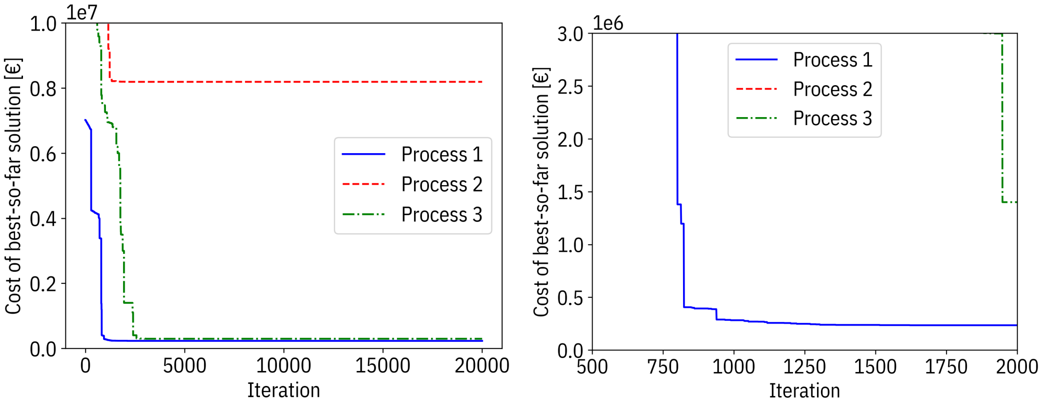


Figure 5.5: Convergence behavior of the local search method. The right plot shows an excerpt of the left plot. The figure shows only the first three of the 25 local search processes.

5.3.4 AntPower

Eventually, Figure 5.4 shows the best solution $s \in \tilde{S}$ found by AntPower. Whereas the two baseline methods select four new line segments, AntPower selects only two. Furthermore, AntPower reinforces only 19 line segments, with a total length of 0.67 km, which is 33% less than the total length of reinforced line segments obtained using the manual method. However, AntPower dismantles more than thrice as many line segments as the manual method (ten vs. three line segments), and closes thrice as many switches (three switches vs. one switch). Finally, the number of opened switches (two) is far below that of the manual method (eleven). With a cost of $c_{\text{exp}}(G_{\text{now}}, G(s)) =$ 84K €, AntPower's expansion plan is 60% cheaper than the manually created expansion plan, and 64% cheaper than the expansion plan obtained using the local search method.

In analogy to Figure 5.5, Figure 5.7 shows the cost of the best-so-far solution after each iteration for the first three of the 25 colonies. Initially, the costs decrease rapidly for all colonies, and then steadily converge to values between 84K € and 116K €. The costs' standard deviation, estimated using a sample of 50 colonies, equals 16K €, which is far below the standard deviation estimated for the local search method (2.6M €). Thus the quality of generated solutions is relatively stable. Still, it may be surprising to see that the second and the third colony fail to catch up to the first colony, although the first colony's solution hardly improves throughout the end of the search. The reason is that, as the search goes on, pheromones increasingly bias the ants' behavior to the point that the ants hardly deviate from the best solutions found so far. This pheromone bias is a requirement for ACS-based algorithms (like AntPower) to perform well; ideally, ants initially explore a variety of different solutions, but increasingly focus on the neighborhood of the best solutions found so far [40].

To ensure that the given pheromone bias makes a reasonable trade-off between the exploration of novel solutions and the exploitation of already found solutions, I evaluate how the degree to which ants show exploratory behavior develops over time. I measure this *degree of exploration* by the number of solution components that are visited by at least one ant in each iteration, rela-

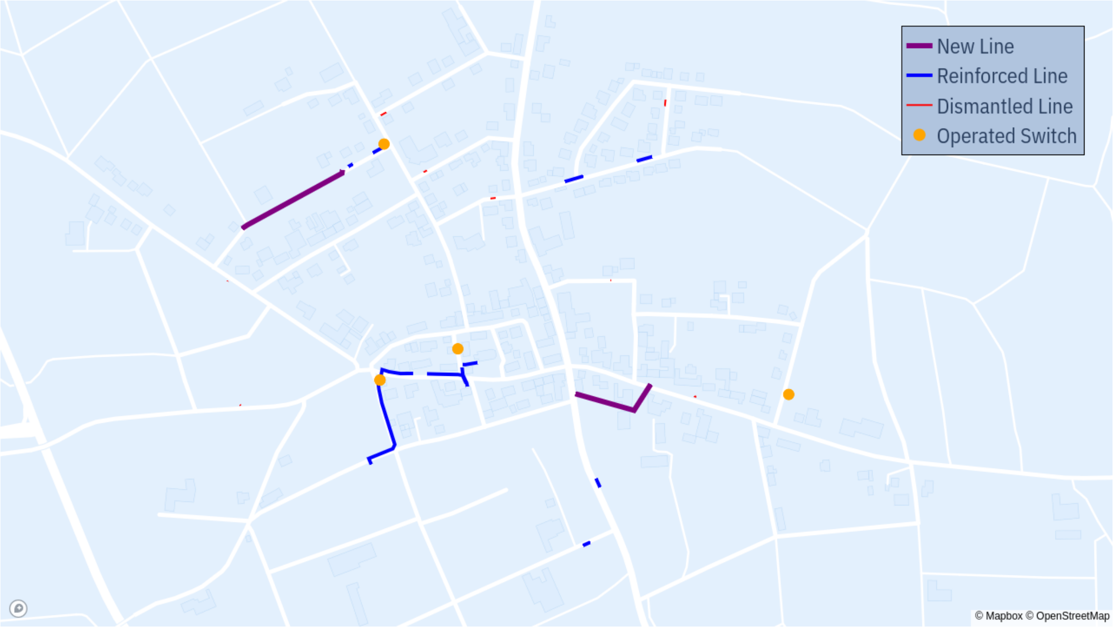


Figure 5.6: The best expansion plan found by AntPower. Each orange dot can represent several switch operations.

tive to the total number of solution components (856). To capture temporal dynamics, I measure the degree of exploration in each iteration, and, to smooth out local variations, I compute the moving average using a window size of 50 iterations. Figure 5.8 shows the resulting plot. In the first 1K iterations, the average degree of exploration falls from 64% to 57%, where it remains throughout the second half of the search. With that, the ants show a high degree of exploration initially, but increasingly search in the neighborhood of previously found solutions, as desired.

In terms of runtime, AntPower benefits from the parallel implementation of colonies. As each colony runs in a separate process, changing the number of colonies hardly affects AntPower's runtime, provided that the number of available CPU cores exceeds the number of colonies. Running 25 colonies in parallel on a computer with 32 CPU cores and a clock rate of 2.5 GHz takes about two days—an acceptable duration, considering that grid planning is not a time critical task. Notably, PFA consumes 24% of all runtime, even though the given use case (§ 5.1) comprises only two snapshots (the feed case and the load case). With ten snapshots, the share of time consumed by PFA rises to 46%, and overall runtime increases by 40%. With an even larger number of snapshots, that is, with time series comprising tens or hundreds of snapshots, the use of a less accurate PFA algorithm is necessary to achieve reasonable runtime performance. However, rather than downgrading to a less accurate PFA algorithm, I recommend to not pass a time series to the PFA algorithm in the first place; instead, I recommend to aggregate the time series into few representative snapshots using a clustering technique—two snapshots generally suffice to accurately represent a time series in the context of distribution grid planning [48].

Finally and not surprisingly, AntPower—not being a data-intense application—has low space requirements. It uses 12 GB of memory, and generates 2 GB of outputs, most of which are eval-

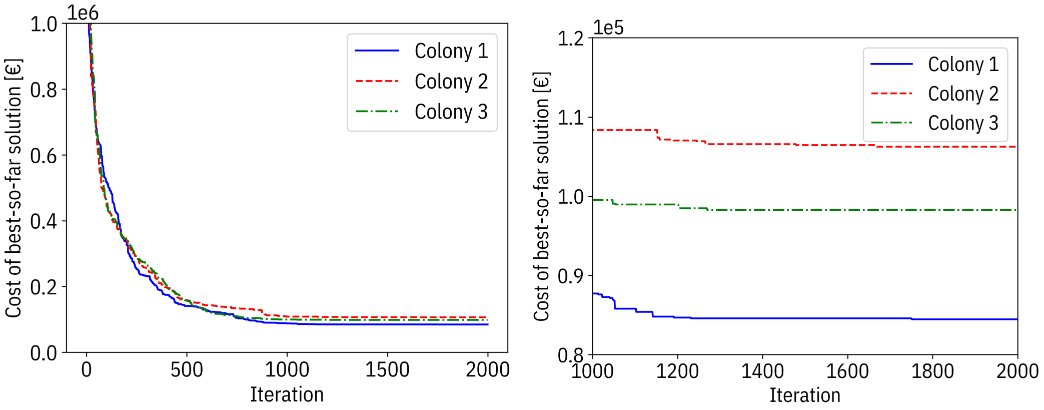


Figure 5.7: Convergence behavior of AntPower. The right plot shows an excerpt of the left plot. For better clarity, the figure shows only the first three of the 25 colonies.

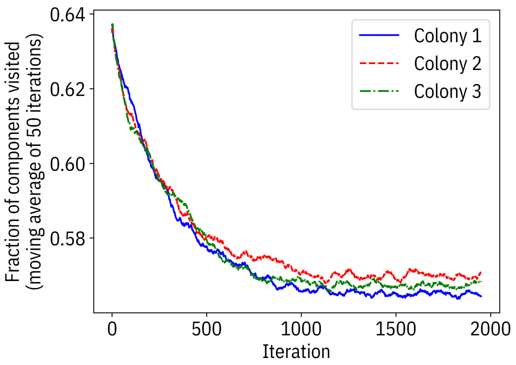


Figure 5.8: Degree to which the ants show exploratory behavior across iterations. For better clarity, the figure shows only the first three of the 25 colonies.

uation data. Given the abundance of inexpensive memory and storage that is available today, AntPower's space requirements are insignificant.

5.4 Effect of Control Parameters

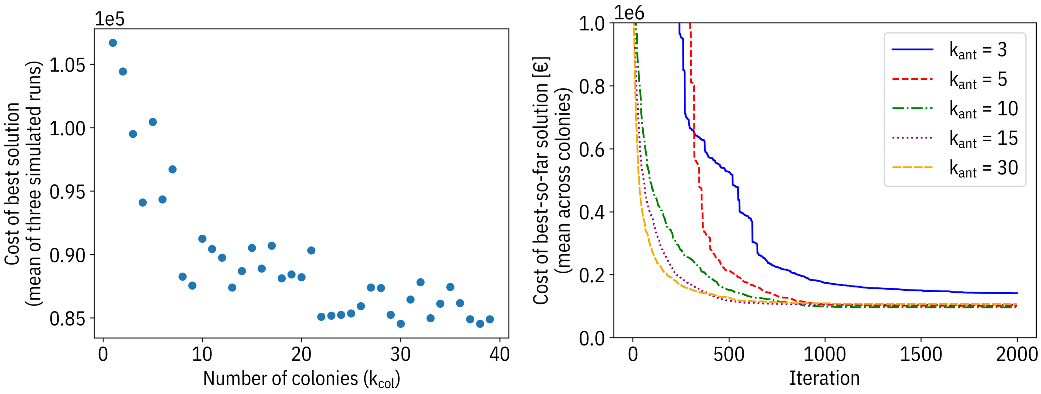
As a final evaluation step, I examine how AntPower reacts to changes in control parameter settings. More precisely, I study for each control parameter how varying its value affects AntPower's performance, while keeping all other parameters at their base values. As base values, I use the values given in section § 5.3, except that I reduce the number of colonies from 25 to three due to limited computational resources. Furthermore, due to the large number of control parameters, I limit the analysis to the effects of individual parameters, and leave an evaluation of interaction effects for future research.

This section has the following structure. Subsection § 5.4.1 studies the parameters that control the number of constructed solutions (k_{col}, k_{ant}, and k_{itr}). Subsection § 5.4.2 studies the parameters of the transition rule (q_0, β, and τ_0); and subsection § 5.4.3 studies the parameters that control the update of pheromones (ξ, ρ, and $\hat{f}(s^*)$).

5.4.1 Parameters Controlling the Number of Constructed Solutions

Three parameters control the total number of solutions that ants construct during a run of AntPower: the number of colonies (k_{col}), the number of ants per colony (k_{ant}), and the number of iterations (k_{itr}). In the following, I focus on each of the three parameters individually.

I begin by considering the number of colonies (k_{col}). With an increasing number of colonies, the cost of the best found solution decreases, until it remains constant at some point. To limit computational effort (which rises proportionally to k_{col}), it is reasonable to choose a value for k_{col} beyond which the cost of the best found solution hardly decreases any further. To find such a value, I examine the effect of varying k_{col} as follows. For each $i \in \{1, 2, \dots, 30\}$, I simulate three runs of AntPower, each of which uses the setting $k_{\mathrm{col}} = i$. I simulate a run of AntPower by drawing a random subsample of size k_{col} from a sample of 180 colonies. For each k_{col}, I then calculate the mean cost across the three simulated runs. Figure 5.9a shows the results. As k_{col} increases, the mean cost decreases, but stagnates at around $k_{\mathrm{col}} = 25$, indicating that $k_{\mathrm{col}} = 25$ is a reasonable setting.



(a) Effect of k_{col} on the cost of the best solution (b) Effect of k_{ant} on the convergence of costs

Figure 5.9: Effect of the number and size of colonies on AntPower's performance

Next, I examine the effect of varying the number of ants per colony (k_{ant}). As Figure 5.9b shows, k_{ant} hardly affects the cost of the best found solution, but rather influences the speed at which the cost function converges to its final value; the larger k_{ant}, the higher the speed of convergence. For $3 < k_{\mathrm{ant}} \leq 30$, costs converge steadily and reasonably fast, so using any value in this range is a reasonable choice, although $k_{\mathrm{ant}} = 10$ yields slightly better results than the other settings. That $k_{\mathrm{ant}} = 10$ gives good results confirms the ACO authors' finding that using ten ants per colony is a good default value [40].

Finally, parameter k_{itr} sets the number of iterations after which AntPower terminates the search for solutions. In a future version of AntPower, this parameter will be obsolete, because AntPower will be able to automatically decide when to terminate its search (e.g., when costs have not decreased within the last, say, 50 iterations). Yet, the current need to specify k_{itr} does not significantly impair AntPower's usability, because—as the analyses throughout this section indicate—the base setting ($k_{\mathrm{itr}} = 2\mathrm{K}$) leads to satisfactory results independent of other settings.

5.4.2 Parameters Controlling the Transition Rule

I now turn toward the parameters of the transition rule, that is, the parameters that control how ants move from one solution component to the next.

I begin by studying parameter q_0, which modulates the degree to which the search for solutions concentrates around the best solutions found so far. In doing so, q_0 controls the trade-off between the exploitation of knowledge about the best solutions found so far and the acquisition of additional knowledge through further exploration. Accordingly, Figure 5.10a shows a strong correlation between q_0 and the degree to which ants show exploratory behavior. The exploration-exploitation trade-off largely affects the convergence of costs, as Figure 5.10b shows. If q_0 is too low ($q_0 < 0.85$), an overly strong focus on exploration hinders convergence toward low-cost solutions. If q_0 is too high ($q_0 > 0.95$), the ants run into sub-optimal solutions, too, because they concentrate too much on the best solutions found so far. Using $q_0 = 0.9$ (as recommended by the authors of ACS [40]) yields best results.

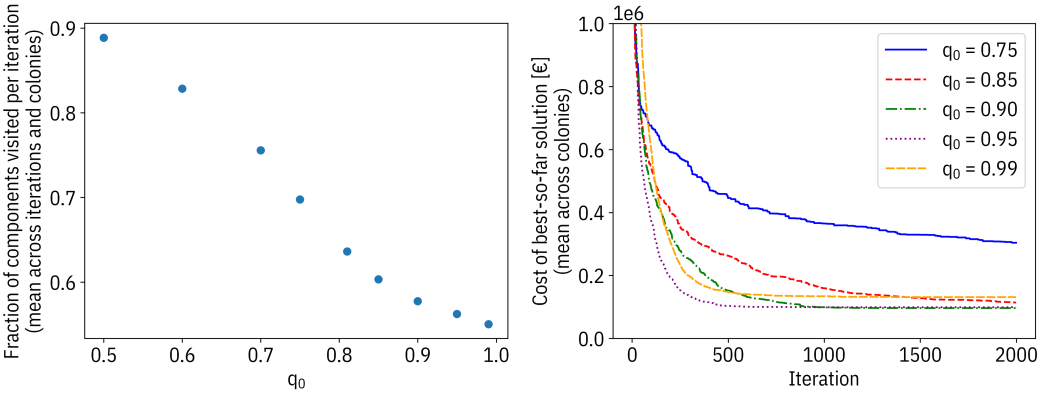


Figure 5.10: Effect of parameter q_0 on: (a) the degree of exploration; and (b) the convergence of costs

The purpose of parameters β and τ_0 is to adjust how much weight the ants give to pheromones relative to heuristic values when deciding which solution component to visit next. Figure 5.11 indicates that AntPower is robust against changes in the values of β and τ_0, considering that a large range of settings ($\beta = 1.9 \pm 60\%$, $\tau_0 = 7.5 \times 10^{-3} \pm 50\%$) yields almost equal results. Performance deteriorates only for $\beta < 0.8$, $\beta > 4$, $\tau_0 < 4.0 \times 10^{-3}$, and $\tau_0 > 1.1 \times 10^{-2}$. The best-performing setting is:

$$\beta = 1; \;\; \tau_0 = \frac{\hat{f}(s^*)}{\hat{f}(s^\circ)} = \frac{0.1\text{M}\,€}{12.5\text{M}\,€} = 8 \times 10^{-3}$$

where $\hat{f}(s^*)$ estimates the cost of the overall best solution, and $\hat{f}(s^\circ)$ the mean cost (including penalty) of a randomly created solution. The formula $\tau_0 = \hat{f}(s^*) \,/\, \hat{f}(s^\circ)$ is an adaptation of a formula that the authors of ACS propose for the solution of the TSP [40].

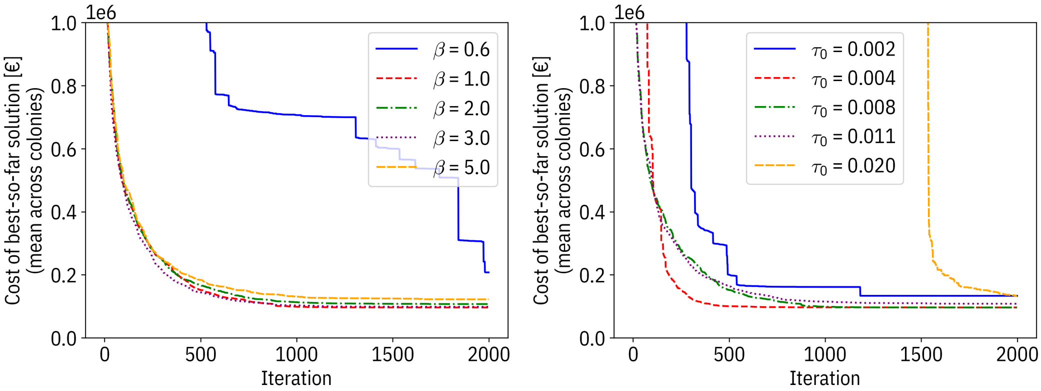


Figure 5.11: Effect of parameters that control the weight of pheromones relative to heuristic values on the convergence of costs: (a) parameter β; (b) parameter τ_0

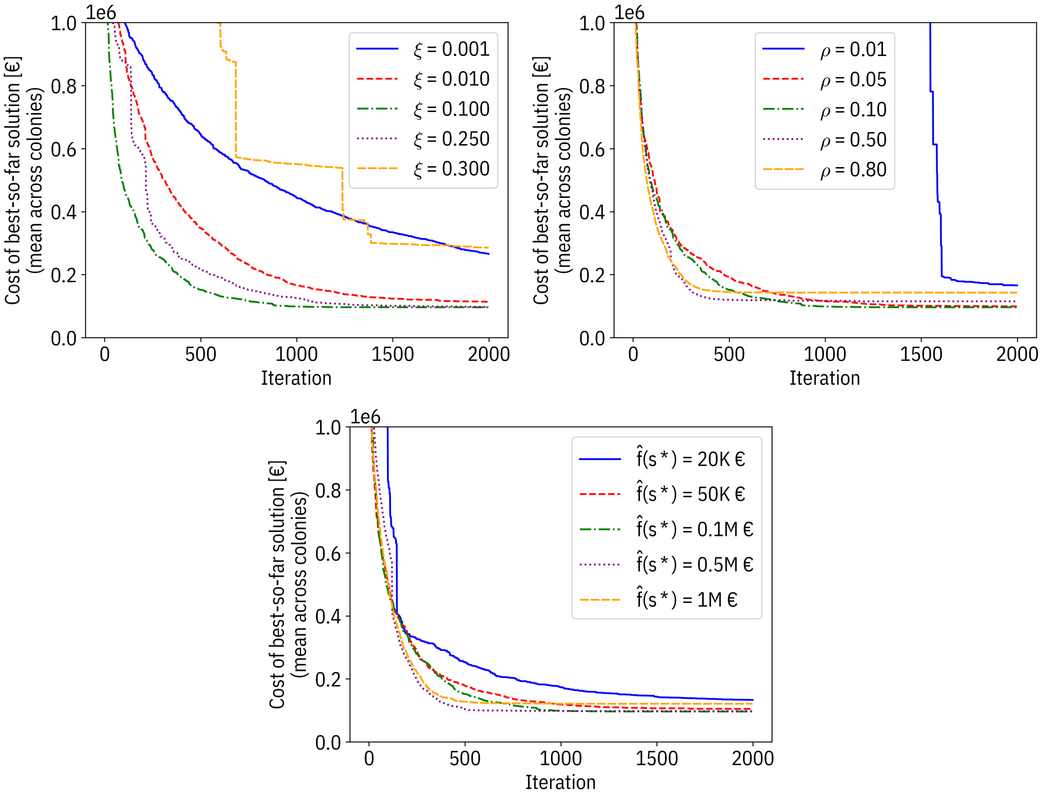


Figure 5.12: Effect of parameters controlling the pheromone update on the convergence of costs: (a) parameter ξ; (b) parameter ρ; and (c) parameter $\hat{f}(s^*)$

5.4.3 Parameters Controlling the Pheromone Update

Lastly, I evaluate parameters ξ, ρ, and $\hat{f}(s^*)$, which control the update of pheromones.

Parameter ξ adjusts the speed at which pheromone values deviating from base value τ_0 converge back to τ_0. A large range of settings ($0.01 \leq \xi \leq 0.25$) yields satisfactory results, as Figure 5.12a shows. Yet, AntPower's performance deteriorates beyond this range. To demonstrate why, Figure 5.13 shows how ξ influences the temporal dynamics of pheromone levels. With a large value of ξ (as in Figure 5.13a), the pheromone values converge back to base value $\tau_0 = 8 \times 10^{-3}$ rapidly (as indicated by the blue, horizontal lines that appear over and over again, and by the low maximum pheromone level). As a result, ants lack guidance and thus fail to concentrate on high-quality solutions. In contrast, with a low value of ξ (as in Figure 5.13c), the pheromone values return only slowly to their base level, so that the amount of newly added pheromones far outweighs the amount of pheromone that "evaporates" in each iteration (as indicated by the decreasing number of blue, horizontal lines). As a result, many solution components accumulate large amounts of pheromones (the highest pheromone level is more than four times as high as in Figure 5.13a). The high pheromone levels in turn "trap" the ants in neighborhoods of solutions found in previous iterations. Using $\xi = 0.1$ (as in Figure 5.13b, and as recommended by the authors of ACS [40]) yields an optimal balance between the two extremes shown in Figure 5.13a and Figure 5.13c.

Just as to parameter ξ, AntPower appears to be insensitive to changes in parameters ρ and $\hat{f}(s^*)$. To recall, ρ controls the global pheromone update, and $\hat{f}(s^*)$ estimates the cost of an optimal solution. According to Figure 5.12b and Figure 5.12c, the values for which AntPower shows best performance range from 0.05 to 0.5 for ρ, and from 50K € to 0.5M € for $\hat{f}(s^*)$. The best-performing setting for ρ, 0.1, equals the default setting recommended for ACS applications [40], just as it was the case for parameters k_{ant}, q_0, and ξ. The best-performing setting for $\hat{f}(s^*)$ is 0.1M €. That AntPower is insensitive to $\hat{f}(s^*)$, that is, to the estimated cost of the best expansion plan, is strongly in favor of AntPower's usability, because estimating this cost precisely would involve much effort.

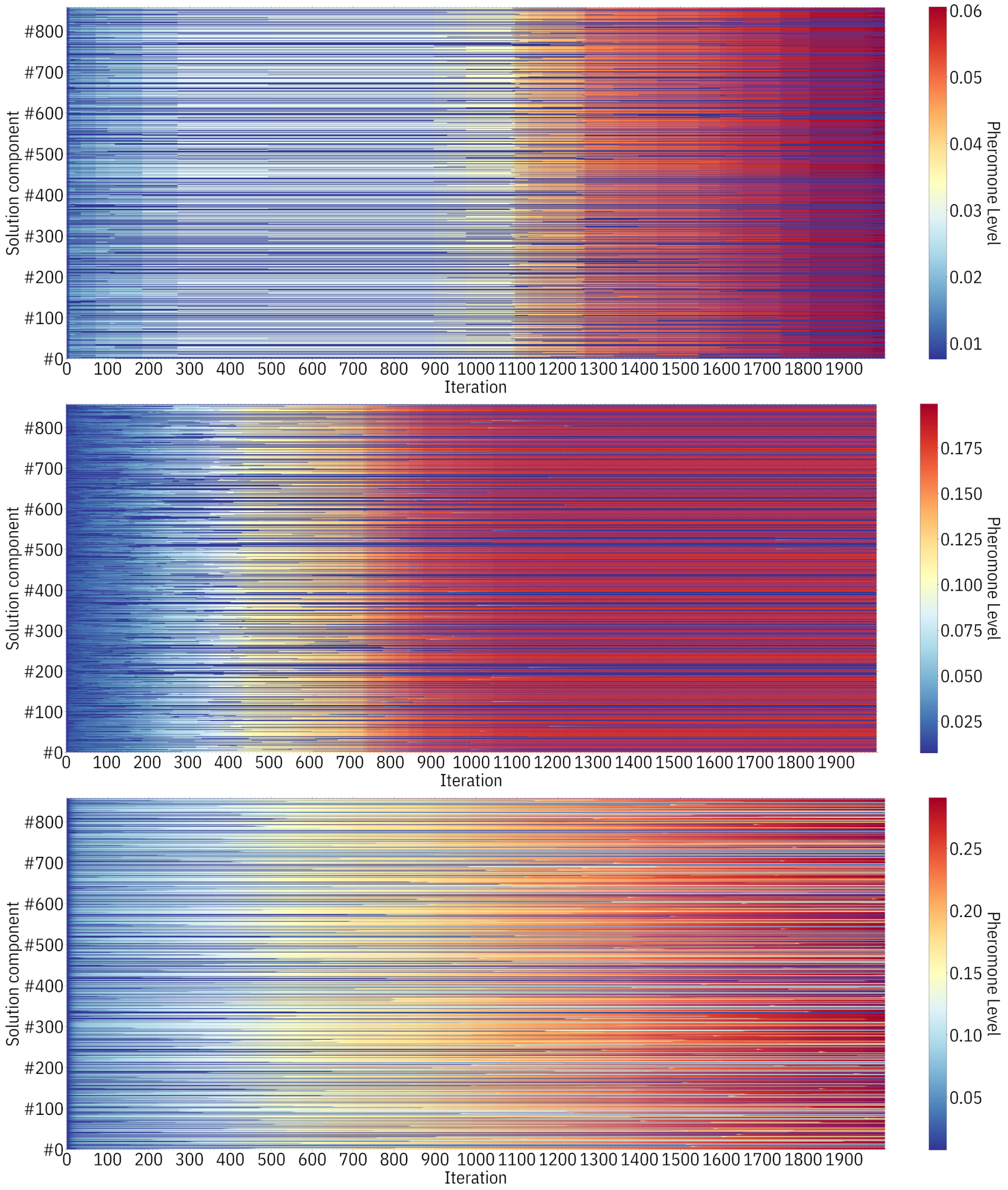


Figure 5.13: Temporal dynamics of pheromone levels for: (a) $\xi = 0.9$; (b) $\xi = 0.1$; and (c) $\xi = 0.001$

6 Conclusion

In this thesis, I presented AntPower, a software tool that facilitates the expansion planning of LV grids. I started by formulating the grid expansion problem that AntPower seeks to solve, and characterized it as a static, deterministic, single-objective, and non-linear optimization problem. As a method to solve this problem, I selected ACO, an optimization framework that outperformed other methods in previous grid planning research. Applying ACO to the grid expansion problem involved various design decisions, such as how to define the cost function and how to implement the ants' search strategy. To evaluate AntPower, I considered the real-world task of expanding an LV grid of 837 buses and five MV-LV transformers. I compared AntPower's performance to that of a conventional, manual planning method and a one-opt local search algorithm. Each of the three methods yielded expansion plans that combine the installation, reinforcement, and dismantling of line segments with the opening and closing of switches, thereby ensuring that the expanded grid meets all topological and electrical constraints. The expansion plan generated by AntPower turned out to be 60% cheaper than the manually created expansion plan, and 64% cheaper than that of the local search method. Eventually, a sensitivity analysis demonstrated AntPower's robustness against most of the control parameters, and indicated that, for the few sensitive control parameters, the default parameter settings of ACS are optimal.

Following are my main findings:

1. Planning grid expansion using ACO can reduce expansion costs tremendously, both compared to using a manual planning method that is based on expert knowledge and compared to using a local search algorithm.

2. That ACO has many control parameters does not pose a challenge in the context of LV grid expansion planning, because most control parameters are insensitive to changes in their values, and for the sensitive parameters, good default values exist.

3. Due to the computational complexity of high-accuracy PFA algorithms, using such an algorithm at the core of an automated grid planning tool is advisable only when considering few time steps. Otherwise, an aggregation of time steps or, alternatively, the use of a computationally less demanding PFA algorithm is necessary to achieve reasonable computation time.

I acknowledge that a holistic assessment of AntPower's performance requires further evaluation. In particular, AntPower should be benchmarked against other state-of-the-art grid planning tools (e.g., genetic algorithms) using an as diverse as possible set of realistic problem instances. The creation and publication of such problem instances would not only facilitate the evaluation of AntPower, but would likely fuel the advancement of grid planning methods in general. Another direction for future research is the generalization of AntPower to a wider range of grid expansion problems. Particularly, increasing AntPower's planning freedom (e.g., allowing the installation of new switches and transformers) would make AntPower more attractive to grid planners. Furthermore, adding support for different grid topologies would extend AntPower's scope of application from LV grids to MV and HV grids.

Bibliography

[1] C. Rehtanz et al. *Verteilnetzstudie für das Land Baden-Württemberg*. Ministry of the Environment, Climate Protection and the Energy Sector Baden-Württemberg, 2017. URL: https://www.baden-wuerttemberg.de/fileadmin/redaktion/m-um/intern/Dateien/Dokumente/5_Energie/Versorgungssicherheit/170413_Verteilnetzstudie_BW.pdf (visited on 05/18/2021).

[2] J. Büchner et al. *Moderne Verteilernetze für Deutschland*. German Federal Ministry for Economic Affairs and Energy, 2014. URL: https://www.bmwi.de/Redaktion/DE/Publikationen/Studien/verteilernetzstudie.html (visited on 05/18/2021).

[3] E. Hartvigsson et al. "Estimating National and Local Low-Voltage Grid Capacity for Residential Solar Photovoltaic in Sweden, UK and Germany". In: *Renewable Energy* 171 (2021). DOI: 10.1016/j.renene.2021.02.073. (Visited on 05/14/2021).

[4] J. von Appen et al. "Time in the Sun: The Challenge of High PV Penetration in the German Electric Grid". In: *IEEE Power and Energy Magazine* 11.2 (2013). DOI: 10.1109/MPE.2012.2234407.

[5] R. Moreno et al. "Planning Low-Carbon Electricity Systems Under Uncertainty Considering Operational Flexibility and Smart Grid Technologies". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 375.2100 (2017). DOI: 10.1098/rsta.2016.0305.

[6] M. Edeburn. *Electricity Sector Uncertainty Calls for New Decision-Making Tools*. Phys.org. 2017. URL: https://phys.org/news/2017-10-electricity-sector-uncertainty-decision-making-tools.html (visited on 06/14/2021).

[7] K. Kaufmann. "Distributed, Renewable and Uncertain: Grid Planning in the 21st Century". In: *Smart Electric Power Alliance* (2017). URL: https://sepapower.org/knowledge/grid-planning-21st-century/ (visited on 06/14/2021).

[8] German Federal Ministry for Economic Affairs and Energy. *An Electricity Grid for the Energy Transition*. URL: https://www.bmwi.de/Redaktion/EN/Dossier/grids-grid-expansion.html (visited on 05/14/2021).

[9] A. Agrillo et al. *Solare Fotovoltaico*. Gestore dei Servizi Energetici, 2019. URL: https://www.gse.it/documenti_site/Documenti%20GSE/Rapporti%20statistici/Solare%20Fotovoltaico%20-%20Rapporto%20Statistico%202018.pdf (visited on 05/24/2021).

[10] Bundesnetzagentur. *EEG in Numbers*. 2021. URL: https://www.bundesnetzagentur.de/EN/Areas/Energy/Companies/RenewableEnergy/Facts_Figures_EEG/FactsFiguresEEG_node.html (visited on 05/24/2021).

[11] E-Bridge Consulting GmbH. *Moderne Verteilernetze für Deutschland*. Final presentation. 2014. URL: https://www.bmwi.de/Redaktion/DE/Publikationen/Studien/verteilernetzstudie.html (visited on 06/18/2021).

[12] P. S. Georgilakis and N. D. Hatziargyriou. "A Review of Power Distribution Planning in the Modern Power Systems Era: Models, Methods and Future Research". In: *Electric Power Systems Research* 121 (2015). DOI: 10.1016/j.epsr.2014.12.010.

[13] A. R. Jordehi. "Optimisation of Electric Distribution Systems: A Review". In: *Renewable and Sustainable Energy Reviews* 51 (2015). DOI: 10.1016/j.rser.2015.07.004.

[14] P. A. Arias. "Planning Models for Distribution Grid: A Brief Review". In: *U.Porto Journal of Engineering* 4.1 (2018). DOI: 10.24840/2183-6493_004.001_0004.

[15] S. Khator and L. Leung. "Power Distribution Planning: A Review of Models and Issues". In: *IEEE Transactions on Power Systems* 12.3 (1997). DOI: 10.1109/59.630455.

[16] M. Dorigo. "Optimization, Learning and Natural Algorithms". Dissertation. Milan, Italy: Politechnico di Milano, 1992.

[17] J. Shu et al. "Enhanced Multi-Dimensional Power Network Planning Based on Ant Colony Optimization". In: *International Transactions on Electrical Energy Systems* 25.7 (2015). DOI: 10.1002/etep.1897.

[18] J. Gomez et al. "Ant Colony System Algorithm for the Planning of Primary Distribution Circuits". In: *IEEE Transactions on Power Systems* 19.2 (2004). DOI: 10.1109/TPWRS.2004.825867.

[19] A. M. Leite da Silva et al. "Reliability Worth Applied to Transmission Expansion Planning Based on Ant Colony System". In: *International Journal of Electrical Power & Energy Systems* 32.10 (2010). DOI: 10.1016/j.ijepes.2010.06.003.

[20] K. Lee and J. Vlachogiannis. "Optimization of Power Systems Based on Ant Colony System Algorithms: An Overview". In: *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*. Arlington, Virginia, USA: IEEE, 2005. DOI: 10.1109/ISAP.2005.1599237.

[21] S. Favuzza et al. "Optimal Electrical Distribution Systems Reinforcement Planning Using Gas Micro Turbines by Dynamic Ant Colony Search Algorithm". In: *IEEE Transactions on Power Systems* 22.2 (2007). DOI: 10.1109/TPWRS.2007.894861.

[22] N. Leeprechanon, P. Limsakul, and S. Pothiya. "Optimal Transmission Expansion Planning Using Ant Colony Optimization". In: *Journal of Sustainable Energy & Environment* 1.2 (2010). URL: http://www.jseejournal.com/journal.php?id=2 (visited on 05/24/2021).

[23] E. Carpaneto and G. Chicco. "Distribution System Minimum Loss Reconfiguration in the Hyper-Cube Ant Colony Optimization Framework". In: *Electric Power Systems Research* 78.12 (2008). DOI: 10.1016/j.epsr.2008.06.009.

[24] J. Alvarado et al. "Ant Colony Systems Application for Electric Distribution Network Planning". In: 2009 15th International Conference on Intelligent System Applications to Power Systems (ISAP). Curitiba, Brazil: IEEE, 2009. DOI: 10.1109/ISAP.2009.5352816.

[25] T. Paulun. "Strategische Ausbauplanung für elektrische Netze unter Unsicherheit". ISBN: 978-3-934318-77-9. Dissertation. Aachen, Germany: RWTH Aachen, 2007.

[26] N. Rotering. "Zielnetzplanung von Mittelspannungsnetzen unter Berücksichtigung von dezentralen Einspeisungen und steuerbaren Lasten". ISBN: 978-3-941704-26-8. Dissertation. Aachen, Germany: RWTH Aachen, 2013.

[27] L. Verheggen. “Kombinierte Grundsatzplanung von Mittel- und Niederspannungsnetzen unter Berücksichtigung betrieblicher Maßnahmen und Unsicherheiten”. ISBN: 978-3-941 704-62-6. Dissertation. Aachen, Germany: RWTH Aachen, 2017.

[28] W. Schufft and J. Backes, eds. *Taschenbuch der elektrischen Energietechnik*. München, Germany: Fachbuchverlag Leipzig im Carl-Hanser-Verlag, 2007. ISBN: 978-3-446-40475-5.

[29] A. von Meier. *Electric Power Systems: A Conceptual Introduction*. Hoboken, NJ, USA: John Wiley & Sons, Inc., July 13, 2006. ISBN: 978-0-470-03642-6.

[30] K. Heuck, K.-D. Dettmann, and D. Schulz. *Elektrische Energieversorgung: Erzeugung, Übertragung und Verteilung elektrischer Energie für Studium und Praxis*. 9., aktualisierte und korrigierte Auflage. Wiesbaden, Germany: Springer Vieweg, 2013. ISBN: 978-3-8348-1699-3.

[31] J. Kellermann. “Bewertung von Netzausbauplänen in Hochspannungsnetzen unter Berücksichtigung von betrieblicher Flexibilität und planerischen Unsicherheiten”. ISBN: 978-3-941704-88-6. Dissertation. Aachen, Germany: RWTH Aachen, 2019.

[32] B. Özalay. “Multikriterielles Verfahren zur Ausbauplanung zukünftiger Stromerzeugungsstrukturen”. ISBN: 978-3-95886-049-0. Dissertation. Aachen, Germany: RWTH Aachen, 2015.

[33] P. Cuffe and A. Keane. “Visualizing the Electrical Structure of Power Systems”. In: *IEEE Systems Journal* 11.3 (2017). DOI: 10.1109/JSYST.2015.2427994.

[34] P. Cuffe. *A Network Diagram of the ’Nesta Case1354 Pegase High Voltage’ Electrical Power System*. Wikimedia Commons. 2018. URL: https://commons.wikimedia.org/wiki/File:A_network_diagram_of_the_%27nesta_case1354_pegase_high_voltage%27_electrical_power_system.png (visited on 10/02/2020).

[35] F. Potratz. “Grundsatzplanung in der Mittel- und Niederspannung unter Berücksichtigung aktiver Netzbetriebsmittel”. ISBN: 978-3-95886-192-3. Dissertation. Aachen, Germany: RWTH Aachen, 2017.

[36] X. Tao. “Automatisierte Grundsatzplanung von Mittelspannungsnetzen”. ISBN: 978-3-934318-74-8. Dissertation. Aachen, Germany: RWTH Aachen, 2007.

[37] L. Bird, J. Cochran, and X. Wang. *Wind and Solar Energy Curtailment: Experience and Practices in the United States*. Technical Report. National Renewable Energy Laboratory (NREL) of the U.S. Department of Energy, 2014. DOI: 10.2172/1126842. (Visited on 07/27/2021).

[38] M. Albadi. “Power Flow Analysis”. In: *Computational Models in Engineering*. Ed. by K. Volkov. IntechOpen, 2020. ISBN: 978-1-78923-869-3.

[39] T. Brown, J. Hörsch, and D. Schlachtberger. “PyPSA: Python for Power System Analysis”. In: *Journal of Open Research Software* 6 (2018). DOI: 10.5334/jors.188.

[40] M. Dorigo and T. Stützle. *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004. ISBN: 978-0-262-04219-2.

[41] V. Neimane. “On Development Planning of Electricity Distribution Networks”. Dissertation. Stockholm, Sweden: Royal Institute of Technology, 2001.

[42] K. Schumacher. “Optimization Algorithms for Power Grid Planning and Operational Problems”. Dissertation. Michigan, MI, USA: University of Michigan, 2014. URL: http://hdl.handle.net/2027.42/107076 (visited on 11/05/2021).

[43] C. Blum. “Ant Colony Optimization: Introduction and Recent Trends”. In: *Physics of Life Reviews* 2.4 (2005). DOI: 10.1016/j.plrev.2005.10.001.

[44] M. Dorigo and L. Gambardella. “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”. In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997). DOI: 10.1109/4235.585892.

[45] M. Middendorf, F. Reischle, and H. Schmeck. “Multi Colony Ant Algorithms”. In: *Journal of Heuristics* 8.3 (2002). DOI: 10.1023/A:1015057701750.

[46] M. Arnold. “Planungsgrundsätze für Niederspannungsnetze unter Berücksichtigung regelbarer Ortsnetztransformatoren”. ISBN: 978-3-8440-6495-7. Dissertation. Kaiserslautern, Germany: Technische Universität Kaiserslautern, 2019.

[47] S. Meinecke et al. “SimBench—A Benchmark Dataset of Electric Power Systems to Compare Innovative Solutions Based on Power Flow Analysis”. In: *Energies* 13.12 (2020). DOI: 10.3390/en13123290.

[48] S. Patzack. “Ermittlung von planungsrelevanten Netznutzungsfällen für elektrische Verteilnetze”. ISBN: 978-3-941704-72-5. Dissertation. Aachen, Germany: RWTH Aachen, 2017.

List of Symbols

β	Parameter modulating the influence of heuristic values	*Real number*
η_i	Heuristic value of $c_i \in C$	*Real number*
$\hat{f}(s^*)$	Estimated cost of $s^* \in S^*$	*Currency*
$\hat{p}_l$	Transmission capacity of line segment l	*Real number*
λ	Maximum usable fraction of a line segment’s transmission capacity	*Percentage*
Ω	Constraints of Π	*Set*
Π	Optimization problem	*Tuple*
ρ	Parameter of the global pheromone update rule	*Percentage*
τ_0	Base level of pheromones	*Real number*
τ_i	Pheromone value of $c_i \in C$	*Real number*
θ	Planning horizon	*Time*
$\tilde{S}$	Feasible solutions	*Set*
$\tilde{X}$	Viable states	*Set*
ξ	Parameter of the local pheromone update rule	*Percentage*
B	Buses of G_{now}	*Set*
b_i	Connection point i of G_{now}	*Node*
C	Solution components	*Set*
$c_{\mathrm{cab}}(z)$	Market price of line type z, per kilometer	*Currency*
$c_{\mathrm{ins}}(z)$	Cost of installing a line of type z, per kilometer	*Currency*
c_{swi}	Cost of operating a switch	*Currency*
$c_{\mathrm{exp}}(G_{\mathrm{now}}, G_{\mathrm{exp}})$	Cost of the transition from G_{now} to G_{exp}	*Currency*
E^*	Edges of G^*	*Set*
E_{add}	Edges for the expansion of G_{now}	*Set*
E_{exp}	Edges of G_{exp}	*Set*

E_{now}	Edges of G_{now}	*Set*
f	Cost function of Π	*Function*
G^*	Optimally expanded version of G_{now}	*Graph*
G_{exp}	Expanded version of G_{now}	*Graph*
G_{now}	Grid that is to be expanded	*Graph*
k_{ant}	Number of ants	*Integer number*
k_{col}	Number of colonies	*Integer number*
k_{itr}	Number of iterations	*Integer number*
L_{add}	Line segments for the expansion of G_{now}	*Set*
L_{exp}	Line segments of G_{exp}	*Set*
L_{now}	Line segments of G_{now}	*Set*
m	Number of snapshots	*Integer number*
n	Size of C	*Integer number*
$p_b(t)$	Power at bus b and time t	*Real number*
$p_l(t)$	Power that flows through line segment l at time t	*Real number*
q_0	Parameter of the pseudorandom-proportional transition rule	*Probability*
S	Candidate solutions of Π	*Set*
S^*	Optimal solutions	*Set*
v_{max}	Upper limit on bus voltages	*Real number*
v_{min}	Lower limit on bus voltages	*Real number*
$v_b(t)$	Voltage at bus b and time t	*Real number*
W_{add}	Switches that are open in G_{now}	*Set*
W_{exp}	Switches that are closed in G_{exp}	*Set*
W_{now}	Switches that are closed in G_{now}	*Set*
X	Sequences over C	*Set*
Z	Types of all line segments in $L_{\mathrm{now}} \cup L_{\mathrm{add}}$	*Set*