UCL Université catholique de Louvain

École polytechnique de Louvain (EPL)

epl

Application of SDDP to SOCP and SDP relaxations of distribution networks

Dissertation presented by
Antoine LEGAT

for obtaining the Master's degree in
Mathematical Engineering

Supervisor
Anthony PAPAVASILIOU

Readers
François GLINEUR, Christian MERCKX

Academic year 2017–2018

Abstract

The growing influence of power generation in the distribution network poses an important challenge to power system operation. The environmental context and the recent developments in solar generation and power storage are profoundly changing the structure of power systems, adding distributed uncertain generation along with storage devices that could be capable of mitigate the variability of renewable resources. In addition to this, the modern communication infrastructure makes it possible to decentralize the operation of the network, relying on a hierarchical organization. The advantage of a decentralized approach is that, if possible, it is scalable to systems of arbitrary size.

The resulting optimization problem covers multiple time stages and must deal with the uncertainty brought by renewables. This is typically the class of problems tackled by the Stochastic Dual Dynamic Programming (`SDDP`) algorithm, developed in the framework of linear programming and which has found great commercial success. However, distribution networks are governed by a set of nonconvex constraints, Kirchhoff's laws, which render the problem challenging to solve efficiently. Under the ideal assumption of a balanced network, the problem can be relaxed into a second-order cone program (`SOCP`). Otherwise, if the network is unbalanced like in practical low-voltage distribution networks, we must resort to a semi-definite program (`SDP`).

The goal of this thesis is first to try to extend the `SDDP` algorithm to convex programming, i.e. the `SOCP` and `SDP` relaxations. To this end, these relaxations are first formally derived, and theoretical tightness conditions are provided. After that, the next chapter introduces the `SDDP` algorithm and argues that it can be extended to convex programming. Then, tightness and efficiency are experimentally tested, discussed and compared to commercial solvers on a 15-bus network. At last, we will study the influence of network imbalance, and try to understand its consequences on the solution and the computational effort required.

Acknowledgements

My first thanks go to my supervisor, Professor Papavasiliou, for his patience, tolerance and sincerity. I have not been the most regular student during the semester, and I was often ashamed of my poor weekly progress while entering your office. However, you were always kind and tolerant. After every meeting I was boosted because I did not want to disappoint you, I hope that this work will make you proud. Certainly, Arnaud Fabri did not lie to me when he warmly recommended me to work with you. In addition to you human qualities, you always showed interest in my findings and you were always there to help me in case of struggle.

My thanks also go to Professor François Glineur and Christian Merckx for agreeing to be part of the thesis jury. I hope that you will appreciate reading this document.

Then, I would also like to thank my older brother, Benoît for his precious help. In addition to your invaluable advice and endless expertise concerning `JuMP`, convex programming and by extension any subject in applied mathematics or computer science, you have always been an exemple for me, and I am proud of what you achieve.

Moreover, special thanks go to my parents, for their steady support and unfailing care. Your encouragement and advice have been crucial throughout my years of engineering studies, and thank you for everything.

Last but not least, I would like to thank all my family, simply for being who they are, because that defines who I am today.

Contents

List of Notations

Sets

$\hat{\mathcal{D}}_{\texttt{LP}}$	The dual feasible domain of `LP`. The same notation applies for any other problem.
$\mathcal{D}_{\texttt{LP}}$	The primal feasible domain of `LP`. The same notation applies for any other problem.
$\mathcal{E}$	The set of edges. $\mathcal{E} = \mathcal{N}^+$
$\mathcal{N}$	The set of nodes, $\mathcal{N} = \{0, \ldots, n\}$
$\mathcal{N}^+$	The set of nodes except node 0, $\mathcal{N}^+ = \{1, \ldots, n\}$
$\mathcal{T}$	The set of time stages, $\mathcal{T} = \{1, \ldots, H\}$
Ω_t	The set of possible realizations of uncertainty at stage t
$\Omega_{[t]}$	The set of possible histories of realizations of uncertainty at stage t
Φ_i	The set of phases of bus or line i, $\Phi_i = \{a, b, c\}$
C_i	The set of children of bus i

Indices

A_i	The ancestor of bus i

Single-Phase Variables

ℓ_i	Squared magnitude of the complex current at bus i, i.e. $\ell_i = \lvert I_i \rvert^2$
$bc_{i,t,\omega_{[t]}}$	Real power charged in the battery at bus i, time t, history $\omega_{[t]}$
$bd_{i,t,\omega_{[t]}}$	Real power discharged from the battery at bus i, time t, history $\omega_{[t]}$
I_i	Complex current flowing from bus i to its ancestor A_i
$S_i = P_i + \mathbf{i}Q_i$	Power flowing from bus i to its ancestor A_i
$s_i = p_i + \mathbf{i}q_i$	Power injected at bus i
V_i	Complex voltage at bus i
v_i	Squared magnitude of the complex voltage at bus i, i.e. $v_i = \lvert V_i \rvert^2$
$x_{i,t,\omega_{[t]}}$	Real energy stored in the battery at bus i, time t, history $\omega_{[t]}$

Three-Phase Variables

$\boldsymbol{\ell}_i$	$\begin{bmatrix} I_i^a & I_i^b & I_i^c \end{bmatrix}^T$
$\mathbf{bc}_{i,t,\omega_{[t]}}$	$\begin{bmatrix} bc^a_{i,t,\omega_{[t]}} & bc^b_{i,t,\omega_{[t]}} & bc^c_{i,t,\omega_{[t]}} \end{bmatrix}^T$

$\mathbf{bd}_{i,t,\omega_{[t]}}$ $\begin{bmatrix} bd^a_{i,t,\omega_{[t]}} & bd^b_{i,t,\omega_{[t]}} & bd^c_{i,t,\omega_{[t]}} \end{bmatrix}^T$

$\mathbf{S}_i$ Complex matrix with diagonal elements denoting power flowing from bus i to its ancestor A_i, $\mathbf{S}_i \in \mathbb{C}^{|\Phi_i|\times|\Phi_i|}$. The off-diagonal elements do not have a physical interpretation

$\mathbf{s}_i$ $\begin{bmatrix} s^a_i & s^b_i & s^c_i \end{bmatrix}^T$

$\mathbf{v}_i$ $\begin{bmatrix} V^a_i & V^b_i & V^c_i \end{bmatrix}^T$

$\mathbf{x}_{i,t,\omega_{[t]}}$ $\begin{bmatrix} x^a_{i,t,\omega_{[t]}} & x^b_{i,t,\omega_{[t]}} & x^c_{i,t,\omega_{[t]}} \end{bmatrix}^T$

$bc^{\phi}_{i,t,\omega_{[t]}}$ Real power charged in the battery at bus i, time t, history $\omega_{[t]}$, phase ϕ

$bd^{\phi}_{i,t,\omega_{[t]}}$ Real power discharged from the battery at bus i, time t, history $\omega_{[t]}$, phase ϕ

I^{ϕ}_i Complex current flowing from bus i to its ancestor A_i for phase ϕ

s^{ϕ}_i Complex power injected at bus i for phase ϕ

V^{ϕ}_i Complex voltage at bus i for phase ϕ

$x^{\phi}_{i,t,\omega_{[t]}}$ Real energy stored in the battery at bus i, time t, history $\omega_{[t]}$, phase ϕ

Single-Phase Parameters

ηc_i Battery charge efficiency at bus i

ηd_i Battery discharge efficiency at bus i

$bc_{\max,i}$ Maximal battery charge rate at bus i

$bd_{\max,i}$ Maximal battery discharge rate at bus i

$MC_{i,t}$ Marginal cost of power generation at bus i, time t

$p_{\max,i,t}$ Maximal real power generation at bus i, time t

PL Power loss constant

$q_{\max,i,t}$ Maximal reactive power generation at bus i, time t

$sD_i = pD_i + \mathbf{i}qD_i$ Power demanded at bus i

$v_{\max,i}$ Maximal squared voltage magnitude at bus i

$v_{\min,i}$ Minimal squared voltage magnitude at bus i

$x_{\text{init},i}$ Initial energy stored in the battery at bus i, before time 1

$Y_i = G_i - \mathbf{i}B_i = 1/Z_i$ Admittance of bus i

$Z_i = R_i + \mathbf{i}X_i = 1/Y_i$ Impedance of the line connecting bus i to its ancestor A_i

Three-Phase Parameters

$\mathbf{bc}_{\max,i}$ $\begin{bmatrix} bc^a_{\max,i} & bc^b_{\max,i} & bc^c_{\max,i} \end{bmatrix}^T$

$\mathbf{bd}_{\max,i}$ $\begin{bmatrix} bd^a_{\max,i} & bd^b_{\max,i} & bd^c_{\max,i} \end{bmatrix}^T$

$\mathbf{p}_{\max,i,t}$	$\begin{bmatrix} p^a_{\max,i,t} & p^b_{\max,i,t} & p^c_{\max,i,t} \end{bmatrix}^T$
$\mathbf{q}_{\max,i,t}$	$\begin{bmatrix} q^a_{\max,i,t} & q^b_{\max,i,t} & q^c_{\max,i,t} \end{bmatrix}^T$
$\mathbf{sD}_i$	$\begin{bmatrix} sD^a_i & sD^b_i & sD^c_i \end{bmatrix}^T$
$\mathbf{v}_{\max,i}$	$\begin{bmatrix} v^a_{\max,i} & v^b_{\max,i} & v^c_{\max,i} \end{bmatrix}^T$
$\mathbf{v}_{\min,i}$	$\begin{bmatrix} v^a_{\min,i} & v^b_{\min,i} & v^c_{\min,i} \end{bmatrix}^T$
$\mathbf{x}_{\max,i}$	$\begin{bmatrix} x^a_{\max,i} & x^b_{\max,i} & x^c_{\max,i} \end{bmatrix}^T$
$\mathbf{Y}_i$	Phase admittance matrix of bus i, $\mathbf{Y}_i \in \mathbb{C}^{\lvert\Phi_i\rvert\times\lvert\Phi_i\rvert}$
$\mathbf{Z}_i$	Phase impedance matrix of the line connecting bus i to its ancestor A_i, $\mathbf{Z}_i \in \mathbb{C}^{\lvert\Phi_i\rvert\times\lvert\Phi_i\rvert}$
$bc^{\phi}_{\max,i}$	Maximal battery charge rate at bus i, phase ϕ
$bd^{\phi}_{\max,i}$	Maximal battery discharge rate at bus i, phase ϕ
$MC^{\phi}_{i,t}$	Marginal cost of power generation at bus i, time t, phase ϕ
$p^{\phi}_{\max,i,t}$	Maximal real power generation at bus i, time t, phase ϕ
$q^{\phi}_{\max,i,t}$	Maximal reactive power generation at bus i, time t, phase ϕ
sD^{ϕ}_i	Complex power demanded at bus i for phase ϕ
$v^{\phi}_{\max,i}$	Maximal squared voltage magnitude at bus i, phase ϕ
$v^{\phi}_{\min,i}$	Minimal squared voltage magnitude at bus i, phase ϕ
$x^{\phi}_{\max,i}$	Battery capacity at bus i, phase ϕ

Abbreviations

`ACOPF`	Alternating Current Optimal Power Flow
`AC`	Alternating Current
`ADMM`	Alternating Direction Method of Multiplier
`BFM`	Branch Flow Model
`BIM`	Bus Injection Model
`DCOPF`	Direct Current Optimal Power Flow
`DC`	Direct Current
`DERs`	Distributed Energy Resources
`LP`	Linear Program(ming)
`NLDS`	Nested L-shaped Decomposition Subproblem
`OPF1-ar`	The problem obtained after performing the angle relaxation on the `OPF1`
`OPF1`	Single-Phase Optimal Power Flow
`OPF3rank`	The equivalent formulation of `OPF3` with the `SDP` and rank constraints

`OPF3`	Three-Phase Optimal Power Flow
`OPF`	Optimal Power Flow
`SDDP`	Stochastic Dual Dynamic Programming
`SDP`	Semi-definite Program(ming)
`SOCP`	Second-Order Cone Program(ming)
swing	The swing bus, cfr section 3.1.1

Mathematics

$\|z\|$	The norm of z, i.e. $\|z\| = \sqrt{\Re(z)^2 + \Im(z)^2}$, or the cardinality of a set
$[t]$	$\{1, \ldots, t\}$
$\text{diag}(\mathbf{M})$	The column vector composed of the diagonal of matrix $\mathbf{M}$
$\Im(z)$	The imaginary part of z
$\mathbf{i}$	The imaginary number, i.e. $\mathbf{i} = \sqrt{-1}$
$\mathbf{M} \succeq 0$	Matrix $\mathbf{M}$ is positive semi-definite
$\mathbf{M}$	An upper boldface letter denotes a matrix
$\mathbf{M}^H$	The hermitian transpose of matrix $\mathbf{M}$, i.e. $\mathbf{M}^H = \mathbf{M}^{T*}$
$\mathbf{v}$	A lower boldface letter denotes a column vector
rank $\mathbf{M}$	The rank of matrix $\mathbf{M}$
$\Re(z)$	The real part of z
z^*	The complex conjugate of z, i.e. $z^* = \Re(z) - \mathbf{i}\Im(z)$

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Energy Challenge

Benjamin Franklin must be turning over in his grave. How could he even conceive the idea that the extensive research he conducted on electricity [9] would eventually lead more than two centuries later to the massive power systems as we know them today ? Since then, researchers have understood this physical curiosity to its core and discovered how to make the best use of their findings. Indeed, the considerable scientific effort put on that physical curiosity during all those years led to some of the most complex and expensive entities in the world nowadays: power systems. Today, electricity flows all over the world and one could not easily imagine our modern society without it. Our institutions and infrastructures rely heavily on it, such that electricity has become a fundamental resource. However, even today, 17% of people around the world still live with no access to electricity [3]. Furthermore, the current world population is 7.6 billion as of July 2018 according to the most recent United Nations estimates and is still growing. Besides, the traditional ways of generating power are no longer acceptable due to their environmental footprint, and there is a growing concern for green alternatives. Subsequently, providing environmentally friendly power to the growing population of human beings, including the ones living in a precarious situation, is one of the most crucial challenges of our time.

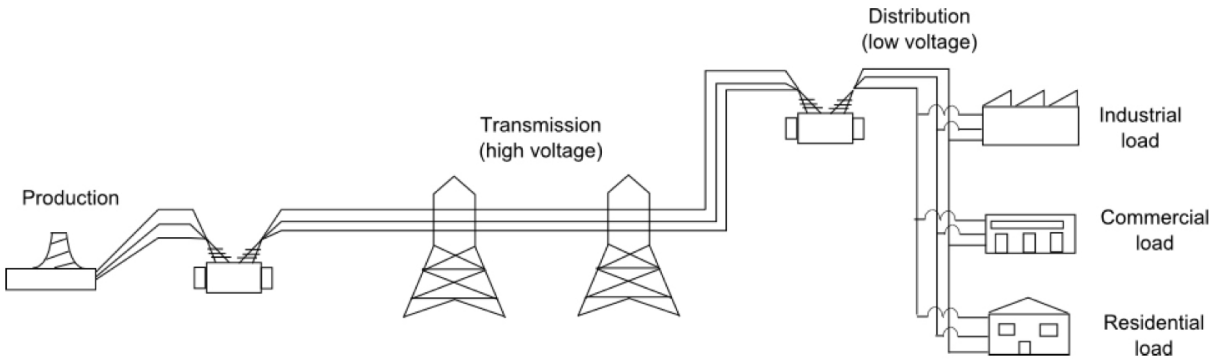


Figure 1.1: The electricity supply chain, taken from [1].

From its production in Tihange (for instance) to the power outlet which supplies your computer in the Euler building, electrical energy has travelled 50 km. Let us take a closer look at what it went through, see figure 1.1. After its generation in the power plant at low voltage, it goes into a first transformer that converts it to high voltage for efficient transport. Indeed, as we will see in chapter 3, power loss depends on the current and by increasing the voltage, the current is decreased since the power, which is the product of the two latter quantities, is conserved. The

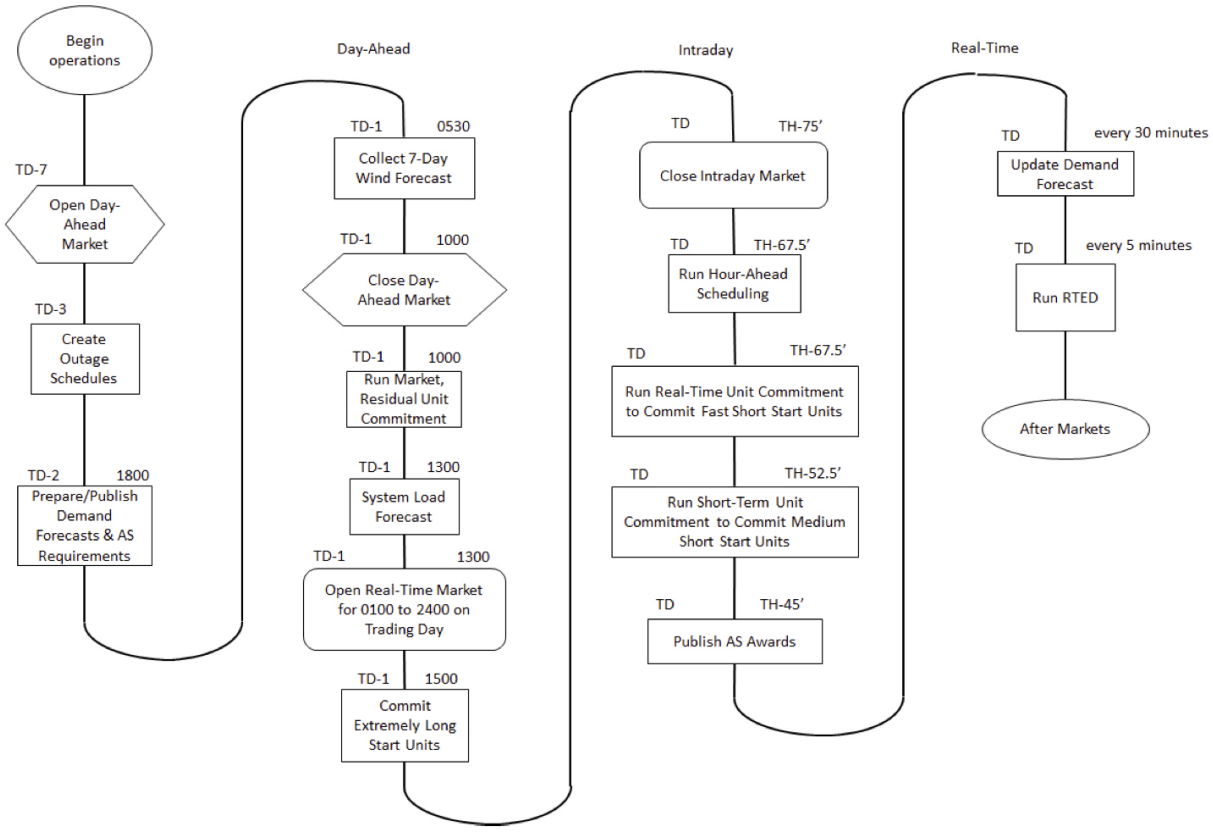


Figure 1.2: A typical flow chart of system operations. TD and TH stand for Trading Day and Trading Hour respectively. Taken from [1].

power is thus transported through transmission lines over long distances with minor loss. It then reaches the second transformer, close to the Euler building. This transformer delineates the border between the **Transmission** network (left in Figure 1.1), and the **Distribution** network (right in Figure 1.1). This transformer, called substation transformer, converts the high voltage electricity to low voltage electricity that can be consumed over its downstream substation. It is then carried through distribution lines, that operate at low voltage but on shorter distances, until its destination, typically classified between industrial, commercial and residential load. Note that industrial consumers may also be connected to the distribution network.

There are traditionally three actors on this chain. First, the system operator is in charge of the high-voltage transmission network and is responsible for 'keeping the lights on' [1]. Then, distribution system operators manage the low-voltage substations, ensuring that power is delivered in acceptable levels, mostly 220 – 230 Volts in Europe. Eventually, consumers are connected to the distribution network. They may produce power as well, e.g. if they own solar panels, in which case they can sell it and inject it into the distribution network. In order to conclude this quick overview, we attach a typical US flow chart of system operations in Figure 1.2. We observe that it is a particularly strict process and models are run literally every 5 minutes in the USA in order to meet the demand as precisely as possible. For further comments about this, we invite you to consult [1].

Today, mankind is facing the greatest ultimatum of the century. Due to the human activity, our planet as we have known it so far is vacillating and we must take urgent action before we reach the point of no return. In 40 years, we have lost 50 % of the vertebrate population, 80 % of flying bugs these 30 last years in Europe, 30 % of the birds. On the geopolitical front, countries will wage war for the control of natural resources. Populism and xenophobia are already rising today

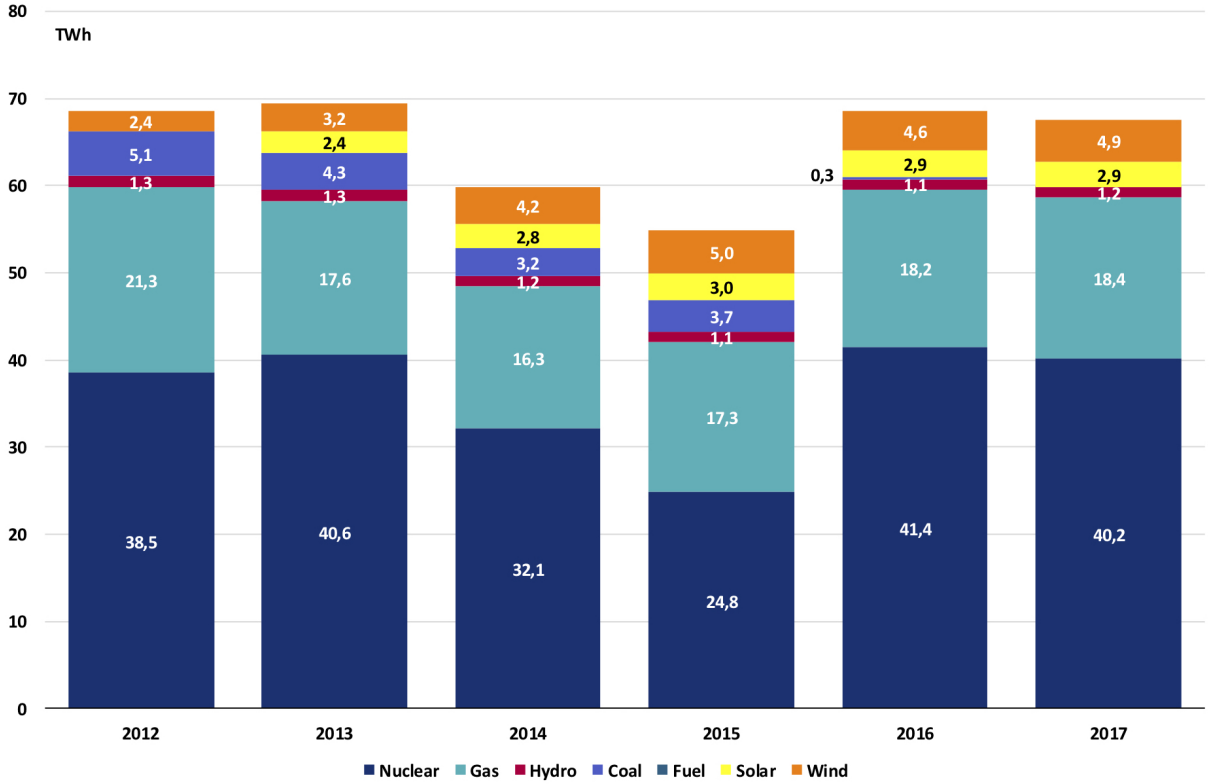


Figure 1.3: The power mix in Belgium, taken from [2].

in our countries, and the United Nations predicts 250 million climate refugees by 2050, offering a fertile ground for radical right politics [10]. This constitutes the most critical challenge of our generation, and a substantial part of the solution lies in power generation. Indeed, traditional ways of producing power appear to be noticeably damaging for the environment. Burning fossil fuels contributes to the greenhouse effect, and nuclear power plants, in addition to constantly threatening the people living around, provide radioactive waste that we are not able to treat safely. Without the shadow of a doubt, these energy sources are not sustainable. However, we are still extensively depending on them, as you can see in Figure 1.3. This constitutes a compelling call for sustainable alternative ways of producing power, and solutions have been available for a few years already. Taking advantage of the free energy provided by the wind and exploiting the almost infinite energy of the sun, wind turbines and solar panels are the energy of the future.

Combined heat and power (CHP) systems[1], also known as cogeneration, generate electricity and useful thermal energy in a single, integrated system. While the conventional method of producing usable heat and power separately has a typical combined efficiency of 45 %, CHP systems can operate at levels as high as 80 % [11]. Thanks to its flexibility, this technology can balance the erratic and uncertain behavior of solar and wind. Electric vehicles provide another solution to mitigate this issue. Indeed, batteries such as Tesla Powerwalls could be used to store power during the peak power production, and restore it when needed. Furthermore, these alternatives are more than castles in the air. More than half of Norway's new car sales were electric or hybrid in 2017, as the Norwegian Road Federation calculated. And even though Belgium does not seem to be well engaged in light of figure 1.3, the European Union has set a target of 20 % energy coming from renewable sources in 2020 [12]. Eventually, solar panel prices dramatically dropped recently because the fragmented industry has consolidated around a single technology, which should hopefully involve a rise of their installations.

[1]One could highlight the fact that this is the technology that UCL is using to heat most university buildings and produce its power.

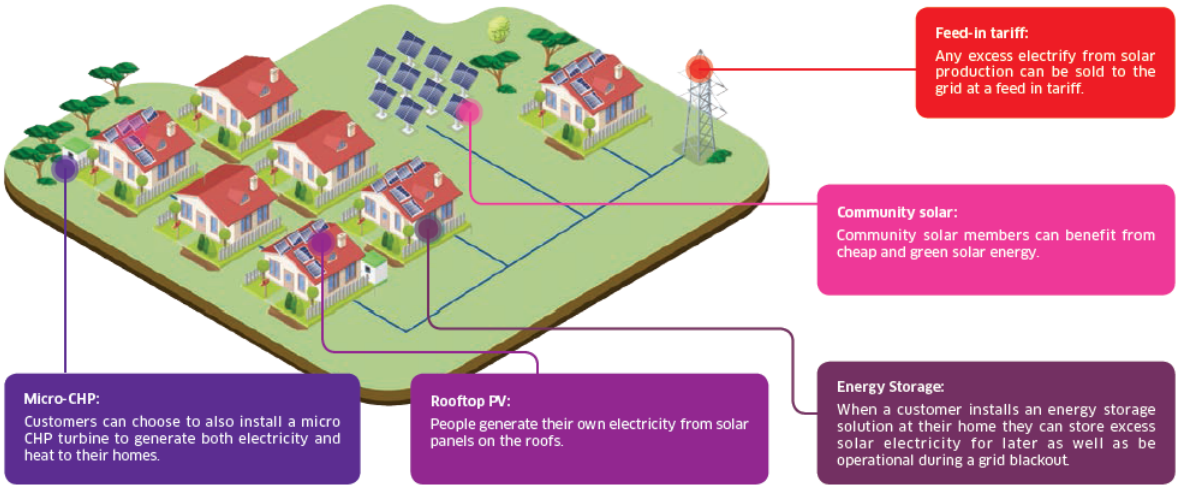


Figure 1.4: Illustration of a residential microgrid. Taken from [3].

A common particularity between these salutary technologies is their localization in the distribution network. In the traditional setup, power is unidirectionally routed from the few large power plants over long distances to the consumers. On the economic front, installing such extended infrastructure implies an important fixed cost and significant maintenance cost. Moreover, transmitting power over long distances, even at high voltage, incurs substantial power loss.

1.2 A Decentralized Approach

Consequently, one needs to change the power system paradigm. From this ancient centralized unidirectional network, we must move to a decentralized, smart, flexible and resilient system: the cornerstone of today's energy challenge. The solution lies in **Microgrids**: a microgrid is an energy supply network built around local power and heat generation facilities. It is designed to operate autonomously or in synchronization with a national grid within a clearly defined area [3]. As you can see in Figure 1.4, a microgrid makes the best use of distributed generation (DG) resources. Since power is generated locally, we spare the loss on the transmission lines. The needed infrastructure being more modest, it is truly cheaper to set up in comparison to large power plants and a wide network. In addition, an electrical power system with a high penetration rate of microgrids is resilient to terrorist/cyber attacks, technological failures, a global shortage of supply or disastrous meteorological conditions [13]. Moreover, the local combination of CHP, solar panels and storages offers the flexibility required by solar panels and users can adapt their consumption to the intensity of the generation.

Operating a grid with a high penetration of renewables at large scale is challenging due to the important number of variables and uncertainty [14]. Nevertheless, decentralizing gives the opportunity to operate simultaneously subparts of the network, making the implementation of scalable algorithms possible. A distributed approach allows us to exploit parallelization. On the social front, a microgrid can be managed by the community living with it. With this responsibility, the people would take more sustainable decisions than the capitalist 100-year monopoly in charge so far. Energy would truly belong to the people, as well as the infrastructure. Building and maintaining it would participate to the local economy we are missing today. John Farrell, known as the guru of distributed energy, even compares power with food or waste: in

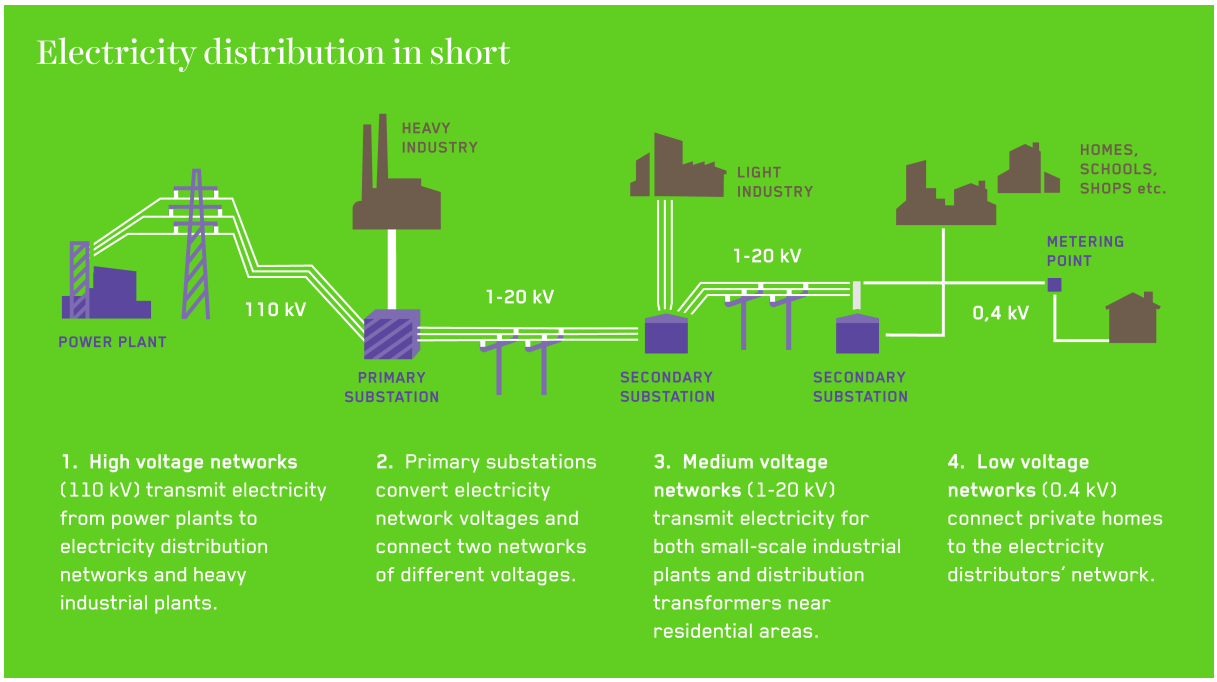


Figure 1.5: A more detailed illustration of distribution networks. This illustration concerns Finland, in most European countries low voltage is set at 220 – 230 V instead of 400 V. Taken from [4].

these three cases, decentralization is the key to the corresponding modern issue (i.e. the energy, food and waste issues) [15]. Some countries have already taken this step: we can cite Denmark and Germany, in their quest for greener energy, where microgrids are owned by cooperatives. Similarly, Japan and the North-East of the USA, victim of hurricane Sandy, are also shifting to microgrids for a less vulnerable network. To summarize, Dall'Anese et al. wrote: *deployment of microgrids promises drastic performance enhancement of the distribution grid in terms of efficiency and stability, along with increased network scalability and resilience to outages* [16].

While we were used to only optimize the transmission network, operating microgrids implies optimizing the distribution network. The latter is slightly more advanced than as depicted in Figure 1.1. As you can see in Figure 1.5, it is split into medium voltage networks and low-voltage networks. On this same figure, you may also note that in this low voltage network, only one line is sketched instead of three. This represents the fact that houses are only connected to one phase whereas the rest of the network is composed of three phases. Indeed, since power is traditionally created by a mechanical rotation, it is obtained in a waveform: it is referred to as alternating current (`AC`). In order to keep the intensity within an acceptable range, most power systems worldwide chose to use three phases of the same amplitude, each separated by $2\pi/3$ radians, as illustrated in Figure 1.6.

- When the three phases are exactly as depicted on this figure, we say that the power flow is **balanced** or **symmetric**. The word *symmetric* comes from the fact that the three line conductors have to be disposed at the same distance from each other (i.e. in the corners of an equilateral triangle), otherwise the inductances will be different, leading to unequal voltages. It is the case in the high and medium voltage networks, thus we can only keep one phase in their modelization since the two others can be inferred from the latter.
- On the other hand, if the three signals do not have a perfect sine form, are not separated exactly by $2\pi/3$ radians or have different magnitudes, the flow is said to be **unbalanced** or **asymmetric**. This is typically the case in low-voltage networks, since different houses

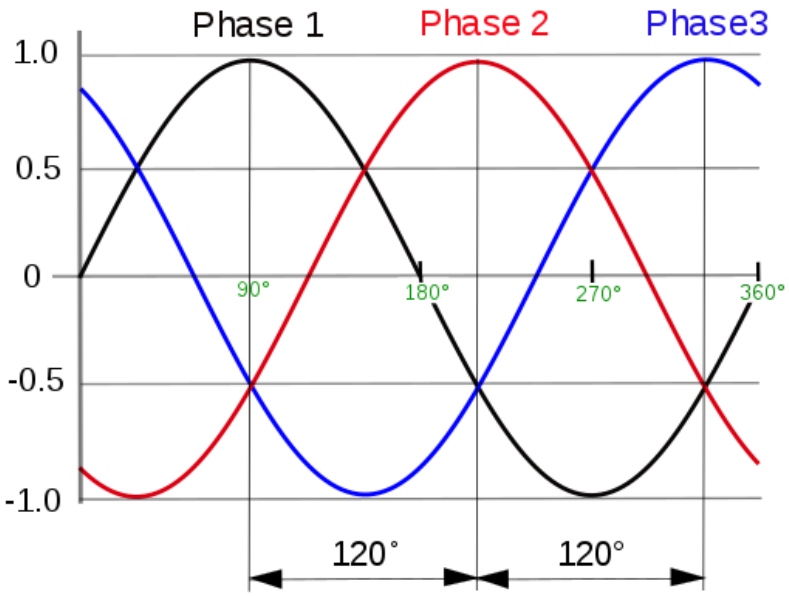


Figure 1.6: AC three-phase voltage, current or power in the ideal balanced case. Taken from [5].

are connected to each phase. In addition, non-equilateral conductor spacings of three-phase line segments are involved and single-phase DG units may worsen the network imbalance [17].

The operation of an unbalanced network is challenging because three times as many variables are involved, and nonlinear relations link the three phases as we will see in chapter 3. The most efficient models usually applied to distribution networks are second-order cone programming (`SOCP`) relaxations, but to the best of my knowledge nobody managed to adapt them on an unbalanced setup. Providentially, researchers recently achieved to formulate the unbalanced case as a semi-definite programming (`SDP`) relaxation [16]. `SDP` relaxations are not as efficient as `SOCP`s and do not scale for extremely large systems [18], but it should not be an issue since we adopt a decentralized approach. Over the last decade, semi-definite programming has been an active research field and solvers are becoming increasingly efficient [19]. One of the most promising ones, `Mosek`, implements a primal-dual interior point method and is widely employed in the financial, energy and forestry industry [20]. This is the one for which we opted in this work.

1.3 Our Objective

The goal of this work is to operate a microgrid similar to the one depicted in Figure 1.4. We consider that each house owns an energy storage, thus the battery state of the present has consequences on the future. Therefore, the optimization needs to be performed over multiple time stages. In addition, solar panels bring uncertainty to the net demand, such that we end up with **a multistage optimization problem under uncertainty.** This is typically the class of problems tackled by the **Stochastic Dual Dynamic Programming** (`SDDP`) algorithm, designed by Pereira and Pinto in 1991 [21].

Relying on the combination of dynamic programming, cutting plane methods and a decomposition of the mathematical program in order to exploit parallel computing, the `SDDP` algorithm has found great commercial success. The `SDDP` algorithm was initially developed in the framework of

linear programming, and to the best of my knowledge, has not been applied to nonlinear programs so far. Indeed, it requires to solve many subproblems a substantial amount of times, such that SDDP was only tractable for linear programs before. Nevertheless, thanks to the extensive progress achieved regarding nonconvex solvers [22], we are able to extend SDDP to SOCP and SDP in this work.

We will also consider a linear approximation (LP) of our problem which will be used for purpose of comparison. Putting it all together, we will:

- apply SDDP on an LP approximation, in the balanced case
- apply SDDP on an SOCP relaxation, in the balanced case
- apply SDDP on an SDP relaxation, in the balanced and unbalanced case

for a microgrid similar to the one depicted in Figure 1.4. In order to discuss the efficiency of these implementations, we will also solve the problem as a whole using a commercial solver. The ideal balanced case being an approximation of the practical unbalanced case, the unbalanced solution given by SDP will be more accurate, and will lead to a better operation. On the other hand, it will require more computational resources than the two other ones. The questions that this document will address are the following:

1. Is it possible to extend the SDDP algorithm to nonlinear programs?
2. If so, is this algorithm worth the implementation, compared to the efficiency achieved by commercial solvers?
3. Study the influence of network imbalance, and try to understand its consequences on
   (a) the solution;
   (b) the computational effort required.

Our broader objective is to analyze the possibility of a hierarchical organization of electricity markets whereby pockets of decision making are developed that are capable of managing storage/flexibility, in order to mitigate local uncertainty/variability of renewable resources. This hierarchical organization is inspired by the fact that in radial distribution systems, agents interact weakly, in the sense that they only interact on the interface edges so as to balance the power leaving the higher level of the hierarchy and entering the lower level of the hierarchy. The advantage of a hierarchical organization is the fact that, if it is possible, it becomes scalable to systems of arbitrary size.

This thesis is organized as follows. First, a literature review about the optimal power flow and its relaxations is presented. Then, the three addressed models are derived, and theoretical conditions ensuring tightness are provided. After that, the SDDP algorithm is presented, and we argue that it can be extended to our convex programs. Next, this algorithm is tested on a small 15-bus network for the three formulations. Tightness and efficiency are discussed from this experiment, before numerically testing the effect of imbalance on the same network. At last, the three above questions can be answered along with additional observations, and perspectives are provided for potential future work on the subject.

Chapter 2

State of the Art

2.1 Optimal Power Flow Convex Relaxations

The Optimal Power Flow (OPF) is the problem of dispatching power over a network while minimizing the generation cost or power loss. Supply and demand need to be balanced, and capacity, stability and contingency constraints must be satisfied. More significantly, power cannot be routed arbitrarily over the network. The voltage, current and complex power are related by nonconvex physical rules, Kirchoff's laws, that must be respected in order to have a feasible flow. These latter constraints cause the domain of the OPF to be nonconvex, which is regrettable since this problem is used massively nowadays. All over the world, system operators solve it to do long-term planning, days- to hour-ahead scheduling, real-time dispatch (every few minutes) ..., making the OPF one of the most frequently employed optimization routines in power systems [23]. In addition to the substantial industrial interest, the OPF is highly nonconvex and NP-hard, and has therefore been an active research field since Carpentier's first formulation in 1962 [24]. A lot of nonlinear optimization algorithms have been proposed for solving this nonconvex problem, see surveys [25, 26, 27]. Most of these methods are based on the Karush-Kuhn-Tucker (KKT) necessary conditions, which can only guarantee a locally optimal solution [28]. Researchers have therefore oriented their research towards convex relaxations of the OPF rather than trying to solve this nonconvex problem efficiently.

In the late 1880s, Thomas Edison and George Westinghouse fought the so-called War of the Currents to decide whether the incumbent direct current or Nicola Tesla's alternating current technology would become the standard for future power systems [23]. As you may know, Westinghouse won with the AC, and the OPF, also called ACOPF, is defined on alternating current. Nevertheless, a popular approximation is called the direct current power flow problem (DCOPF), ironically recalling Edison's defeat. DCOPF is obtained by linearizing Kirchoff's laws under some assumptions inspired from typical power flow systems [29, 30]. It is a linear program, which makes it easy to solve and is still used nowadays by power system operators when the more elaborate methods are not scalable or not efficient enough [31, 32]. DCOPF was the only convex option for power system optimization until 2006, when the SOCP relaxation was first proposed.

Although the DCOPF is widely used for some power system applications, it does not account for voltage magnitudes and reactive powers, which can lead to non-physical solutions. On the other hand, convex relaxations of the OPF manifest several advantages even though they cannot compete with DCOPF regarding efficiency. First, one can always check whether the relaxation is exact, i.e. if the obtained solution satisfies the original problem. Second, an infeasible relaxation certifies that the original problem is infeasible as well. Over the last decade, researchers have been actively developing two convex relaxations: second-order cone programs (SOCP) and semi-definite

programming formulations (SDP). In [33], Steven H Low lists the inconveniences of DCOPF and compares the two latter convex relaxations.

In 2006, Jabr et al. pioneered the first convex relaxation of the OPF. Their model is formulated as an SOCP on a radial distribution system, enabling them to find the solution in polynomial time using interior-point methods [34]. Building on this breakthrough, Taylor et al. proposed other conic relaxations on radial networks [35, 36]. Inspired by 20-year-old publications [37, 38], Farivar et al. proved that Jabr's relaxation is exact for radial networks, provided there are no upper bounds on loads [6]. For meshed networks, this result does not hold but they developed solution recovery algorithms. Many publications have used Jabr's model: [39, 40, 8, 41] to name a few. The nonconvexity of the OPF problem stemming notably from the arctangent function on voltage angles, Kocuk et al. recently proposed three new SOCP relaxations approximating the arctangent [42].

Notwithstanding the valuable performances of the latter relaxation, it ignores potential system imbalance. As described in chapter 1.2, imbalance is a non-negligible phenomenon that arises in low-voltage distribution grids, notably due to single-phase loading. In order to address this issue, we need to wait for the semi-definite relaxation, which was first proposed in 2008 by Bai et al. [43]. They formulated the OPF as a quadratically constrained quadratic program, before approximating it by a semi-definite program. SDP was already an active research field by that time [44, 45], so they could efficiently solve their model by primal-dual interior point algorithms on several IEEE test systems. In [46], another SDP based on the Lagrangian dual of the OPF was proposed by Lavaie et al., along with a necessary and sufficient condition for a zero duality gap, that guarantees the recovery of the optimal OPF solution. For both models, the exactness of the relaxation can be assessed by checking the unitary of the rank of the voltage-related matrix and the authors observed that it is the case for many practical instances. Examining the geometry of the OPF, Lavaei et al. then provided sufficient conditions on the exactness of a convex relaxation on a radial network [47]. [48] extended this work by taking into account limits on the reactive power injections, and considering tight voltage magnitude constraints instead of fixed values. Interestingly, [49] recently presented an example of OPF for which the SDP relaxation fails for one formulation but succeeds in finding the global solution to another formulation, revealing that success of the SDP relaxation must involve factors beyond just the network physics.

Yet, these SDP publications still assume a single-phase network, while distribution networks are typically multiphase and unbalanced [50]. For multiphase *balanced* networks, single-phase methods can be applied by only keeping one phase, from which the others can be inferred. But when unequal single-phase loads must be served, the network becomes *unbalanced*. Dall'Anese et al. permitted the benefits of SDP relaxation techniques [51] in an unbalanced setup [16] and explained why they expect the results of [47] to hold even in the unbalanced case. However, Dall'Anese's model is prone to numerical instability when the network size grows. Gan et al. indicted the nature of the model itself: in a bus injection model[1] (BIM) like the one of [16], similar variables are subtracted, leading to a bad numerical behavior. They proposed a novel and more stable approach: an SDP branch flow model (BFM). Then, they proved that BIM-SDP is exact if and only if BFM-SDP is exact, and empirically showed that BFM-SDP is numerically exact for several IEEE networks [52]. In light of this work, [53] applied this branch flow model on multiphase unbalanced radial networks. One should also mention that these SDP relaxations do not scale for extremely large systems [18], resulting in publications exploiting the sparsity of power systems [54, 55].

For a survey and more references to a growing literature on convex relaxations of OPF and their exactness, see [56, 57]. In contrast to preceding studies which aim to explore a provable exactness guarantee, [58] recently developed an algorithm to recover a global solution from an

[1]The reader can refer to chapter 3 for a definition of bus injection and branch flow model.

approximate solution offered by an inexact relaxation. Finally, one can wonder which relaxation to use among all these models. Comparisons can be found in [54, 56, 55, 42]. What stands out from these studies is the prohibitive amount of computational resource required by SDP relaxations to solve large size networks.

In this document, we adopt the BFM-SDP relaxation of [52] on multiphase unbalanced radial networks, as in [53] because of its satisfactory numerical behavior. We compare it with the classical SOCP relaxation of [34] and [6] on the equivalent single-phase network. Indeed, we will be working on a radial network, for which the relaxation is exact [6]. Furthermore, as mentioned above, it is a relevant benchmark regarding efficiency. At last, we also implement a linear program that will also be used for comparison purpose.

2.2 Distributed Algorithms for Microgrids

Most algorithms proposed in the literature are centralized and meant for applications in today's energy management systems that, e.g., centrally schedule a relatively small number of generators [40]. However, with the advent of renewable variable generation resources (e.g. photovoltaics) and new storage-capable loads (e.g. batteries for electric vehicles) in the distribution network (commonly referred as distributed energy resources, DERs), power systems will undergo radical transformations, as described in chapter 1. First, the numerous former passive consumers become producers-consumers, increasing significantly the size of the problem. Second, their production relies on the weather, introducing uncertainty that can notably be balanced by their energy storage device. Subsequently, a massive penetration of DERs renders an efficient centralized approach unthinkable due to the complexity of the resulting OPF and extensive amount of computational and communication resources that would be required.

In order to address that issue, researchers have focused on distributed algorithms. Dividing the huge main OPF into multiple scalable subinstances, they are able to benefit from parallelization and dispersed computational resources and hence to meet this modern challenge. The very first to propose a large-scale distributed algorithm for the OPF were Kim et al. in 1997 [59], but they were not considering the nonconvexity of the OPF. From there on, various decentralized algorithms have been developed. The SDP relaxation is solved by an efficient distributed algorithm in [48]. Building on these publications, Peng and Low applied ADMM to obtain a distributed algorithm on a radial network using (i) the classic SOCP relaxation for the balanced case [40] (ii) the SDP relaxation for the unbalanced case [53]. We will proceed similarly in this document, except that we will use SDDP instead of ADMM. Indeed, we will apply SDDP on the classic SOCP relaxation for the balanced case, before applying it on the SDP relaxation for the unbalanced case. A review about SDDP is provided in the next section.

2.3 Stochastic Dual Dynamic Programming

In 1991, Pereira and Pinto proposed the Stochastic Dual Dynamic Programming algorithm to solve multistage stochastic linear optimization problems [21]. By simulating the future state instead of enumerating all possibilities, they manage to avoid the well-known *curse of dimensionality* of dynamic programming. The algorithm can also be parallelized and they illustrated it in a case study of optimal stochastic scheduling for a 39-reservoir system. The next year, Rotting and Gjelsvik were already applying SDDP to the planning of the Norwegian hydro-dominated power system. Obtaining good results, they even proposed improving implementation details and concluded that the studied algorithm had a promising future in the stochastic scheduling of hydro-dominated systems [60]. Since 1993, Brazil has been managing its reservoirs using

SDDP-based algorithms to solve the large scale long term hydrothermal coordination problem for the national hydro system [61]. The SDDP algorithm has been extensively studied in the framework of hydrothermal scheduling [62, 63, 64, 65]. In fact, it is the principal tool used for the purpose of scheduling hydrothermal systems, namely planning the level of water in hydro reservoirs in order to meet the annual demand of a system at minimum expected cost [66].

[67] provides a novel convergence proof and [68] discusses statistical properties and convergence of the SDDP algorithm. SDDP has found a great commercial success and could be valuable for numerous other applications beyond hydrothermal scheduling. Recently, it has been applied to the real-time dispatch of storage under renewable supply uncertainty [8], opening a fertile ground for our work. As mentioned in section 1.3, SDDP has never been applied to nonlinear programs to the best of my knowledge. With the implementation of SDDP on SOCP and SDP relaxations, this thesis stands out of what has been explored by the literature so far.

Chapter 3

Models

In the present chapter, we address one of the most studied optimization problem in the literature, the Optimal Power Flow. The single-phase case (`OPF1`) is first derived before the more elaborate three-phase case (`OPF3`). The `OPF1` is simpler but seems ineffective at first sight since power systems are designed with three phases. Nonetheless, we show that in a balanced setup, an instance of the `OPF3` can be equivalently formulated as an instance of the `OPF1`.

This problem being highly nonconvex and NP-hard, we need to resort to approximations in order to solve it on practical power networks. Concerning `OPF1`, a rough linear approximation (`LP`) is first presented. Then we introduce a popular convex relaxation: the second-order cone program (`SOCP`). In the more realistic case of network imbalance, we develop the equally well-known semi-definite program (`SDP`). A summary of these models is presented on figure 3.1. Eventually, we will discuss the inclusion relation between the primal and dual domains of these formulations.

3.1 Optimal Power Flow

As in the literature, power networks will be represented using directed graphs throughout this document. Places where power is injected or withdrawn constitute nodes, also called buses. Connecting these buses, power lines form the edges of the graph. In this way, power can only be exchanged out of the network through the nodes of the graph, and is able to span over the

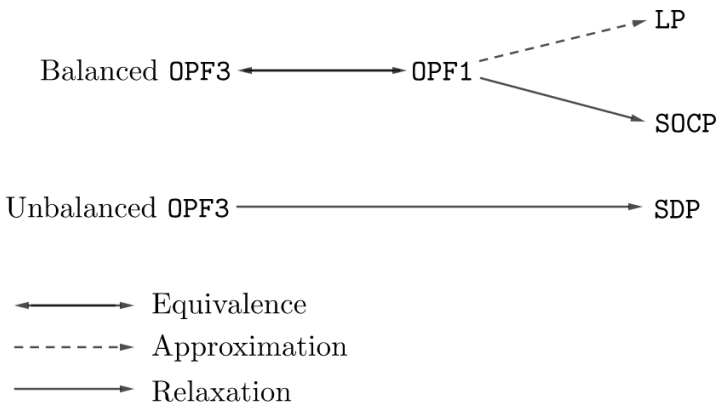


Figure 3.1: Relations between the problems used in this document. The reason why `LP` is not a relaxation is given in section 3.2.

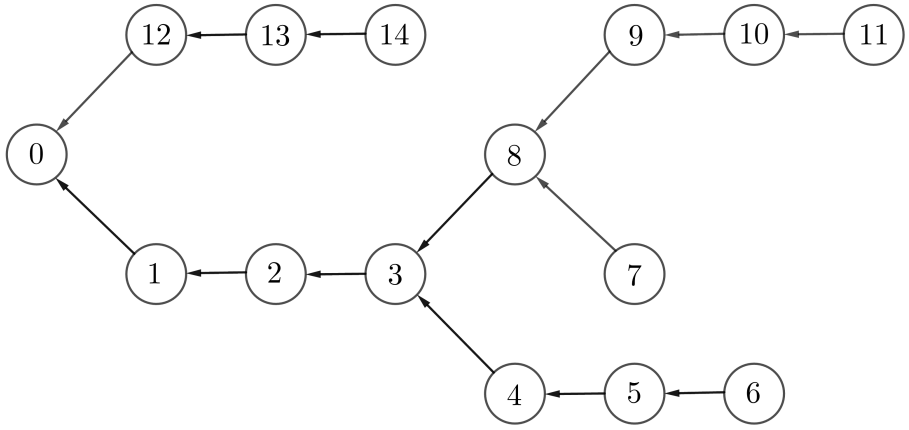


Figure 3.2: The graph representation of a power network.

nodes by using the edges. One can already highlight that the OPF can be cast as a network flow problem and foretell that power will have to be balanced at each bus. An example of a graph representation can be found in Figure 3.2.

In 1962, Carpentier pioneered power system research field by introducing the first formulation of the Optimal Power Flow [24]. His work extends the **Economic Dispatch** problem, which determines the optimal output of various generation facilities in order to satisfy loads at minimal cost. In addition, the OPF accounts for physical and technical constraints such as limits on voltage at the nodes and the maximal power that electrical lines can carry. More significantly, power cannot be routed arbitrarily over the network. The voltage, current and complex power are related by nonconvex physical rules, Kirchoff's laws, that must be respected in order to have a feasible flow.

As mentioned in chapter 2, two popular models of the OPF have been studied in the literature. [69] summarizes: the bus injection model (BIM) is the standard model for power flow analysis and optimization. It focuses on nodal variables such as voltages, current and power injections and does not directly deal with power flows on individual branches. Instead of nodal variables, the branch flow model (BFM) focuses on currents and powers on the branches. It has been used mainly for modeling distribution circuits which tend to be radial, but has received far less attention. Those two models and their various relaxations have been proven to be equivalent [69]. The BFM-SDP relaxation displaying a better numerical behavior than the BIM-SDP [52], we opt for a branch flow model in this document, not only for our SDP relaxation but also for the two other formulations. In this way, we adopt the same SOCP relaxation as Farivar et al. [6].

Throughout this document, we will adopt the 'per unit system' [23]: all physical quantities will be normalized about a nominal value. For example, in the low-voltage part of the network, one might use 220 V to normalize the voltage

$$V_{pu} = \frac{V}{220 \text{ V}},$$

such that has V_{pu} is dimensionless and is expressed as [p.u.] (per unit). Since this has the benefit of improving the numerical conditioning of computations [23], we will implicitly assume that all quantities are per unit normalized for all this document. Furthermore, steady state is assumed, so the dynamics of power flows are omitted.

In the present section, we briefly derive the Optimal Power Flow branch flow model. We first warm up on the single-phase case before addressing the three-phase case. Then, clarifications are presented on the equivalence between a balanced three-phase flow and a single-phase flow. For a more detailed approach, the interested reader can refer to [23, 1].

Type	Notation	Meaning
Sets	A_i	The ancestor of bus i
	C_i	The set of children of bus i
Variables	$S_i = P_i + \mathbf{i}Q_i$	Power flowing from bus i to its ancestor A_i
	$s_i = p_i + \mathbf{i}q_i$	Power injected at bus i
	V_i	Complex voltage at bus i
	I_i	Complex current flowing from bus i to its ancestor A_i
Parameters	$sD_i = pD_i + \mathbf{i}qD_i$	Power demanded at bus i
	$Z_i = R_i + \mathbf{i}X_i$	Impedance of the line connecting bus i to its ancestor A_i
	$Y_i = G_i - \mathbf{i}B_i = 1/Z_i$	Admittance of bus i
Math	z^*	The complex conjugate of z, i.e. $z^* = \Re(z) - \mathbf{i}\Im(z)$

Table 3.1: Notations used in power flow equations, illustrated in Figure 3.3.

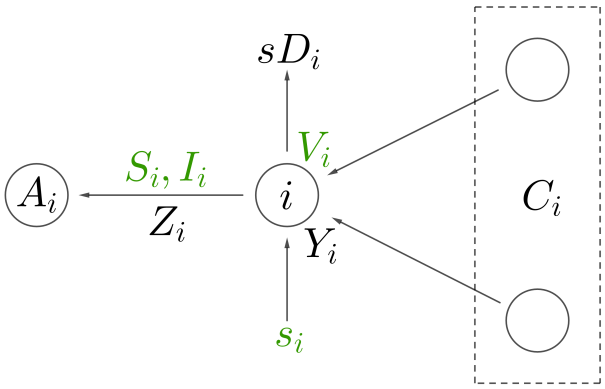


Figure 3.3: Illustration of the notations used in power flow equations, listed on table 3.1. Variables are written in green.

3.1.1 Single-Phase Flow

In order to derive the single-phase `OPF`, we proceed in two steps. First, we present the so-called *power flow equations*, the nonlinear set of equations governing the physical behavior of power. Then, we will refine this nonconvex skeleton with multiple time stages, uncertainty and operational constraints, leading us to the full mathematical formulation of the single-phase `OPF`.

Power Flow Equations

Defining the notations listed in table 3.1 and represented in Figure 3.3, one can write the power flow equations of bus i

$$V_i - V_{A_i} = Z_i I_i \tag{3.1}$$

$$S_i = V_i I_i^* \tag{3.2}$$

$$s_i + \sum_{j \in C_i} \left(S_j - Z_j |I_j|^2 \right) = S_i + sD_i + Y_i^* |V_i|^2. \tag{3.3}$$

This set of equations is also often referred as *Kirchhoff's Laws*. These relations are fundamental:

(3.1) is none other than the reknown *Ohm's Law*: it simply states that the current going through the line is proportional to the difference of voltage between its extremities. In addition, the impedance is constant, independent of the current. This elementary law was named after Georg Ohm's work in 1827 [70].

(3.2) merely defines the electrical energy transmitted by a current I_i at a voltage V_i.

(3.3) constitutes the customary *Power Balance*: the total amount of power entering bus i must compensate the power flowing out, i.e. power cannot accumulate. Power loss comes from

- the impedance of the line, and is proportional to $|I_j|^2$,
- the admittance of the bus, and is proportional to $|V_i|^2$.

One can highlight that if we neglect power losses, the well-known nonlinearity of the `OPF` solely comes from equation (3.2), the basic definition of power. This single quadratic equality constraint renders the problem nonconvex. [23] rightly points out that formulating the `OPF` using current and voltage instead of power would avoid this issue since equation (3.1) is linear. In this way, the resulting problem would be far more tractable, so why not do it ? The reason is that generators produce *power* and loads need *power*. We could not formulate the `OPF` in terms of voltage without mentioning and constraining the current, in this way effectively modeling the power. In the grave words of [23], *modeling power systems in terms of power is unfortunately an inescapable reality of power system engineering.*

Optimal Power Flow

Now that we have the nonconvex skeleton of our model, we can refine it with the particularities of our problem:

- Each bus is considered to own a battery. The current state of the battery depends on past decisions and will affect the future state, so our model needs to span over **multiple time stages**.
- We consider a high penetration of renewables, e.g. each house owns rooftop solar panels. In this way, each bus generates an uncertain amount of power. We subtract it from their demand and therefore model the demand as a **stochastic** process.

Subsequently, we end up with a multistage stochastic model. In order to define the `OPF` rigorously, we must mathematically model the uncertainty. To this end, we employ a common graphical tool in stochastic programming: a **Scenario Tree** [66]. Our model starts at time 1 that constitutes the root of the tree. Then, the realization of uncertainty at time t is denoted as ω_t and the set of values that ω_t can take is designated by Ω_t. But in order to account for the previous realizations, we have to introduce $\omega_{[t]} = (\omega_1, \dots, \omega_t)$ that represent the *history of realizations* until time t. In the scenario tree, the nodes represent the $\omega_{[t]}$'s and the edges the ω_t's. Denoting the set of values that $\omega_{[t]}$ can take by $\Omega_{[t]}$, we can write $\Omega_{[t]} = \Omega_1 \times \cdots \times \Omega_t$.

A small example of scenario tree is given in Figure 3.4 to illustrate these notations. We have

$$\Omega_1 = \{1\} \qquad \Omega_2 = \Omega_3 = \{1, 2\} \qquad \Omega_{[3]} = \Omega_1 \times \Omega_2 \times \Omega_3.$$

The probabilities are also given on the figure. Let us assume that a quantity h is defined at the third time stage. Its expectation is given by

$$\mathbb{E}\,[h] = P_1 \cdot \Big(P_{1,1} h(1,1,1) + P_{1,2} h(1,1,2)\Big) + P_2 \cdot \Big(P_{2,1} h(1,2,1) + P_{2,2} h(1,2,2)\Big).$$

Moreover, we must add thermal and technical constraints bounding the quantities in order to account for physical limits of the network. The final ingredient that upgrades our model to an optimization problem is the objective function. There are multiple relevant choices, as the total real power generation, the resistive power losses or the cost of real power generation. We will opt for the cost of real power generation since it is the most used in the industry. In addition, the

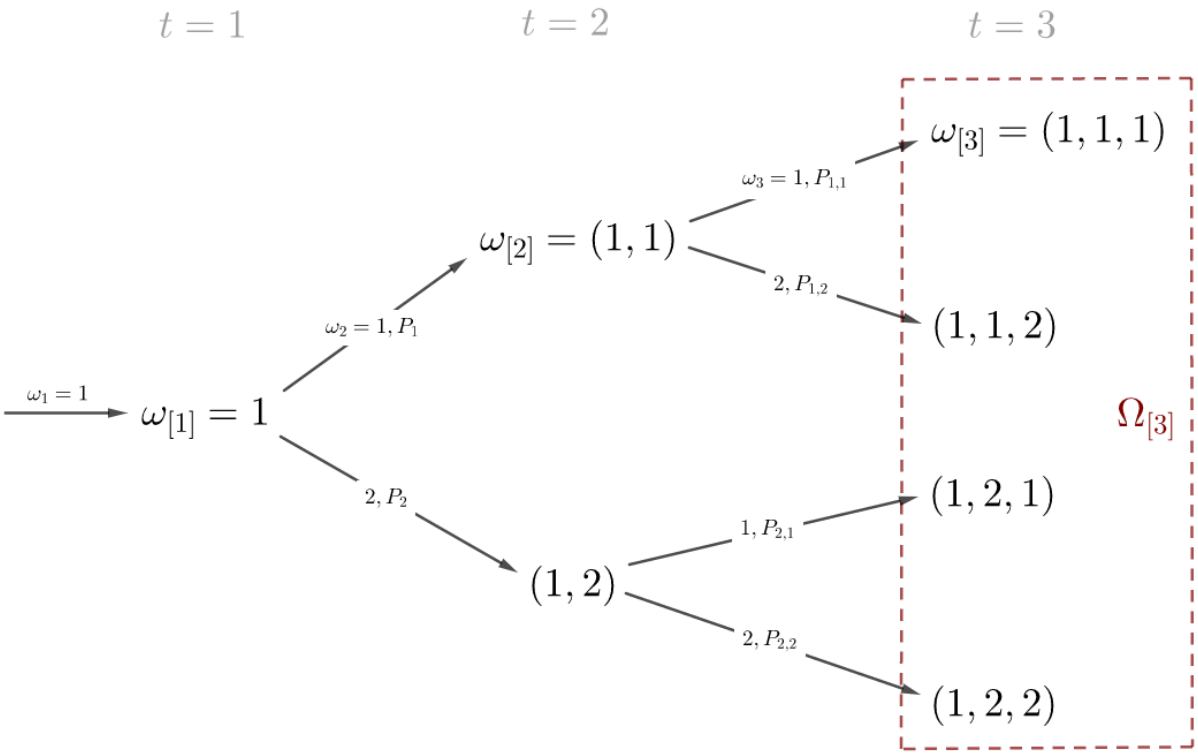


Figure 3.4: Example of scenario tree.

objective function must be increasing with current in order to ensure the tightness of the SOCP and SDP relaxations (cfr sections 3.3.3 and 3.4.2). Therefore, resistive power losses are added to the objective.

When solving the resulting problem, we obtain an infinite number of solutions with the same objective. This is due to the fact that all voltage angles can be shifted by any value and provide the same objective. As we are only interested in keeping one of them, we arbitrarily decide to fix the angle of a bus, called the *swing bus* [23], to zero. The voltage at this bus will also be used as a nominal value for normalizing all voltages, and in this document bus 0 will be the designated swing bus, so we end up with constraint (3.12).

Defining all necessary new notations listed in table 3.2 and illustrated in Figure 3.5, one can finally write the comprehensive single-phase OPF accounting for all particularities of the problem that we are addressing in this work.

$$\min_{S,s,I,V,x,bc,bd} \quad f(s,I) \qquad \text{(OPF1)}$$

$$\text{s.t.} \quad V_{i,t,\omega_{[t]}} - V_{A_i,t,\omega_{[t]}} = Z_i I_{i,t,\omega_{[t]}} \qquad i \in \mathcal{E}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \quad (3.4)$$

$$s_{i,t,\omega_{[t]}} + \sum_{j \in C_i} \left(S_{j,t,\omega_{[t]}} - Z_j \left| I_{j,t,\omega_{[t]}} \right|^2 \right) + bd_{i,t,\omega_{[t]}} = S_{i,t,\omega_{[t]}} + sD_{i,t,\omega_t} + Y_i^* \left| V_{i,t,\omega_{[t]}} \right|^2 + bc_{i,t,\omega_{[t]}}$$

$$i \in \mathcal{N}^+, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}, \omega_t \in \Omega_t \quad (3.5)$$

$$s_{0,t,\omega_{[t]}} + \sum_{j \in C_0} \left(S_{j,t,\omega_{[t]}} - Z_j \left| I_{j,t,\omega_{[t]}} \right|^2 \right) + bd_{0,t,\omega_{[t]}} = Y_0^* \left| V_{0,t,\omega_{[t]}} \right|^2 + bc_{0,t,\omega_{[t]}}$$

$$t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \quad (3.6)$$

$$S_{i,t,\omega_{[t]}} = V_{i,t,\omega_{[t]}} I^*_{i,t,\omega_{[t]}} \qquad i \in \mathcal{E}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \quad (3.7)$$

Type	Notation	Meaning
Sets	$\mathcal{N}$	The set of nodes, $\mathcal{N} = \{0, \dots, n\}$
	$\mathcal{N}^+$	The set of nodes except node 0, $\mathcal{N}^+ = \{1, \dots, n\}$
	$\mathcal{E}$	The set of edges. $\mathcal{E} = \mathcal{N}^+$
	$\mathcal{T}$	The set of time stages, $\mathcal{T} = \{1, \dots, H\}$
	Ω_t	The set of possible realizations of uncertainty at stage t
	$\Omega_{[t]}$	The set of possible histories of realizations of uncertainty at stage t
Variables	$x_{i,t,\omega_{[t]}}$	Real energy stored in the battery at bus i, time t, history $\omega_{[t]}$
	$bc_{i,t,\omega_{[t]}}$	Real power charged in the battery at bus i, time t, history $\omega_{[t]}$
	$bd_{i,t,\omega_{[t]}}$	Real power discharged from the battery at bus i, time t, history $\omega_{[t]}$
Parameters	$MC_{i,t}$	Marginal cost of power generation at bus i, time t
	PL	Power loss constant
	ηc_i	Battery charge efficiency at bus i
	ηd_i	Battery discharge efficiency at bus i
	$x_{\text{init},i}$	Initial energy stored in the battery at bus i, before time 1
	$p_{\max,i,t}$	Maximal real power generation at bus i, time t
	$q_{\max,i,t}$	Maximal reactive power generation at bus i, time t
	$v_{\min,i}$	Minimal squared voltage magnitude at bus i
	$v_{\max,i}$	Maximal squared voltage magnitude at bus i
	$x_{\max,i}$	Battery capacity at bus i
	$bc_{\max,i}$	Maximal battery charge rate at bus i
	$bd_{\max,i}$	Maximal battery discharge rate at bus i
Math	$[t]$	$\{1, \dots, t\}$

Table 3.2: Notations used in the single-phase `OPF`, illustrated in Figure 3.5.

$$x_{i,1} = x_{\text{init},i} + \eta c_i \cdot bc_{i,1} - \frac{bd_{i,1}}{\eta d_i} \qquad i \in \mathcal{N} \quad (3.8)$$

$$x_{i,t,\omega_{[t]}} = x_{i,t-1,\omega_{[t]}} + \eta c_i \cdot bc_{i,t,\omega_{[t]}} - \frac{bd_{i,t,\omega_{[t]}}}{\eta d_i} \qquad i \in \mathcal{N}, t = 2 \dots H, \omega_{[t]} \in \Omega_{[t]} \quad (3.9)$$

$$0 \le \Re(s_{i,t,\omega_{[t]}}) \le p_{\max,i,t} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \quad (3.10)$$

$$0 \le \Im(s_{i,t,\omega_{[t]}}) \le q_{\max,i,t} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$v_{\min,i} \le \left|V_{i,t,\omega_{[t]}}\right|^2 \le v_{\max,i} \qquad i \in \mathcal{N}^+, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \le x_{i,t,\omega_{[t]}} \le x_{\max,i} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \le bc_{i,t,\omega_{[t]}} \le bc_{\max,i} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \le bd_{i,t,\omega_{[t]}} \le bd_{\max,i} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \quad (3.11)$$

$$V_{0,t,\omega_{[t]}} = 1 \qquad t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}, \quad (3.12)$$

where

$$f(s, I) = \sum_{i \in \mathcal{N}} MC_{i,1} \cdot \Re(s_{i,1}) + PL \sum_{i \in \mathcal{E}} R_i \cdot |I_{i,1}|^2 + \mathbb{E}\left[\sum_{t=2\dots H} \sum_{i \in \mathcal{N}} MC_{i,t} \cdot \Re(s_{i,t,\omega_{[t]}}) + PL \sum_{t=2\dots H} \sum_{i \in \mathcal{E}} R_i \cdot \left|I_{i,t,\omega_{[t]}}\right|^2\right].$$

Some clarifications about `OPF1` are required:

All variables are defined for each history of realizations $\omega_{[t]}$, for every time t. Indeed, since the battery state x depends on past decisions, different solutions may be chosen for different past

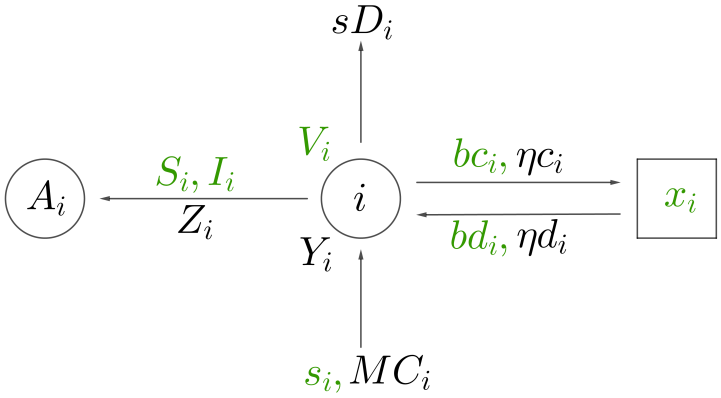


Figure 3.5: Illustration of the notations used in the single-phase `OPF`, listed on table 3.2. Variables are written in green.

decisions. Hence, each node of the scenario tree must have a different set of variables. Note that the only stochastic parameter is sD and only depends on the current realization of uncertainty ω_t, i.e. sD is a **Markov** stochastic process.

The objective to minimize is the expected total cost of generation, added to the total resistive power loss, weighted by a constant PL. The expectation is computed over all histories of realizations $\omega_{[t]}$, over all time stages except the first deterministic one.

(3.5) & (3.6) constitute the power balance. The charging and discharging of batteries are appended, but one should keep in mind that unlike the other quantities of the equation, they are real. By convention, there is no load and no edge leaving node 0 so its case needs to be handled separately.

(3.8) & (3.9) describe the battery state dynamics. Since $0 \leq \eta c_i, \eta d_i \leq 1$, one can check that the imperfect efficiency of the batteries induces power loss when charging and discharging. (3.9) is the only constraint that involves other time stages in the model, and is responsible for our multistage approach.

(3.10) – (3.11) ensure that the thermal and technical limits of the network are respected.

3.1.2 Three-Phase Flow

The single-phase `OPF` is one of the most frequently employed optimization routines in power systems [36] and has been an active research field since Carpentier's first formulation in 1962 [24]. The extensive amount of research led to numerous promising convex relaxations, including the second-order cone program that will be introduced in section 3.3 and studied in this document.

However, power is constituted by three phases in modern power systems. In the ideal balanced case depicted in Figure 3.6, we show that resorting to `OPF1` is equivalent in the next subsection. Nevertheless, in an unbalanced setup one cannot have recourse to this simplified single-phase problem. The three phases interact with each other, and these interactions are modeled by nonlinear relations as we will see hereafter. Furthermore, we are studying distribution networks, which are typically multiphase and unbalanced [17].

The three-phase `OPF` is more elaborate than `OPF1`, but the physical laws governing those two models are the same. Therefore, the equations look like each other and the derivation will be proceeded similarly as the single-phase case. We first present the power flow equations before refining the model with multiple time stages and uncertainty.

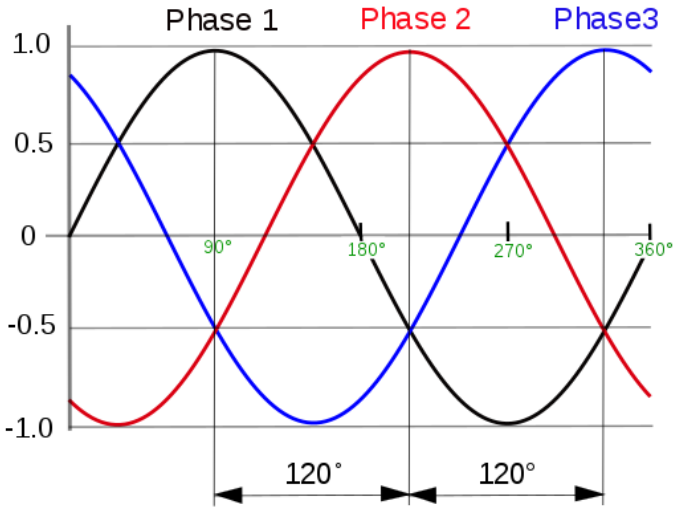


Figure 3.6: AC three-phase voltage, current or power in the ideal balanced case. Taken from [5].

Power Flow Equations

The three phases of the network will be denoted by a, b, c and indexed by $\phi \in \Phi_i$, $\Phi_i = \{a, b, c\}$ denoting the phases of bus $i \in \mathcal{N}$ or line $i \in \mathcal{E}$. Each variable and parameter is now provided for the three phases. For example, we denote by V_i^ϕ the complex voltage of bus i for phase $\phi \in \Phi_i$. For a more concise formulation, we define the column vector $\mathbf{v} = \begin{bmatrix} V_i^a & V_i^b & V_i^c \end{bmatrix}^T$. Throughout all this document, a lower boldface letter denotes a column vector and an upper boldface letter a matrix.

Defining the notations listed in table 3.3 and represented in Figure 3.7, one can write the power flow equations of bus i

$$\mathbf{v}_i - \mathbf{v}_{A_i} = \mathbf{Z}_i \boldsymbol{\ell}_i \tag{3.13}$$

$$\mathbf{S}_i = \mathbf{v}_i \boldsymbol{\ell}_i^H \tag{3.14}$$

$$\mathbf{s}_i + \sum_{j \in C_i} \operatorname{diag}\left(\mathbf{S}_j - \mathbf{Z}_j \boldsymbol{\ell}_j \boldsymbol{\ell}_j^H\right) = \mathbf{sD}_i + \operatorname{diag}\left(\mathbf{S}_i + \mathbf{v}_i \mathbf{v}_i^H \mathbf{Y}_i^H\right). \tag{3.15}$$

These relations are the matrix versions of the fundamental equations (3.1) – (3.3), which explains the striking similarity.

(3.13) simply states Ohm's law for each phase.

(3.14) defines the matrix $\mathbf{S}_i$, its diagonal elements denote the complex power flowing from bus i to its ancestor A_i. This equation is the first source of nonlinearity.

(3.15) constitutes the power balance. The mutual coupling between the phases stems from the phase impedance and admittance matrices. These matrices account for the electromagnetic effect of each line on the others, due to the electric and magnetic fields generated by the current flowing through the lines. In addition to coupling the three phases, power losses are the second source of nonlinearity to the model. At last, it can be noted that non-diagonal elements of $\mathbf{S}_i$, which have no physical interpretation, are not used in the equation, since we only take the diagonal of the matrix.

Type	Notation	Meaning
Variables	s_i^ϕ	Complex power injected at bus i for phase ϕ
	V_i^ϕ	Complex voltage at bus i for phase ϕ
	I_i^ϕ	Complex current flowing from bus i to its ancestor A_i for phase ϕ
	$\mathbf{s}_i$	$\begin{bmatrix} s_i^a & s_i^b & s_i^c \end{bmatrix}^T$
	$\mathbf{v}_i$	$\begin{bmatrix} V_i^a & V_i^b & V_i^c \end{bmatrix}^T$
	$\boldsymbol{\ell}_i$	$\begin{bmatrix} I_i^a & I_i^b & I_i^c \end{bmatrix}^T$
	$\mathbf{S}_i$	Complex matrix with diagonal elements denoting power flowing from bus i to its ancestor A_i, $\mathbf{S}_i \in \mathbb{C}^{\lvert\Phi_i\rvert\times\lvert\Phi_i\rvert}$. The off-diagonal elements do not have a physical interpretation
Parameters	sD_i^ϕ	Complex power demanded at bus i for phase ϕ
	$\mathbf{sD}_i$	$\begin{bmatrix} sD_i^a & sD_i^b & sD_i^c \end{bmatrix}^T$
	$\mathbf{Z}_i$	Phase impedance matrix of the line connecting bus i to its ancestor A_i, $\mathbf{Z}_i \in \mathbb{C}^{\lvert\Phi_i\rvert\times\lvert\Phi_i\rvert}$
	$\mathbf{Y}_i$	Phase admittance matrix of bus i, $\mathbf{Y}_i \in \mathbb{C}^{\lvert\Phi_i\rvert\times\lvert\Phi_i\rvert}$
Math	$\text{diag}(\mathbf{M})$	The column vector composed of the diagonal of matrix $\mathbf{M}$
	$\mathbf{M}^H$	The hermitian transpose of matrix $\mathbf{M}$, i.e. $\mathbf{M}^H = \mathbf{M}^{T*}$

Table 3.3: Notations used in power flow equations, illustrated in Figure 3.7.

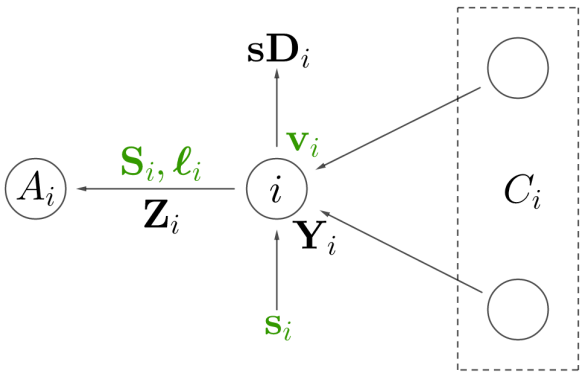


Figure 3.7: Illustration of the notations used in power flow equations, listed on table 3.3. Variables are written in green.

Optimal Power Flow

Similarly to the single-phase case, we now add time, uncertainty and technical constraints to the power flow equations. To this end, one must introduce the battery variables and bounding parameters for each phase. Since it is just the vectorial version of table 3.2 and figure 3.5, it is not redone here but can be found in the list of notations. This results in the comprehensive formulation of the three-phase OPF:

$$
\begin{aligned}
\min_{\mathbf{S},\mathbf{s},\boldsymbol{\ell},\mathbf{v},\mathbf{x},\mathbf{bc},\mathbf{bd}} \quad & f(\mathbf{s},\boldsymbol{\ell}) && && \text{(OPF3)}\\
\text{s.t.} \quad & \mathbf{v}_{i,t,\omega_{[t]}} - \mathbf{v}_{A_i,t,\omega_{[t]}} = \mathbf{Z}_i \boldsymbol{\ell}_{i,t,\omega_{[t]}} && && i\in\mathcal{E}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}\\
& \mathbf{s}_{i,t,\omega_{[t]}} + \sum_{j\in C_i} \operatorname{diag}\left(\mathbf{S}_{j,t,\omega_{[t]}} - \mathbf{Z}_j \boldsymbol{\ell}_{j,t,\omega_{[t]}} \boldsymbol{\ell}^H_{j,t,\omega_{[t]}}\right) + \mathbf{bd}_{i,t,\omega_{[t]}} && &&\\
& = \mathbf{sD}_{i,t,\omega_{[t]}} + \operatorname{diag}\left(\mathbf{S}_{i,t,\omega_{[t]}} + \mathbf{v}_{i,t,\omega_{[t]}} \mathbf{v}^H_{i,t,\omega_{[t]}} \mathbf{Y}_i^H\right) + \mathbf{bc}_{i,t,\omega_{[t]}} && &&\\
& && && i\in\mathcal{N}^+, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}, \omega_t\in\Omega_t\\
& \mathbf{s}_{0,t,\omega_{[t]}} + \sum_{j\in C_0} \operatorname{diag}\left(\mathbf{S}_{j,t,\omega_{[t]}} - \mathbf{Z}_j \boldsymbol{\ell}_{j,t,\omega_{[t]}} \boldsymbol{\ell}^H_{j,t,\omega_{[t]}}\right) + \mathbf{bd}_{0,t,\omega_{[t]}} && &&\\
& = \operatorname{diag}\left(\mathbf{v}_{0,t,\omega_{[t]}} \mathbf{v}^H_{0,t,\omega_{[t]}} \mathbf{Y}_0^H\right) + \mathbf{bc}_{0,t,\omega_{[t]}} && && t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}\\
& \mathbf{S}_{i,t,\omega_t} = \mathbf{v}_{i,t,\omega_t} \boldsymbol{\ell}^H_{i,t,\omega_t} && && i\in\mathcal{E}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}\\
& \mathbf{x}_{i,1} = \mathbf{x}_{\text{init},i} + \eta c_i \cdot \mathbf{bc}_{i,1} - \frac{\mathbf{bd}_{i,1}}{\eta d_i} && && i\in\mathcal{N}\\
& \mathbf{x}_{i,t,\omega_{[t]}} = \mathbf{x}_{i,t-1,\omega_{[t]}} + \eta c_i \cdot \mathbf{bc}_{i,t,\omega_{[t]}} - \frac{\mathbf{bd}_{i,t,\omega_{[t]}}}{\eta d_i} && && i\in\mathcal{N}, t=2\dots H, \omega_{[t]}\in\Omega_{[t]}\\
& 0 \le \Re\left(\mathbf{s}_{i,t,\omega_{[t]}}\right) \le \mathbf{p}_{\max,i,t} && && i\in\mathcal{N}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}\\
& 0 \le \Im\left(\mathbf{s}_{i,t,\omega_{[t]}}\right) \le \mathbf{q}_{\max,i,t} && && i\in\mathcal{N}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}\\
& \mathbf{v}_{\min,i} \le \operatorname{diag}\left(\mathbf{v}_{i,t,\omega_{[t]}} \mathbf{v}^H_{i,t,\omega_{[t]}}\right) \le \mathbf{v}_{\max,i} && && i\in\mathcal{N}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}\\
& 0 \le \mathbf{x}_{i,t,\omega_{[t]}} \le \mathbf{x}_{\max,i} && && i\in\mathcal{N}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}\\
& 0 \le \mathbf{bc}_{i,t,\omega_{[t]}} \le \mathbf{bc}_{\max,i} && && i\in\mathcal{N}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}\\
& 0 \le \mathbf{bd}_{i,t,\omega_{[t]}} \le \mathbf{bd}_{\max,i} && && i\in\mathcal{N}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}\\
& \mathbf{v}_{0,t,\omega_{[t]}} = \mathbf{v}_{\text{swing}} && && t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]},
\end{aligned}
$$

where

$$
\begin{aligned}
f(\mathbf{s},\boldsymbol{\ell}) = & \sum_{i\in\mathcal{N}} \sum_{\phi\in\Phi_i} MC^{\phi}_{i,1} \cdot \Re(s^{\phi}_{i,1}) + PL \sum_{i\in\mathcal{E}} \sum_{\phi\in\Phi_i} R_i^{\phi,\phi} \cdot \ell^{\phi}_{i,1} \ell^{\phi *}_{i,1}\\
& + \mathbb{E}\left[\sum_{t=2\dots H} \sum_{i\in\mathcal{N}} \sum_{\phi\in\Phi_i} MC^{\phi}_{i,t} \cdot \Re(s^{\phi}_{i,t,\omega_{[t]}}) + PL \sum_{t=2\dots H} \sum_{i\in\mathcal{E}} \sum_{\phi\in\Phi_i} R_i^{\phi,\phi} \cdot \ell^{\phi}_{i,t,\omega_{[t]}} \ell^{\phi *}_{i,t,\omega_{[t]}} \right],
\end{aligned}
$$

and $\mathbf{v}_{\text{swing}} = \begin{bmatrix} 1 & -\frac{1}{2} + \mathbf{i}\frac{\sqrt{3}}{2} & -\frac{1}{2} - \mathbf{i}\frac{\sqrt{3}}{2} \end{bmatrix}^T$. A swing bus must be chosen for the same reason as in the single-phase case. $V^a_{0,t,\omega_{[t]}}$ is arbitrarly set to 1, and we set the two other phases in order to have a balanced flow: same magnitude and a $2\pi/3$ phase shift.

3.1.3 The Equivalence Between a Balanced Three-Phase Flow and a Single-Phase Flow

In a balanced setup, the three phases have a perfect sine form, same magnitude and an exact $2\pi/3$ phase shift. So if we solve the `OPF` for one phase, we should be able to easily infer the two other ones by just shifting them by $2\pi/3$. However, we saw in equation (3.3) that the phases are mutually *coupled*:

An alternating current flowing through a conductor creates an alternating magnetic field around this conductor. Since the three power lines are set next to each other, each line is exposed to the magnetic field of the others. Moreover, Faraday's law of induction dictates that a change of the magnetic flux induces a voltage drop/increase across the line. In this way, each line undergoes voltage variations because of the magnetic fields of the two other lines. This phenomenon is known as *mutual inductance*.

The power losses $\mathbf{Z}_j \boldsymbol{\ell}_j \boldsymbol{\ell}_j^H$ and $\mathbf{v}_i \mathbf{v}_i^H \mathbf{Y}_i^H$ account for such electromagnetic effects connecting the phases with each other. More precisely, this mutual coupling is modeled by the non-diagonal elements of the phase impedance and admittance matrices (respectively $\mathbf{Z}$ and $\mathbf{Y}$).

In a balanced setup, conductors carrying the current are evenly spaced, so that the mutual inductances and other electromagnetic effects are the same. By the way, it is why the word *symmetric* is sometimes used instead of balanced. Since each signal is assumed to be perfectly regular, the electromagnetic effects underwent by one line from its neighbours can be modeled by a constant term. Added to the self impedance of this line, we obtain a special impedance that accounts for the mutual coupling without having to explicitly equate the different phases with each other.

This allows us to model a balanced three-phase power flow by a single-phase one, where Z and Y account for the non-diagonal elements of $\mathbf{Z}$ and $\mathbf{Y}$, in addition to their diagonal elements (which are nothing but the self impedance and admittance of the line). Eventually, $\mathbf{Z}$ and $\mathbf{Y}$ can be calculated by solving Maxwell's equations, but usually computational models are being used to approximate them [71, 72].

3.2 Linear Program

In this section, we present a first approximated formulation of the Optimal Power Flow. This approximation ignores significant aspects of the original `OPF1` and will mainly serve to assess the impact of Kirchhoff's laws on the solution. Indeed, this model is obtained by linearizing these nonlinear constraints. Moreover, since we are working on a radial network, the problem can be further simplified [35]. In this way, we are left with an ordinary network flow problem. Although trivial, this approximation has valuable aspects:

- `LP` is linear, and can thus be solved efficiently. Hence, it provides a challenging benchmark for the `SOCP` and `SDP`.
- `LP` is intuitive and the solution is easy to understand unlike the other formulations. The `SOCP` and `SDP` may return perplexing solutions due to Kirchhoff's laws, which are difficult to apprehend. In that case, the `LP` solution would help us to value the effect of them on the solution. Indeed, by comparing the `LP` solution with the conic solutions, we could be able to understand the consequences of Kirchhoff's laws, that transform the `LP` solution into the conic ones.

Since Kirchhoff's laws are ignored and the objective only involves the real power, voltage and current can merely be discarded from the model. Similarly, real and reactive power are fully

decoupled without these fundamental laws. Dismissing the reactive power might lead to an LP feasible solution whereas OPF1 is infeasible if the maximal amount of reactive power that can be injected is not sufficient to balance the reactive demand. However, we are not interested in such cases, hence reactive power is also discarded.

Subsequently, removing I, V, Q, q and the related constraints lead to the formulation of our Linear Program approximating the OPF1:

$$
\begin{aligned}
\min_{P,p,x,bc,bd} \quad & f(p) && && \text{(LP)}\\
\text{s.t.} \quad & p_{i,t,\omega_{[t]}} - pD_{i,t,\omega_t} + \sum_{j\in C_i} P_{j,t,\omega_{[t]}} - P_{i,t,\omega_{[t]}} + bd_{i,t,\omega_{[t]}} - bc_{i,t,\omega_{[t]}} = 0 && && \\
& && && i \in \mathcal{N}^+, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}, \omega_t \in \Omega_t\\
& p_{0,t,\omega_{[t]}} + \sum_{j\in C_0} P_{j,t,\omega_{[t]}} + bd_{0,t,\omega_{[t]}} - bc_{0,t,\omega_{[t]}} = 0 && && t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}\\
& x_{i,1} = x_{\text{init},i} + \eta c_i \cdot bc_{i,1} - \frac{bd_{i,1}}{\eta d_i} && && i \in \mathcal{N}\\
& x_{i,t,\omega_{[t]}} = x_{i,t-1,\omega_{[t]}} + \eta c_i \cdot bc_{i,t,\omega_{[t]}} - \frac{bd_{i,t,\omega_{[t]}}}{\eta d_i} && && i \in \mathcal{N}, t = 2 \dots H, \omega_{[t]} \in \Omega_{[t]}\\
& 0 \le p_{i,t,\omega_{[t]}} \le p_{\max,i,t} && && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}\\
& 0 \le x_{i,t,\omega_{[t]}} \le x_{\max,i} && && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}\\
& 0 \le bc_{i,t,\omega_{[t]}} \le bc_{\max,i} && && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}\\
& 0 \le bd_{i,t,\omega_{[t]}} \le bd_{\max,i} && && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]},
\end{aligned}
$$

where

$$
f(p) = \sum_{i\in\mathcal{N}} MC_{i,1} \cdot p_{i,1} + \mathbb{E}\left[\sum_{t=2\dots H}\sum_{i\in\mathcal{N}} MC_{i,t} \cdot p_{i,t,\omega_{[t]}}\right].
$$

One could mention that since power loss is neglected on the lines and reactive power is discarded, a solution of LP is not necessarily a solution of OPF1. Thus, LP cannot be called a relaxation of OPF1, but only an approximation of this problem.

3.3 Second-Order Cone Program

The second-order cone program is obtained after two relaxation steps as illustrated in Figure 3.8. The first one consists in dropping voltage and current angles from the model, resulting in the OPF1-ar. The latter is still nonconvex due to a quadratic equality constraint. Nevertheless, changing this equality into an inequality yields an SOCP: this constitutes the second relaxation step. This second step is proved to always be exact provided there are no upper bounds on real and reactive loads. The first one, however, can only be reversed if we can recover the voltage and current angles from an optimal solution of OPF1-ar. Fortunately, the condition ensuring this, called *the angle recovery condition*, is satisfied if the network is *radial*, i.e. its topological representation is a tree. In addition, most distribution systems are radial [52].

This section briefly performs those two relaxation steps in order to obtain the SOCP, before discussing the exactness of this model. For a more detailed approach, the interested reader can refer to [6].

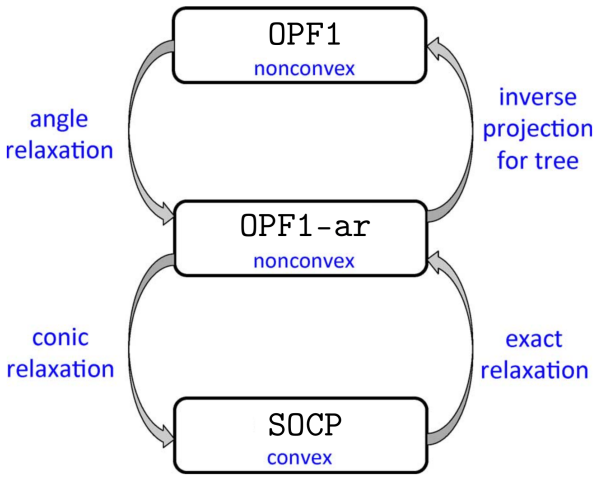


Figure 3.8: Relaxation and recovery steps for solving `OPF1` using the `SOCP`. Taken from [6].

3.3.1 Angle relaxation

Defining

$$v_i = |V_i|^2 \ \forall\ i \in \mathcal{N} \qquad\qquad \ell_i = |I_i|^2 \ \forall\ i \in \mathcal{E},$$

the objective of this relaxation is to formulate `OPF1` using v and ℓ instead of V and I. To this end, we first plug equation (3.2) into (3.1)

$$V_{A_i} = V_i - Z_i \frac{S_i^*}{V_i^*},$$

taking the squared magnitude of this reveals v and ℓ

$$v_{A_i} = v_i + |Z_i|^2 \ell_i - (Z_i S_i^* + Z_i^* S_i).$$

Expressing the latter equation in terms of real variables yields

$$v_{A_i} = v_i - 2(R_i P_i + X_i Q_i) + \left(R_i^2 + X_i^2\right) \ell_i, \tag{3.16}$$

similarly, equation (3.3) gives the following real equations

$$p_i + \sum_{j \in C_i} (P_j - R_j \ell_j) = P_i + pD_i + G_i v_i$$
$$q_i + \sum_{j \in C_i} (Q_j - X_j \ell_j) = Q_i + qD_i - B_i v_i.$$

Eventually, we take the magnitude squared of equation (3.2)

$$\ell_i = \frac{P_i^2 + Q_i^2}{v_i}, \tag{3.17}$$

leading to the `OPF-ar` formulation, obtained by replacing equations (3.4) – (3.7) by equations (3.16) – (3.17). The complete formulation of this model is presented in appendix A.1.

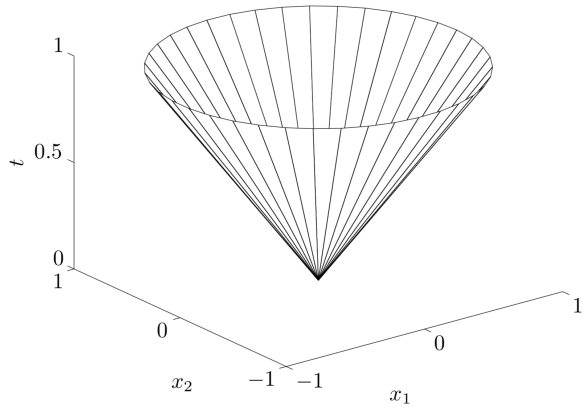


Figure 3.9: Boundary of $\mathbb{L}^2$ (t stands for x_0). Taken from [7].

3.3.2 Conic relaxation

The sole nonconvexity remaining in the `OPF1-ar` stems from the quadratic equality (3.17). However, one can note that this equation represents the border of a conic domain. From this observation comes the conic relaxation, which consists in 'filling' this conic set by relaxing it to an inequality

$$\ell_i \geq \frac{P_i^2 + Q_i^2}{v_i}. \tag{3.18}$$

Recalling that $(x+y)^2 - (x-y)^2 = 4xy$, we can give a more familiar look to this inequality

$$4P_i^2 + 4Q_i^2 + (\ell_i - v_i)^2 \leq (\ell_i + v_i)^2 \qquad \Leftrightarrow \qquad \left\| \begin{matrix} 2P_i \\ 2Q_i \\ \ell_i - v_i \end{matrix} \right\| \leq \ell_i + v_i, \tag{3.19}$$

which constraints these variables to stay inside a *second-order cone*.

Definition 1. The second-order cone, also called Lorentz cone (or 'ice cream cone'), is defined by the following convex set [73]

$$\mathbb{L}^n = \left\{ (x_0; \mathbf{x}) \in \mathbb{R}^{n+1} \middle| \ ||\mathbf{x}|| \leq x_0 \right\}.$$

An illustration is given in Figure 3.9

This second relaxation step consists in replacing the nonconvex quadratic equality (3.17) by the convex second-order cone inequality (3.19). Hence, the resulting problem aims at minimizing a linear objective function over the intersection of a polyhedron and a Lorentz cone. This constitutes a **Second-Order Cone Program** (`SOCP`) and can be solved efficiently by modern convex solvers.

$$\begin{aligned}
\min_{P,Q,p,q,\ell,v,x,bc,bd} \quad & f(p,\ell) && \text{(SOCP)} \\
\text{s.t.} \quad & v_{i,t,\omega_{[t]}} - 2 \cdot (R_i P_{i,t,\omega_{[t]}} + X_i Q_{i,t,\omega_{[t]}}) + (R_i^2 + X_i^2) \cdot \ell_{i,t,\omega_{[t]}} = v_{A_i,t,\omega_{[t]}} \\
& \qquad\qquad i \in \mathcal{E}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& p_{i,t,\omega_{[t]}} - pD_{i,t,\omega_t} + \sum_{j \in C_i} (P_{j,t,\omega_{[t]}} - R_j \ell_{j,t,\omega_{[t]}}) - P_{i,t,\omega_{[t]}} - G_i v_{i,t,\omega_{[t]}} + bd_{i,t,\omega_{[t]}} - bc_{i,t,\omega_{[t]}} = 0 \\
& \qquad\qquad i \in \mathcal{N}^+, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}, \omega_t \in \Omega_t
\end{aligned}$$

$$p_{0,t,\omega_{[t]}} + \sum_{j \in C_0} (P_{j,t,\omega_{[t]}} - R_j \ell_{j,t,\omega_{[t]}}) - G_0 v_{0,t,\omega_{[t]}} + bd_{0,t,\omega_{[t]}} - bc_{0,t,\omega_{[t]}} = 0$$
$$t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$q_{i,t,\omega_{[t]}} - qD_{i,t,\omega_t} + \sum_{j \in C_i} (Q_{j,t,\omega_{[t]}} - X_j \ell_{j,t,\omega_{[t]}}) - Q_{i,t,\omega_{[t]}} + B_i v_{i,t,\omega_{[t]}} = 0$$
$$i \in \mathcal{N}^+, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}, \omega_t \in \Omega_t$$

$$q_{0,t,\omega_{[t]}} + \sum_{j \in C_0} (Q_{j,t,\omega_{[t]}} - X_j \ell_{j,t,\omega_{[t]}}) + B_0 v_{0,t,\omega_{[t]}} = 0 \qquad t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$\left\| \begin{array}{c} 2P_{i,t,\omega_{[t]}} \\ 2Q_{i,t,\omega_{[t]}} \\ \ell_{i,t,\omega_{[t]}} - v_{i,t,\omega_{[t]}} \end{array} \right\| \leq \ell_{i,t,\omega_{[t]}} + v_{i,t,\omega_{[t]}} \qquad i \in \mathcal{E}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$x_{i,1} = x_{\text{init},i} + \eta c_i \cdot bc_{i,1} - \frac{bd_{i,1}}{\eta d_i} \qquad i \in \mathcal{N}$$

$$x_{i,t,\omega_{[t]}} = x_{i,t-1,\omega_{[t]}} + \eta c_i \cdot bc_{i,t,\omega_{[t]}} - \frac{bd_{i,t,\omega_{[t]}}}{\eta d_i} \qquad i \in \mathcal{N}, t = 2 \dots H, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \leq p_{i,t,\omega_{[t]}} \leq p_{\max,i,t} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \leq q_{i,t,\omega_{[t]}} \leq q_{\max,i,t} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \leq \ell_{i,t,\omega_{[t]}} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$v_{\min,i} \leq v_{i,t,\omega_{[t]}} \leq v_{\max,i} \qquad i \in \mathcal{N}^+, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \leq x_{i,t,\omega_{[t]}} \leq x_{\max,i} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \leq bc_{i,t,\omega_{[t]}} \leq bc_{\max,i} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \leq bd_{i,t,\omega_{[t]}} \leq bd_{\max,i} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$v_{0,t,\omega_{[t]}} = 1 \qquad t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]},$$

where

$$f(p,\ell) = \sum_{i \in \mathcal{N}} MC_{i,1} \cdot p_{i,1} + PL \sum_{i \in \mathcal{E}} R_i \cdot \ell_{i,1} + \mathbb{E}\left[\sum_{t=2\dots H} \sum_{i \in \mathcal{N}} MC_{i,t} \cdot p_{i,t,\omega_{[t]}} + PL \sum_{t=2\dots H} \sum_{i \in \mathcal{E}} R_i \cdot \ell_{i,t,\omega_{[t]}} \right].$$

Since the feasible set of `OPF1` is included inside the one of `OPF1-ar`, which is itself included in the one of `SOCP`, the optimal solution of the conic relaxation provides a lower bound on `OPF1-ar` and the initial problem. The next subsection addresses the tightness of these two relaxations: when are these sets the same, when are the optimal solutions equal ?

3.3.3 Exactness

First, the angle relaxation will always be tight in our case since we will exclusively work on radial networks in this document. Indeed, most distribution systems are radial [52] and the angle recovery condition is satisfied for radial networks, as mentioned above. The detailed angle recovery condition can be found in [6].

Concerning the conic relaxation, Farivar and Low proved the following theorem [6].

Theorem 1. Suppose that the cost function $f(p,\ell)$ is strictly increasing in ℓ, $pD_{i,t,\omega_{[t]}}$ and $qD_{i,t,\omega_{[t]}}$ are variables with no upper bound $\forall\ i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$. Then any optimal solution of `SOCP` is also optimal for `OPF1-ar`.

This theorem can be intuitively interpreted.

- First, since the objective is strictly increasing with current, we aim at minimizing ℓ, and therefore tightening constraint 3.18, which results in constraint 3.17.
- However, in case of oversupply (i.e. more power is injected than withdrawn), the solution might be tempted to get rid of this extra power by artificially creating resistive power losses, in order to satisfy the power balance and ensure feasibility. Nevertheless, this would result in a surge of current, rendering constraint 3.18 not tight and the solution not physical. To address that issue, Farivar and Low suppose no upper bound on the real and reactive loads. In this way, the solution has the possibility to give away surplus of power in any case.

In our case, the condition on the objective is fulfilled for any $PL > 0$. Although the second condition seems unrealistic [39], we could replace it by an infinite battery capacity $x_{\max,i}$ for all bus i. The conditions of this theorem are sufficient but not necessary, as the authors showed in simulations [6]. At last, the exactness of this relaxation will be further discussed on a practical test case in section 5.3.

3.4 Semi-Definite Program

In `OPF3`, nonlinearity stems from $\mathbf{v}_i\mathbf{v}_i^H$ and $\boldsymbol{\ell}_i\boldsymbol{\ell}_i^H$ terms, as well as the $\mathbf{S}_i = \mathbf{v}_i\boldsymbol{\ell}_i^H$ constraint. Similarly to the conic relaxation approach, we define

$$\mathbf{V}_i = \mathbf{v}_i\mathbf{v}_i^H \ \forall\ i \in \mathcal{N} \qquad\qquad \mathbf{I}_i = \boldsymbol{\ell}_i\boldsymbol{\ell}_i^H \ \forall\ i \in \mathcal{E}, \tag{3.20}$$

so that if we manage to formulate `OPF3` using $\mathbf{S}_i$, $\mathbf{V}_i$ and $\mathbf{I}_i$ instead of $\mathbf{v}_i$ and $\boldsymbol{\ell}_i$, we end up with a linear program. However, we will have to impose nonlinear constraints on $\mathbf{S}_i$, $\mathbf{V}_i$ and $\mathbf{I}_i$ in order to ensure that equations (3.14) and (3.20) are satisfied. These constraints consist in a semi-definite constraint and a highly nonconvex rank constraint. Therefore, relaxing the rank constraint results in a problem composed of a linear objective to minimize over the intersection of a linear domain and a semi-definite cone: a semi-definite program. Semi-definite programming lies in the esteemed field of convex programming. Over the last decade, `SDP` has been an active research field and solvers are becoming increasingly efficient [19]. It is thus tempting to perform this relaxation, but one must study if the rank constraint is satisfied to assess whether this relaxation is practical.

In the present section, we first perform de derivation of the `SDP` relaxation, before taking a closer look at its exactness.

3.4.1 Relaxation

After replacing $\mathbf{v}_i\mathbf{v}_i^H$ and $\boldsymbol{\ell}_i\boldsymbol{\ell}_i^H$ by $\mathbf{V}_i$ and $\mathbf{I}_i$ in equation (3.15), the only constraint involving $\mathbf{v}_i$ and $\boldsymbol{\ell}_i$ remaining is equation (3.13)

$$\mathbf{v}_{A_i} = \mathbf{v}_i - \mathbf{Z}_i\boldsymbol{\ell}_i.$$

In order to rectify this, we multiply both sides by their hermitian transposes to reveal $\mathbf{V}_i$ and $\mathbf{I}_i$ to obtain

$$\mathbf{V}_{A_i} = \mathbf{V}_i - \left(\mathbf{S}_i\mathbf{Z}_i^H + \mathbf{Z}_i\mathbf{S}_i^H\right) + \mathbf{Z}_i\mathbf{I}_i\mathbf{Z}_i^H,$$

which is a linear constraint. The resulting problem is linear, apart from the definitions of $\mathbf{S}_i$, $\mathbf{V}_i$ and $\mathbf{I}_i$

$$\mathbf{S}_i = \mathbf{v}_i\boldsymbol{\ell}_i^H \qquad\qquad \mathbf{V}_i = \mathbf{v}_i\mathbf{v}_i^H \qquad\qquad \mathbf{I}_i = \boldsymbol{\ell}_i\boldsymbol{\ell}_i^H. \tag{3.21}$$

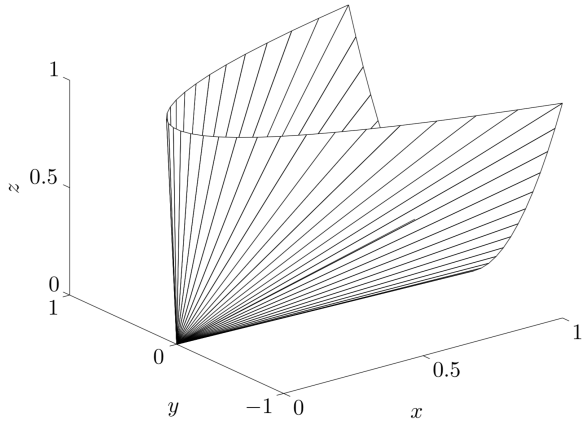


Figure 3.10: Boundary of $\mathbb{S}^2_+$ in the real case, with $[x\ y\ ;\ y\ z] \in \mathbb{S}^2_+$. Taken from [7].

One might consider the possibility of simply ignoring them, leading to a linear formulation of OPF3. A linear formulation of OPF3 would be a major breakthrough in power system scheduling, and would have a dramatic economic impact. However, ignoring these definitions would lead to non-physical solutions. It is equivalent to neglecting the fundamental relation between power, voltage and current. So these solutions would be impractical and we need to account for equations (3.21).

A first step towards the relaxation of these quadratic equality constraints is observing that equations (3.21) are equivalent to

$$\begin{bmatrix} \mathbf{V}_i & \mathbf{S}_i \\ \mathbf{S}_i^H & \mathbf{I}_i \end{bmatrix} = \begin{bmatrix} \mathbf{v}_i \\ \boldsymbol{\ell}_i \end{bmatrix} \begin{bmatrix} \mathbf{v}_i \\ \boldsymbol{\ell}_i \end{bmatrix}^H.$$

The key finding of this relaxation is then to realize that this equation is equivalent to

$$\begin{bmatrix} \mathbf{V}_i & \mathbf{S}_i \\ \mathbf{S}_i^H & \mathbf{I}_i \end{bmatrix} \in \mathbb{S}_+, \qquad \operatorname{rank} \begin{bmatrix} \mathbf{V}_i & \mathbf{S}_i \\ \mathbf{S}_i^H & \mathbf{I}_i \end{bmatrix} = 1, \tag{3.22}$$

a proof can be found in [52].

Definition 2. The hermitian positive semi-definite cone is defined by the following convex set [7]

$$\mathbb{S}^n_+ = \left\{ \mathbf{M} \in \mathbb{C}^{n\times n} \middle|\ \mathbf{M} = \mathbf{M}^H \text{ and } \mathbf{M} \succeq 0 \right\}.$$

The n superscript is sometimes omitted in this document. An illustration is given on figure 3.10.

The optimization problem obtained by replacing equations (3.21) by its equivalent formulation (3.22) is thus equivalent to OPF3. It is denoted as OPF3rank and its complete formulation can be found in appendix A.2. Relaxing the rank constraint from this problem yields the following **Semi-Definite Program** (SDP).

$$\begin{aligned}
\min_{\mathbf{S},\mathbf{s},\mathbf{I},\mathbf{V},\mathbf{x},\mathbf{bc},\mathbf{bd}} \quad & f(\mathbf{s},\mathbf{I}) && \text{(SDP)}\\
\text{s.t.} \quad & \mathbf{V}_{i,t,\omega_{[t]}} - \mathbf{S}_{i,t,\omega_{[t]}}\mathbf{Z}^H_{i,t,\omega_{[t]}} - \mathbf{Z}_{i,t,\omega_{[t]}}\mathbf{S}^H_{i,t,\omega_{[t]}} + \mathbf{Z}_{i,t,\omega_{[t]}}\mathbf{I}_{i,t,\omega_{[t]}}\mathbf{Z}^H_{i,t,\omega_{[t]}} = \mathbf{V}_{A_i,t,\omega_{[t]}} \\
& \qquad\qquad i \in \mathcal{E}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& \mathbf{s}_{i,t,\omega_{[t]}} + \sum_{j\in C_i} \operatorname{diag}\left(\mathbf{S}_{j,t,\omega_{[t]}} - \mathbf{Z}_j\mathbf{I}_{j,t,\omega_{[t]}}\right) + \mathbf{bd}_{i,t,\omega_{[t]}}
\end{aligned}$$

$$= \mathbf{sD}_{i,t,\omega_{[t]}} + \operatorname{diag}\left(\mathbf{S}_{i,t,\omega_{[t]}} + \mathbf{V}_{i,t,\omega_{[t]}}\mathbf{Y}_i^H\right) + \mathbf{bc}_{i,t,\omega_{[t]}}$$
$$i \in \mathcal{N}^+, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}, \omega_t \in \Omega_t$$

$$\mathbf{s}_{0,t,\omega_{[t]}} + \sum_{j \in C_0} \operatorname{diag}\left(\mathbf{S}_{j,t,\omega_{[t]}} - \mathbf{Z}_j\mathbf{I}_{j,t,\omega_{[t]}}\right) + \mathbf{bd}_{0,t,\omega_{[t]}} = \operatorname{diag}\left(\mathbf{V}_{0,t,\omega_{[t]}}\mathbf{Y}_0^H\right) + \mathbf{bc}_{0,t,\omega_{[t]}}$$
$$t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$\begin{bmatrix} \mathbf{V}_{i,t,\omega_{[t]}} & \mathbf{S}_{i,t,\omega_{[t]}} \\ \mathbf{S}_{i,t,\omega_{[t]}}^H & \mathbf{I}_{i,t,\omega_{[t]}} \end{bmatrix} \in \mathbb{S}_+ \qquad i \in \mathcal{E}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$\mathbf{x}_{i,1} = \mathbf{x}_{\text{init},i} + \eta c_i \cdot \mathbf{bc}_{i,1} - \frac{\mathbf{bd}_{i,1}}{\eta d_i} \qquad i \in \mathcal{N}$$

$$\mathbf{x}_{i,t,\omega_{[t]}} = \mathbf{x}_{i,t-1,\omega_{[t]}} + \eta c_i \cdot \mathbf{bc}_{i,t,\omega_{[t]}} - \frac{\mathbf{bd}_{i,t,\omega_{[t]}}}{\eta d_i} \qquad i \in \mathcal{N}, t = 2 \dots H, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \leq \Re\left(\mathbf{s}_{i,t,\omega_{[t]}}\right) \leq \mathbf{p}_{\max,i,t} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \leq \Im\left(\mathbf{s}_{i,t,\omega_{[t]}}\right) \leq \mathbf{q}_{\max,i,t} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$\mathbf{v}_{\min,i} \leq \operatorname{diag}\left(\mathbf{V}_{i,t,\omega_{[t]}}\right) \leq \mathbf{v}_{\max,i} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \leq \mathbf{x}_{i,t,\omega_{[t]}} \leq \mathbf{x}_{\max,i} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \leq \mathbf{bc}_{i,t,\omega_{[t]}} \leq \mathbf{bc}_{\max,i} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$0 \leq \mathbf{bd}_{i,t,\omega_{[t]}} \leq \mathbf{bd}_{\max,i} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}$$

$$\mathbf{V}_{0,t,\omega_{[t]}} = \mathbf{V}_{\text{swing}} \qquad t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]},$$

where

$$f(\mathbf{s}, \mathbf{I}) = \sum_{i \in \mathcal{N}} \sum_{\phi \in \Phi_i} MC_{i,1}^{\phi} \cdot \Re(s_{i,1}^{\phi}) + PL \sum_{i \in \mathcal{E}} \sum_{\phi \in \Phi_i} R_i^{\phi,\phi} \cdot \mathbf{I}_{i,1}^{\phi,\phi}$$
$$+ \mathbb{E}\left[\sum_{t=2\dots H} \sum_{i \in \mathcal{N}} \sum_{\phi \in \Phi_i} MC_{i,t}^{\phi} \cdot \Re(s_{i,t,\omega_{[t]}}^{\phi}) + PL \sum_{t=2\dots H} \sum_{i \in \mathcal{E}} \sum_{\phi \in \Phi_i} R_i^{\phi,\phi} \cdot \mathbf{I}_{i,t,\omega_{[t]}}^{\phi,\phi}\right],$$

and $\mathbf{V}_{\text{swing}} = \mathbf{v}_{\text{swing}} \cdot \mathbf{v}_{\text{swing}}^H = \begin{bmatrix} 1 & \frac{-1-\sqrt{3}}{2} & \frac{-1+\sqrt{3}}{2} \\ \frac{-1+\sqrt{3}}{2} & 1 & \frac{-1-\sqrt{3}}{2} \\ \frac{-1-\sqrt{3}}{2} & \frac{-1+\sqrt{3}}{2} & 1 \end{bmatrix}$.

3.4.2 Exactness

Clearly, if the optimal solution of `SDP` is such that each matrix $\mathbf{W}_{\text{opt}} = \begin{bmatrix} \mathbf{V}_{i,t,\omega_{[t]}} & \mathbf{S}_{i,t,\omega_{[t]}} \\ \mathbf{S}_{i,t,\omega_{[t]}}^H & \mathbf{I}_{i,t,\omega_{[t]}} \end{bmatrix}_{\text{opt}}$ has rank 1, then it is a *globally* optimal solution also for `OPF3`, since `SDP` with the rank constraint is equivalent to `OPF3`. Hence, there exist vectors $\mathbf{v}_{i,t,\omega_{[t]}}$, $\boldsymbol{\ell}_{i,t,\omega_{[t]}}$ such that

$$\mathbf{S}_{i,t,\omega_{[t]}} = \mathbf{v}_{i,t,\omega_{[t]}}\boldsymbol{\ell}_{i,t,\omega_{[t]}}^H \qquad \mathbf{V}_{i,t,\omega_{[t]}} = \mathbf{v}_{i,t,\omega_{[t]}}\mathbf{v}_{i,t,\omega_{[t]}}^H \qquad \mathbf{I}_{i,t,\omega_{[t]}} = \boldsymbol{\ell}_{i,t,\omega_{[t]}}\boldsymbol{\ell}_{i,t,\omega_{[t]}}^H \qquad \forall\ i,t,\omega_{[t]},$$

and the optimal costs of `OPF3` and `SDP` coincide at the optimum. Moreover, this global solution can be obtained via standard interior-point solvers in *polynomial time* [16].

However, $\mathbf{W}_{\text{opt}}$ could have rank greater than 1. In this case, we could resort to rank reduction techniques to approximate this matrix by a feasible rank-1 one, such as the method presented in [51]. But as the authors stress in [51], the resultant solution is generally suboptimal.

In [46], Lavaei et al. observe that the SDP relaxation is tight for many practical instances of balanced radial networks. Examining the geometry of the OPF, they realize that the power injection region can be modeled as a hollow ellipsoid. Relaxing the rank constraint consists in filling the ellipsoid (i.e. taking the convex hull), and the SDP relaxation is tight if the Pareto optimal front of the hollow and filled ellipsoids coincide. The authors formulate a practical sufficient condition for this, involving voltage angles and the impedance of the lines. This condition ensures the exactness of a convex relaxation on a radial network and is derived in a subsequent publication of the same authors [47].

In [48], the authors extended this work by taking into account limits on the reactive power injections, and considering tight voltage magnitude constraints instead of fixed values. Applying their work to our problem yields the following theorem.

Theorem 2. Consider a power distribution network with a tree topology and assume $f(\mathbf{s}, \mathbf{I})$ is strictly increasing with $\text{diag}(\mathbf{I})$. Let $\bar{\theta}_i$ be the smallest positive solution to the equation

$$p_{\max,i} = G_i + B_i \sin(\theta_i) - G_i \cos(\theta_i).$$

Suppose $\bar{\theta}_i$ satisfies

$$-\tan^{-1}\left(\frac{B_i}{G_i}\right) < \bar{\theta}_i < \tan^{-1}\left(\frac{B_i}{G_i}\right), \tag{3.23}$$

and reactive power demand upper bounds satisfy

$$\overline{qD}_i > \beta_i \quad \forall\ i, \tag{3.24}$$

with $\beta_i = -B_i + G_i \sin(\tilde{\theta}_i) + B_i \cos(\tilde{\theta}_i)$, where $\tilde{\theta}_i = \min(\tan^{-1}(G_i/B_i), \bar{\theta}_i)$. Let $\mathbf{W}_{\text{opt}}$ be an optimal solution to the SDP relaxation. Then

1. If $\mathbf{W}_{\text{opt}}$ is rank 1, then $\mathbf{W}_{\text{opt}} = \begin{bmatrix} \mathbf{v}_{\text{opt}} \\ \boldsymbol{\ell}_{\text{opt}} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{\text{opt}} \\ \boldsymbol{\ell}_{\text{opt}} \end{bmatrix}^H$ for some $\mathbf{v}_{\text{opt}}$ and $\boldsymbol{\ell}_{\text{opt}}$, and they are the optimal solution to OPF3.
2. If $\text{rank}(\mathbf{W}_{\text{opt}}) > 1$, then there is no feasible solution to OPF3.
3. If SDP is infeasible, then OPF3 is infeasible.

Condition (3.23) relies on the network structure, and using data from existing networks, authors have showed that it is expected to be satisfied by most networks. Indeed, we will observe that it is the case for the test network addressed in this work in section 5.3. Moreover, the assumption on the objective and condition (3.24) recall the SOCP exactness theorem and has similar consequences, as we will discuss in section 5.3.

Unfortunately, when the power network is unbalanced, these results no longer apply. However, Dall'Anese et al. still expect a rank-1 solution even in the unbalanced setup, and support their intuitive claim with two small meaningful examples [16]. This intuition will be tested on a practical example of unbalanced network in section 5.5.3.

3.5 Primal Domains

We have defined two versions of the optimal power flow and derived three different relaxations, as summarized on figure 3.11. It is easy to get confused with all these models and it would be convenient to have an unifying picture. Moreover, one can rightly wonder about the inclusion relation between their feasible domains. The present section tries to remedy this issue, and provides a geometrical insight of these five models. Since the single- and three-phase cases model two different problems, they are addressed separately.

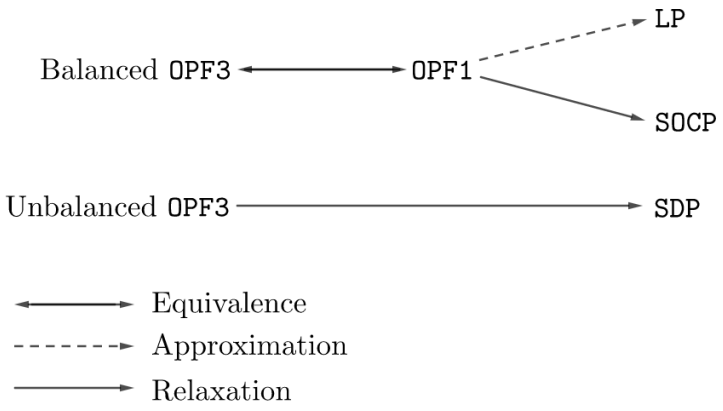


Figure 3.11: Relations between the problems used in this document.

3.5.1 Single-Phase Case

We first formally cover the relation between LP and SOCP before adding OPF1 in order to obtain a complete picture.

LP vs SOCP

In order to compare the LP and SOCP problems, we have to neglect the power loss on the lines due to impedance and admittance effects, i.e. the terms $-\sum_{j\in C_i} R_j \ell_{j,t,\omega_{[t]}}$ and $-G_i v_{i,t,\omega_{[t]}}$ in the power balance of the SOCP model. Indeed, since this effect is neglected in the LP model, one could not draw the following observations otherwise.

The inclusion relation between the primal domains of the LP and the SOCP problems seems trivial at first sight since the LP problem is a relaxation of the SOCP. However, the SOCP problem has more variables than the LP problem so we need to be careful and use a projection of the SOCP domain on the LP variable space. In order to formulate this formally, we define the following notations.

Definition 3.

- We will denote the total number of histories of scenarios by $\xi = \sum_{t=1}^{H} \left|\Omega_{[t]}\right|$.
- $\mathcal{S}_{\texttt{LP}} = \mathbb{R}^{(n-1)\xi(n\xi)^4}$ is the variable space of the LP problem. Indeed, $P \in \mathbb{R}^{(n-1)\xi}$ and $p, x, bc, bd \in \mathbb{R}^{n\xi}$ so $(P, p, x, bc, bd) \in \mathbb{R}^{(n-1)\xi(n\xi)^4}$.
- $\mathcal{S}_{\texttt{SOCP}} = \mathbb{R}^{((n-1)\xi)^3(n\xi)^6}$ is the variable space of the SOCP problem. Indeed, $P, Q, \ell \in \mathbb{R}^{(n-1)\xi}$ and $p, q, v, x, bc, bd \in \mathbb{R}^{n\xi}$.
- $P_{\texttt{SOCP}\to\texttt{LP}} : \mathcal{S}_{\texttt{SOCP}} \to \mathcal{S}_{\texttt{LP}} : P_{\texttt{SOCP}\to\texttt{LP}}(P, Q, p, q, l, v, x, bc, bd) = (P, p, x, bc, bd)$ is the projection from $\mathcal{S}_{\texttt{SOCP}}$ to $\mathcal{S}_{\texttt{LP}}$.
- $\mathcal{D}_{\texttt{LP}} \subset \mathcal{S}_{\texttt{LP}}$ is the feasible domain of the LP problem.
- $\mathcal{D}_{\texttt{SOCP}} \subset \mathcal{S}_{\texttt{SOCP}}$ is the feasible domain of the SOCP problem.

Those notations may appear cumbersome but will help us to formulate the following result concisely. Since the LP problem does not account for Ohm's Law, its domain is larger than the projection of the SOCP domain.

Proposition 1. $P_{\texttt{SOCP}\to\texttt{LP}}(\mathcal{D}_{\texttt{SOCP}}) \subseteq \mathcal{D}_{\texttt{LP}}$.

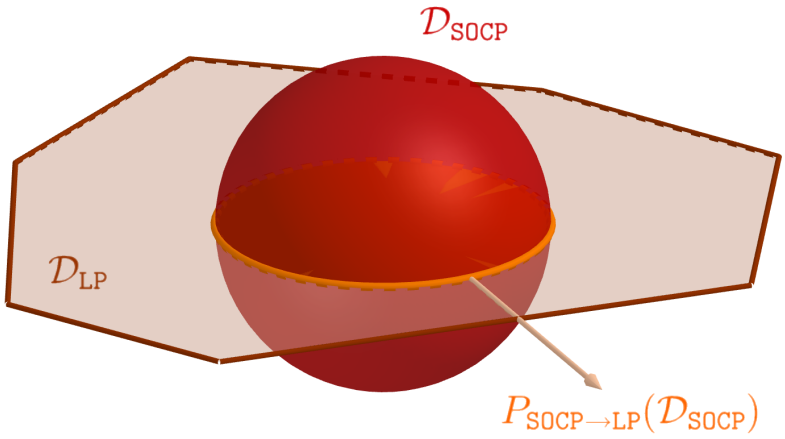


Figure 3.12: Schematic representation of Proposition 1: $P_{\texttt{SOCP}\to\texttt{LP}}(\mathcal{D}_{\texttt{SOCP}}) \subseteq \mathcal{D}_{\texttt{LP}}$.

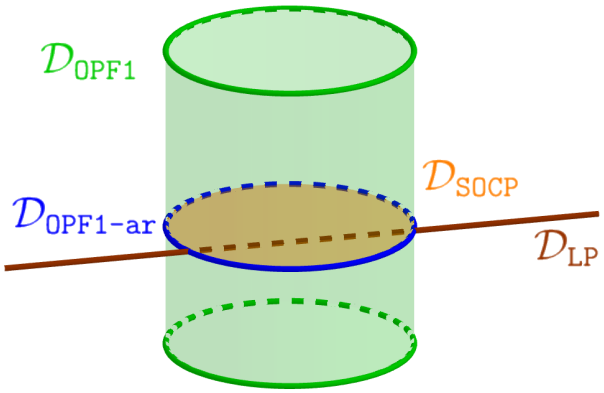


Figure 3.13: Primal feasible domains of `OPF1` and its relaxations.

Proof. Let us consider $(P, Q, p, q, l, v, x, bc, bd) \in \mathcal{D}_{\texttt{SOCP}}$. We verify that each constraint defining $\mathcal{D}_{\texttt{LP}}$ is satisfied by (P, p, x, bc, bd), so $(P, p, x, bc, bd) \in \mathcal{D}_{\texttt{LP}}$. □

This proposition is schematically represented on figure 3.12. As we will see in the next section, the inclusion relation will be contrary for the dual domains.

Adding `OPF1`

For the sake of conciseness, we proceed less formally for `OPF1`. `OPF1` has more variables than `SOCP` since it accounts for voltage and current angles. Its projection on $\mathcal{S}_{\texttt{SOCP}}$, named `OPF1-ar` can be interpreted as the contour of a conic domain as mentioned in section 3.3.2. `SOCP` is then obtained by 'filling' this contour. The unifying picture for the single-phase optimal power flow and its relaxations is presented on figure 3.13.

3.5.2 Three-Phase Case

In this case, we start with the nonconvex feasible domain of `OPF3`. Since its nonconvexity stems from quadratic equalities, we represent it as the contour of an ellipsoid. The `SDP` relaxation first performs an equivalent formulation of `OPF3` by replacing $\mathbf{v}_i$ and $\boldsymbol{\ell}_i$ by $\mathbf{V}_i$ and $\mathbf{I}_i$, resulting in `OPF3rank`. Since it lies in a different variable space, we tilted the ellipsoid, as if the system of axes was inclined. However, its projection on the initial space coincides with `OPF3` since these

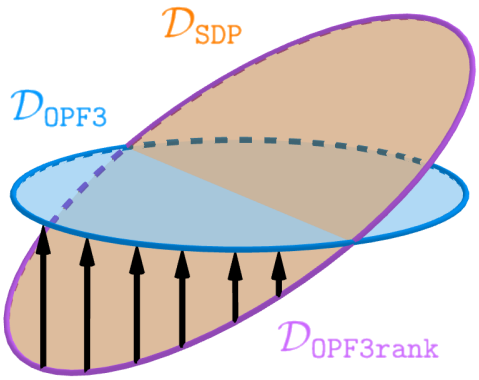


Figure 3.14: Primal feasible domains of `OPF3` and its relaxation.

formulations are equivalent. `OPF3rank` is nonconvex, but removing the rank constraint yields the convex `SDP`. Similarly to the single-phase case, we illustrated the removal of this constraint by filling the nonconvex domain. These feasible sets are depicted on figure 3.14.

[36] summarizes this clearly: *In applying this relaxation, we have lifted a nonconvex feasible set into a higher dimensional space. Intuitively, we can think of this as 'filling in' the nonconvexities.* The author just forgets to specify that the variable space of `OPF3` is not included in the variable space of `SDP`.

3.6 Dual Domains and Feasibility Cuts

It is now tempting to try to find the corresponding inclusion relation of the dual domains. This question is trickier since in order to answer it, we must derive the dual problems, which admit a large amount of variables and constraints. We will therefore perform these computations only for a very basic instance of `LP` and `SOCP` : the one-stage deterministic case. From the two basic duals obtained, we will be able to draw the inclusion relation between their dual domains. Then, we will suggest that this inclusion relation extends to the general multi-stage case with uncertainty.

Taking advantage of these dual computations, we will analytically derive a meaningful feasibility cut in the `LP` case, providing a physical interpretation of this concept. Finally, we will conclude with a geometrical primal-dual interpretation of primal constraints, providing a better understanding of the primal and dual inclusion relations.

3.6.1 Linear Dual

We are considering the deterministic one-stage case of the `LP` problem. In order to gain a meaningful intuition of the feasibility cuts, we relax the three upper-bounding constraints on the battery variables: the batteries have infinite capacity and we can charge and discharge as much as we want at each time stage.

$$
\begin{aligned}
\min_{P,p,x,bd,bc} \quad & \sum_{i\in\mathcal{N}} MC_i \cdot p_i \\
\text{s.t.} \quad & p_i + \sum_{j\in C_i} P_j - P_i + bd_i - bc_i = pD_i && (\sigma_{1,i}) \quad i \in \mathcal{N} \\
& x_i + \frac{bd_i}{\eta d_i} - \eta c_i \cdot bc_i = x_{\text{init},i} && (\sigma_{2,i}) \quad i \in \mathcal{N}
\end{aligned}
$$

$$
\begin{aligned}
& p_i \leq p_{\max,i} && (\sigma_{3,i}) \quad i \in \mathcal{N} \\
& p_i, x_{i,2}, bd_i, bc_i \geq 0 && \quad i \in \mathcal{N},
\end{aligned}
$$

where $pD_0 = P_0 = 0$ for ease of presentation. We rearranged the equations to have the constant terms at the right-hand-side. In order to compute the dual of this problem, we first formulate it in its equivalent matrix form. We define

$$
y^T = \begin{bmatrix} P^T & p^T & x^T & bd^T & bc^T \end{bmatrix}
$$

where without the subscript, we denote the corresponding column vector, e.g. $P^T = \begin{bmatrix} P_0 & \cdots & P_n \end{bmatrix}$. It yields

$$
\begin{aligned}
\min_y \quad & \begin{bmatrix} 0 & MC^T & 0 & 0 & 0 \end{bmatrix} \cdot y \\
\text{s.t.} \quad & \begin{bmatrix} C - I & I & 0 & I & -I \\ 0 & 0 & I & HD & -HC \end{bmatrix} \cdot y = \begin{bmatrix} pD \\ x_{\text{init}} \end{bmatrix} \\
& \begin{bmatrix} 0 & I & 0 & 0 & 0 \end{bmatrix} \cdot y \leq p_{\max,*} \\
& p, x, bd, bc \geq 0,
\end{aligned}
$$

where

$$
C_{ij} = \begin{cases} 1 \text{ if } j \in C_i \\ 0 \text{ otherwise} \end{cases} \qquad HD_{ij} = \begin{cases} \frac{1}{\eta d_i} \text{ if } i = j \\ 0 \text{ otherwise} \end{cases} \qquad HC_{ij} = \begin{cases} \eta c_i \text{ if } i = j \\ 0 \text{ otherwise.} \end{cases}
$$

Since the network is a tree, each node (except the root) is the child of one and only one node, so each non-root column of C is full of zero except one entry equal to 1. Applying the usual duality rules, we can write the dual

$$
\begin{aligned}
\max_\sigma \quad & \sigma^T \cdot \begin{bmatrix} pD \\ x_{\text{init}} \\ p_{\max} \end{bmatrix} \\
\text{s.t.} \quad & \sigma^T \cdot (C - I) = 0 \\
& \sigma^T \cdot \begin{bmatrix} I & 0 & I & -I \\ 0 & I & HD & -HC \\ I & 0 & 0 & 0 \end{bmatrix} \leq \begin{bmatrix} MC^T & 0 & 0 & 0 \end{bmatrix} \\
& \sigma_3 \leq 0.
\end{aligned}
$$

Performing the matrix multiplications gives a more readable formulation of this dual.

$$
\begin{aligned}
\max_\sigma \quad & \sum_{i \in \mathcal{N}} \sigma_{1,i} \cdot pD_i + \sigma_{2,i} \cdot x_{\text{init},i} + \sigma_{3,i} \cdot p_{\max,i} \\
\text{s.t.} \quad & \sigma_{1,i} = \sigma_{1,A_i} && \quad i \in \mathcal{N} \\
& \sigma_{1,i} + \sigma_{3,i} \leq MC_i && \quad i \in \mathcal{N} \\
& \sigma_{2,i} \leq 0 && \quad i \in \mathcal{N}
\end{aligned}
$$

$$\sigma_{1,i} + \frac{\sigma_{2,i}}{\eta d_i} \leq 0 \qquad i \in \mathcal{N}$$
$$-\sigma_{1,i} - \eta c_i \cdot \sigma_{2,i} \leq 0 \qquad i \in \mathcal{N}$$
$$\sigma_{3,i} \leq 0 \qquad i \in \mathcal{N}$$

From the first constraint and the connectivity of the network, we deduce that all $\sigma_{1,i}$ are equal to the same value, let us call it σ_1. Rearranging the equations gives

$$\begin{aligned}
(\texttt{LP Dual}) \quad \max_{\sigma} \quad & \sum_{i \in \mathcal{N}} \sigma_1 \cdot pD_i + \sigma_{2,i} \cdot x_{\text{init},i} + \sigma_{3,i} \cdot p_{\max,i} && && (3.25)\\
\text{s.t.} \quad & \sigma_1 + \sigma_{3,i} \leq MC_i && i \in \mathcal{N} && (3.26)\\
& \sigma_{2,i} \leq -\eta d_i \cdot \sigma_1 && i \in \mathcal{N} && (3.27)\\
& \sigma_{2,i} \geq -\frac{\sigma_1}{\eta c_i} && i \in \mathcal{N} && (3.28)\\
& \sigma_{2,i}, \sigma_{3,i} \leq 0 && i \in \mathcal{N}. && (3.29)
\end{aligned}$$

From this dual, we can now derive a feasibility cut to the primal problem.

The feasibility cuts are used in multi-stage algorithms to impose on the previous stage the feasibility of the current stage. Concerning our problem, for the three different models, only one variable has an impact on the next time stages : the battery state x. Thus, the feasibility cuts will only apply on $x_{i,t-1}$. In this section we are considering the one-stage case, but x_{init} stands for x_{t-1}.

The infeasibility of the primal translates into unboundedness of the dual: we need to find an extreme ray of the dual in order to derive the feasibility cut. The demand pD_i, the power stored $x_{\text{init},i}$ and the maximal production $p_{\max,i}$ are non-negative and constraint (3.29) bounds $\sigma_{2,i}$ and $\sigma_{3,i}$ to 0. The only way to have an unbounded objective is thus an infinite value of σ_1. We derive hereafter the conditions on the constants that would lead to such situation.

We begin with constraint (3.26). At first sight, it seems that this constraint bounds σ_1 to a finite value. It is indeed the case if $pD_i \leq p_{\max,i}\ \forall\ i \in \mathcal{N}$: we have

$$\sigma_1 = \min_{j \in \mathcal{N}} MC_j \qquad\qquad \sigma_{3,i} = 0\ \forall\ i \in \mathcal{N}.$$

But let us consider the case $\sum_{i \in \mathcal{N}} pD_i \geq \sum_{i \in \mathcal{N}'} p_{\max,i}$, where $\mathcal{N}'$ is the set of nodes with low marginal cost $\mathcal{N}' = \{i \in \mathcal{N} | MC_i < \min_{j \in \mathcal{N}} MC_j + 1\}$. We will prefer

$$\sigma_1 = \min_{j \in \mathcal{N}} MC_j + 1 \qquad \sigma_{3,i} = -1\ \forall\ i \in \mathcal{N}' \qquad \sigma_{3,i} = 0\ \forall\ i \in \mathcal{N} \setminus \mathcal{N}'.$$

to the previous choice, since the objective value is increased by $\sum_{i \in \mathcal{N}} pD_i - \sum_{i \in \mathcal{N}'} p_{\max,i}$. The marginal costs can be different but σ_1 is the same for each node. Thus constraint (3.26) is not necessarily tight for all nodes, which is why $\sigma_{3,i}$ can stay at 0 for some of them. However, since the marginal costs are finite, if we want to increase σ_1 to infinity, at some point the constraint will be tight for all nodes, and the objective will be increased by $\sum_{i \in \mathcal{N}} pD_i - \sum_{i \in \mathcal{N}} p_{\max,i}$ per increment.

So far we have totally ignored $\sigma_{2,i}$, we can now consider constraint (3.27). Since $x_{\text{init},i}$ is non-negative, this constraint will always be tight for every node. Thus, if σ_1 is increased by 1 unit, $\sigma_{2,i}$ is decreased by ηd_i for each node, causing a decrease of $\sum_{i \in \mathcal{N}} \eta d_i \cdot x_{\text{init},i}$ on the objective.

Putting it all together, increasing σ_1 to infinity will be strictly favorable if the increase on the objective is strictly larger than the decrease, i.e.

$$\begin{aligned} &\sum_{i\in\mathcal{N}} pD_i - \sum_{i\in\mathcal{N}} p_{\max,i} > \sum_{i\in\mathcal{N}} \eta d_i \cdot x_{\text{init},i} \\ \Leftrightarrow &\sum_{i\in\mathcal{N}} pD_i > \sum_{i\in\mathcal{N}} p_{\max,i} + \sum_{i\in\mathcal{N}} \eta d_i \cdot x_{\text{init},i}. \end{aligned}$$

This equation carries a very meaningful physical interpretation. The dual is unbounded – which is equivalent to the infeasibility of the primal – if the total demand on the network is larger than the sum of the maximal production possible and all the stored power one can discharge at its imperfect efficiency. Indeed, under this condition it is impossible to satisfy the demand since the available resources are not enough.

Thus, the feasibility cut of this problem is the exact complement of the above equation. Feasibility is assured if the total demand is lower or equal to the total available resources

$$\sum_{i\in\mathcal{N}} pD_i \leq \sum_{i\in\mathcal{N}} p_{\max,i} + \sum_{i\in\mathcal{N}} \eta d_i \cdot x_{\text{init},i},$$

which corresponds to the extreme ray

$$\sigma_{1,i} = 1 \qquad \sigma_{2,i} = -\eta d_i \qquad \sigma_{3,i} = -1 \qquad \forall\, i \in \mathcal{N}.$$

We also know that the dual multipliers σ can be interpreted as the marginal impact of the right-hand-side constants on the objective value. If we multiply the extreme ray by $MC_{\min} \triangleq \min_{j\in\mathcal{N}} MC_j$, we obtain

$$\sigma_{1,i} = MC_{\min} \qquad \sigma_{2,i} = -\eta d_i \cdot MC_{\min} \qquad \sigma_{3,i} = -MC_{\min} \qquad \forall\, i \in \mathcal{N}$$

For example, if pD_i is increased of 1 unit, the objective value will be increased of $\sigma_{1,i}$ units. If 1 more unit of power is needed to satisfy the demand, one need to produce it at a cost of $MC_{\min}$, thus $\sigma_{1,i} = MC_{\min}$. On the other hand, if $x_{\text{init},i}$ is increased of 1 unit, we will be able to discharge it at the efficiency ηd_i, saving $\sigma_{3,i} = \eta d_i \cdot MC_{\min}$ unit of power.

3.6.2 Conic Dual

Proceeding similarly as the previous section, we consider the deterministic one-stage case of the `SOCP` problem, the three upper-bounding constraints on the battery variables being relaxed. In addition, we also relax the lower and upper bounds on the voltage and drop de prescribed value of v_0.

$$\begin{aligned} \min_{P,Q,p,q,l,v,x,bd,bc} \quad & \sum_{i\in\mathcal{N}} MC_i \cdot p_i \\ \text{s.t.} \quad & v_i - v_{A_i} - 2\cdot(R_i P_i + X_i Q_i) + (R_i^2 + X_i^2)\cdot \ell_i = 0 && i \in \mathcal{N} \\ & p_i + \sum_{j\in C_i} (P_j - R_j \ell_j) - P_i + bd_i - bc_i = pD_i && i \in \mathcal{N} \end{aligned}$$

$$
\begin{aligned}
& q_i + \sum_{j \in C_i} (Q_j - X_j \ell_j) - Q_i = qD_i && i \in \mathcal{N} \\
& x_i + \frac{bd_i}{\eta d_i} - \eta c_i \cdot bc_i = x_{\text{init},i} && i \in \mathcal{N} \\
& \left\| \begin{matrix} 2P_i \\ 2Q_i \\ \ell_i - v_i \end{matrix} \right\| \leq \ell_i + v_i && i \in \mathcal{N} \\
& p_i \leq p_{\max,i} && i \in \mathcal{N} \\
& q_i \leq q_{\max,i} && i \in \mathcal{N} \\
& p_i, q_i, \ell_i, v_i, x_i, bc_i, bd_i \geq 0 && i \in \mathcal{N}
\end{aligned}
$$

where $pD_0 = qD_0 = P_0 = Q_0 = \ell_0 = 0$ for ease of presentation. We rearranged the equations to have the constant terms at the RHS. In order to dualize properly the conic constraint, we introduce the following change of variables

$$
\begin{array}{lllll}
\rho_i = 2P_i & \phi_i = 2Q_i & \alpha_i = \ell_i + v_i & \beta_i = \ell_i - v_i & \forall\ i \in \mathcal{N} \\
\Rightarrow P_i = \dfrac{\rho_i}{2} & Q_i = \dfrac{\phi_i}{2} & \ell_i = \dfrac{\alpha_i + \beta_i}{2} & v_i = \dfrac{\alpha_i - \beta_i}{2} & \forall\ i \in \mathcal{N},
\end{array}
$$

which gives us, after some rearrangements

$$
\begin{aligned}
\min_{P,Q,p,q,l,v,x,bd,bc} \quad & \sum_{i \in \mathcal{N}} MC_i \cdot p_i \\
\text{s.t.} \quad & \frac{R_i^2 + X_i^2 + 1}{2} \cdot \alpha_i + \frac{R_i^2 + X_i^2 - 1}{2} \cdot \beta_i - \frac{\alpha_{A_i} - \beta_{A_i}}{2} - R_i \rho_i - X_i \phi_i = 0 && (\sigma_{1,i}) \quad i \in \mathcal{N} \\
& p_i + \sum_{j \in C_i} \left(\frac{\rho_j}{2} - R_j \frac{\alpha_j + \beta_j}{2} \right) - \frac{\rho_i}{2} + bd_i - bc_i = pD_i && (\sigma_{2,i}) \quad i \in \mathcal{N} \\
& q_i + \sum_{j \in C_i} \left(\frac{\phi_j}{2} - X_j \frac{\alpha_j + \beta_j}{2} \right) - \frac{\phi_i}{2} = qD_i && (\sigma_{3,i}) \quad i \in \mathcal{N} \\
& x_i + \frac{bd_i}{\eta d_i} - \eta c_i \cdot bc_i = x_{\text{init},i} && (\sigma_{4,i}) \quad i \in \mathcal{N} \\
& p_i \leq p_{\max,i} && (\sigma_{5,i}) \quad i \in \mathcal{N} \\
& q_i \leq q_{\max,i} && (\sigma_{6,i}) \quad i \in \mathcal{N} \\
& \begin{bmatrix} \alpha_i \\ \rho_i \\ \phi_i \\ \beta_i \end{bmatrix} \in \mathbb{L}^3 && i \in \mathcal{N} \\
& p_i, q_i, x_i, bc_i, bd_i \geq 0 && i \in \mathcal{N}.
\end{aligned}
$$

Defining

$$
y^T = \begin{bmatrix} \alpha^T & \beta^T & \rho^T & \phi^T & p^T & q^T & x^T & bd^T & bc^T \end{bmatrix}
$$

yields the equivalent matrix form

$$
\begin{aligned}
V(x_{\text{init}}) = \min_{y} \quad & \begin{bmatrix} 0 & 0 & 0 & 0 & MC^T & 0 & 0 & 0 & 0 \end{bmatrix} \cdot y \\
\text{s.t.} \quad & \begin{bmatrix} \frac{1}{2}(\Gamma - C^T) & \frac{1}{2}(B + C^T) & -R & -X & & & & & \\ -\frac{1}{2}CR & -\frac{1}{2}CR & \frac{1}{2}(C-I) & 0 & I & 0 & 0 & I & -I \\ -\frac{1}{2}CX & -\frac{1}{2}CX & 0 & \frac{1}{2}(C-I) & 0 & I & 0 & 0 & 0 \\ & & & & & & I & HD & -HC \end{bmatrix} \cdot y = \begin{bmatrix} 0 \\ pD \\ qD \\ x_{\text{init}} \end{bmatrix} \\
& \begin{bmatrix} 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \end{bmatrix} \cdot y \leq \begin{bmatrix} p_{\max} \\ q_{\max} \end{bmatrix} \\
& y \in \mathbb{L}^{3n} \times \mathbb{R}^{5n}_+,
\end{aligned}
$$

where

$$
\Gamma_{ij} = \begin{cases} R_i^2 + X_i^2 + 1 \text{ if } i = j \\ 0 \text{ otherwise} \end{cases} \qquad B_{ij} = \begin{cases} R_i^2 + X_i^2 - 1 \text{ if } i = j \\ 0 \text{ otherwise} \end{cases}
$$

$$
R_{ij} = \begin{cases} R_i \text{ if } i = j \\ 0 \text{ otherwise} \end{cases} \qquad X_{ij} = \begin{cases} X_i \text{ if } i = j \\ 0 \text{ otherwise.} \end{cases}
$$

Let us recall the conic duality pair

$$
\begin{array}{lll} \min\limits_{y} & c^T y & \\ \text{s.t.} & Ay = b & \\ & y \succeq_K 0 & \end{array} \qquad \longleftrightarrow \qquad \begin{array}{ll} \max\limits_{\sigma} & b^T \sigma \\ \text{s.t.} & A^T \sigma \preceq_{K^*} c \end{array}
$$

where K^* is the dual cone of K. In our case, $K = \mathbb{L}^{3n} \times \mathbb{R}^{5n}_+$. Using the following properties

$$
(K_1 \times K_2)^* = K_1^* \times K_2^* \ \forall\ K_1, K_2 \qquad (\mathbb{R}^n_+)^* = \mathbb{R}^n_+ \ \forall\ n \qquad (\mathbb{L}^n)^* = \mathbb{L}^n \ \forall\ n,
$$

we deduce that our cone is self-dual, i.e. $(\mathbb{L}^{3n} \times \mathbb{R}^{5n}_+)^* = \mathbb{L}^{3n} \times \mathbb{R}^{5n}_+$. We can now write the dual

$$
\max_{\sigma} \quad \sum_{i \in \mathcal{N}} \sigma_{2,i} \cdot pD_i + \sigma_{3,i} \cdot qD_i + \sigma_{4,i} \cdot x_{\text{init},i} + \sigma_{5,i} \cdot p_{\max,i} + \sigma_{6,i} \cdot q_{\max,i} \tag{3.30}
$$

$$
\text{s.t.} \quad \begin{bmatrix} \frac{1}{2}(C^T - \Gamma) & -\frac{1}{2}(C^T + B) & R & X \\ \frac{1}{2}CR & \frac{1}{2}CR & \frac{1}{2}(I - C) & 0 \\ \frac{1}{2}CX & \frac{1}{2}CX & 0 & \frac{1}{2}(I - C) \end{bmatrix}^T \cdot \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} \in \mathbb{L}^{3n} \tag{3.31}
$$

$$
\sigma_{2,i} + \sigma_{5,i} \leq MC_i \qquad i \in \mathcal{N} \tag{3.32}
$$

$$
\sigma_{3,i} + \sigma_{6,i} \leq 0 \qquad i \in \mathcal{N} \tag{3.33}
$$

$$
\sigma_{4,i} \leq 0 \qquad i \in \mathcal{N} \tag{3.34}
$$

$$
\sigma_{2,i} + \frac{\sigma_{4,i}}{\eta d_i} \leq 0 \qquad i \in \mathcal{N} \tag{3.35}
$$

$$
-\sigma_{2,i} - \eta c_i \cdot \sigma_{4,i} \leq 0 \qquad i \in \mathcal{N} \tag{3.36}
$$

LP	SOCP
σ_1	σ_2
σ_2	σ_4
σ_3	σ_5

Table 3.4: Mapping table for the numbering of the dual variables.

$$\sigma_{5,i}, \sigma_{6,i} \leq 0 \qquad i \in \mathcal{N}. \tag{3.37}$$

Now that we have the dual problems of the deterministic one-stage LP and SOCP, we are able to determine the inclusion relation between their domains. But once again, we need a few notations first.

Definition 4.

- $\hat{\mathcal{S}}_{\texttt{LP}} = \mathbb{R}^{3n}$ is the variable space of the LP dual problem since σ_1, σ_2 and $\sigma_3 \in \mathbb{R}^n$.
- $\hat{\mathcal{S}}_{\texttt{SOCP}} = \mathbb{R}^{6n}$ is the variable space of the SOCP problem.
- $\hat{P}_{\texttt{SOCP}\to\texttt{LP}} : \hat{\mathcal{S}}_{\texttt{SOCP}} \to \hat{\mathcal{S}}_{\texttt{LP}} : \hat{P}_{\texttt{SOCP}\to\texttt{LP}}(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6) = (\sigma_2, \sigma_4, \sigma_5)$ is the projection from $\hat{\mathcal{S}}_{\texttt{SOCP}}$ to $\hat{\mathcal{S}}_{\texttt{LP}}$.
- $\hat{\mathcal{D}}_{\texttt{LP}} \subset \hat{\mathcal{S}}_{\texttt{LP}}$ is the feasible domain of the LP problem.
- $\hat{\mathcal{D}}_{\texttt{SOCP}} \subset \hat{\mathcal{S}}_{\texttt{SOCP}}$ is the feasible domain of the SOCP problem.

Looking back at (LP Dual), the similarity between the LP and the SOCP duals should be clear to the reader's eyes. First, note that the numbering of the σ's is not the same, see table 3.4 for a practical mapping. Then, apart from the variables and constraints related to the reactive power and voltage, only one constraint differs: $\sigma_{1,i} = \sigma_1 \; \forall \; i \in \mathcal{N}$ in the LP case, and the conic constraint in the SOCP case. At first sight, one could claim that the conic contraint is more restrictive and end up with the same conclusion as the primal case. On the contrary, we prove the opposite.

Proposition 2. $\hat{\mathcal{D}}_{\texttt{LP}} \subseteq \hat{P}_{\texttt{SOCP}\to\texttt{LP}}(\hat{\mathcal{D}}_{\texttt{SOCP}})$.

Proof. First, note that the above formulation of the proposition is equivalent to the following

$$\forall \; (\sigma_2, \sigma_4, \sigma_5) \in \hat{\mathcal{D}}_{\texttt{LP}}, \; \exists \; \sigma_1, \sigma_3, \sigma_6 \in \mathbb{R}^n \text{ such that } (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6) \in \hat{\mathcal{D}}_{\texttt{SOCP}},$$

which is more practical.

Let $(\sigma_2, \sigma_4, \sigma_5) \in \hat{\mathcal{D}}_{\texttt{LP}}$. This point already satisfies constraints (3.32), (3.34), (3.35) and (3.36). We choose $\sigma_6 \leq 0$ and $\sigma_3 \leq -\sigma_6$ such that constraints (3.33) and (3.37) are also fulfilled. Now we only have to find σ_1 such that constraint (3.31) is satisfied in order to complete the proof. Let us develop this constraint

$$\begin{bmatrix} C - \Gamma^T & R^T C^T & X^T C^T \\ -C - B^T & R^T C^T & X^T C^T \\ 2R^T & I - C^T & 0 \\ 2X^T & 0 & I - C^T \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} \in \mathbb{L}^{3n}$$

$$\Leftrightarrow \begin{bmatrix} \sum_{j \in C_i} \sigma_{1,j} - \Gamma_i \sigma_{1,i} + R_i \sigma_{2,A_i} + X_i \sigma_{3,A_i} \\ -\sum_{j \in C_i} \sigma_{1,j} - B_i \sigma_{1,i} + R_i \sigma_{2,A_i} + X_i \sigma_{3,A_i} \\ 2R_i \sigma_{1,i} + \sigma_{2,i} - \sigma_{2,A_i} \\ 2X_i \sigma_{1,i} + \sigma_{3,i} - \sigma_{3,A_i} \end{bmatrix} \in \mathbb{L}^3 \; \forall \; i \in \mathcal{N}.$$

The two first entries of this vector can be rewritten as

$$\sum_{j\in C_i} \sigma_{1,j} - \Gamma_i\sigma_{1,i} + R_i\sigma_{2,A_i} + X_i\sigma_{3,A_i} = \sum_{j\in C_i} \sigma_{1,j} - \sigma_{1,i} + \overbrace{R_i\sigma_{2,A_i} + X_i\sigma_{3,A_i} - (R_i^2+X_i^2)\sigma_{1,i}}^{\zeta}$$
$$-\sum_{j\in C_i} \sigma_{1,j} - B_i\sigma_{1,i} + R_i\sigma_{2,A_i} + X_i\sigma_{3,A_i} = -\sum_{j\in C_i} \sigma_{1,j} - \sigma_{1,i} + \zeta.$$

Since $(\sigma_2, \sigma_4, \sigma_5) \in \hat{\mathcal{D}}_{\texttt{LP}}$, $\sigma_{2,i} = \sigma_2 \ \forall\ i$. We impose the same restraint to σ_3: $\sigma_{3,i} = \sigma_3 \ \forall\ i$, such that $\sigma_3 \leq -\max_{i\in\mathcal{N}} \sigma_{6,i}$ in order to satisfy constraint (3.33). This yields

$$\begin{aligned}
&\begin{bmatrix} \sum_{j\in C_i} \sigma_{1,j} - \Gamma_i\sigma_{1,i} + R_i\sigma_2 + X_i\sigma_3 \\ -\sum_{j\in C_i} \sigma_{1,j} - B_i\sigma_{1,i} + R_i\sigma_2 + X_i\sigma_3 \\ 2R_i\sigma_{1,i} \\ 2X_i\sigma_{1,i} \end{bmatrix} \in \mathbb{L}^3 \\
\Leftrightarrow\quad &\left(\sigma_{1,i} - \sum_{j\in C_i} \sigma_{1,j}\right)^2 + 2\zeta\left(\sigma_{1,i} - \sum_{j\in C_i} \sigma_{1,j}\right) + \zeta^2 + 4R_i^2\sigma_{1,i}^2 + 4X_i^2\sigma_{1,i}^2 \\
&\leq \left(\sum_{j\in C_i} \sigma_{1,j} - \sigma_{1,i}\right)^2 + 2\zeta\left(\sum_{j\in C_i} \sigma_{1,j} - \sigma_{1,i}\right) + \zeta^2 \\
\Leftrightarrow\quad &4(R_i^2+X_i^2)\sigma_{1,i}^2 \leq 4\left(\sum_{j\in C_i} \sigma_{1,j} - \sigma_{1,i}\right)\Big(R_i\sigma_2 + X_i\sigma_3 - (R_i^2+X_i^2)\sigma_{1,i}\Big) \\
\Leftrightarrow\quad &0 \leq \left(\sum_{j\in C_i} \sigma_{1,j} - \sigma_{1,i}\right)\Big(R_i\sigma_2 + X_i\sigma_3\Big) - \sigma_{1,i}(R_i^2+X_i^2)\sum_{j\in C_i} \sigma_{1,j} \\
\Leftrightarrow\quad &\underbrace{\frac{\sigma_{1,i}\sum_{j\in C_i} \sigma_{1,j}}{\sum_{j\in C_i} \sigma_{1,j} - \sigma_{1,i}}}_{f(\sigma_1)} \leq \frac{R_i\sigma_2 + X_i\sigma_3}{R_i^2+X_i^2} \qquad \forall\ i \in \mathcal{N}.
\end{aligned}$$

We are now going to build σ_1 such that this last inequality is satisfied for any choice of σ_2 and σ_3. Defining

$$\alpha = \sum_{j\in C_i} \sigma_{1,j} \qquad \beta = \sigma_{1,i} \qquad \gamma = \frac{R_i\sigma_2 + X_i\sigma_3}{R_i^2+X_i^2}$$

allows us to rewrite the inequality

$$\frac{\alpha\beta}{\alpha-\beta} \leq \gamma.$$

An interesting property of the LHS function is $\lim_{\substack{\alpha\to\beta \\ \leq}} \frac{\alpha\beta}{\alpha-\beta} = -\infty$ for $\alpha, \beta \neq 0$. Using this property, we can build a σ_1 that keeps $f(\sigma_1)$ small enough. Proceeding recursively, we propose

$$\sigma_{1,0} = 1 \qquad \sigma_{1,i} = \frac{\sigma_{1,A_i} - \epsilon}{|C_{A_i}|} \text{ with } \epsilon \geq 0\ \forall i \in \mathcal{N},$$

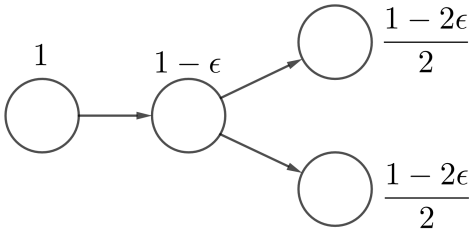


Figure 3.15: The recursive attribution of σ_1 on a 4-node network, the node on the left being the root.

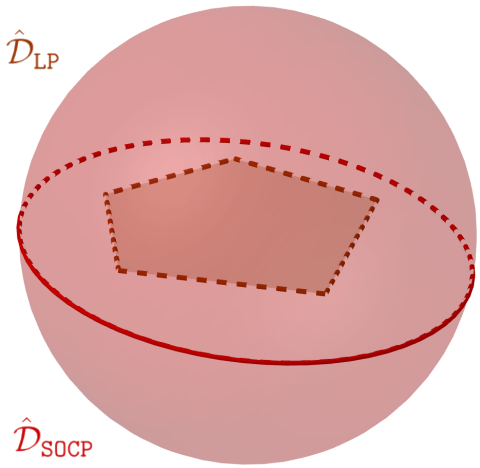


Figure 3.16: Schematic representation of Proposition 2: $\hat{\mathcal{D}}_{\texttt{LP}} \subseteq \hat{P}_{\texttt{SOCP}\to\texttt{LP}}(\hat{\mathcal{D}}_{\texttt{SOCP}})$.

which allows us to reach an arbitrary small value for $f(\sigma_1)$ by decreasing ϵ towards 0. This attribution is represented on a small example on figure 3.15.

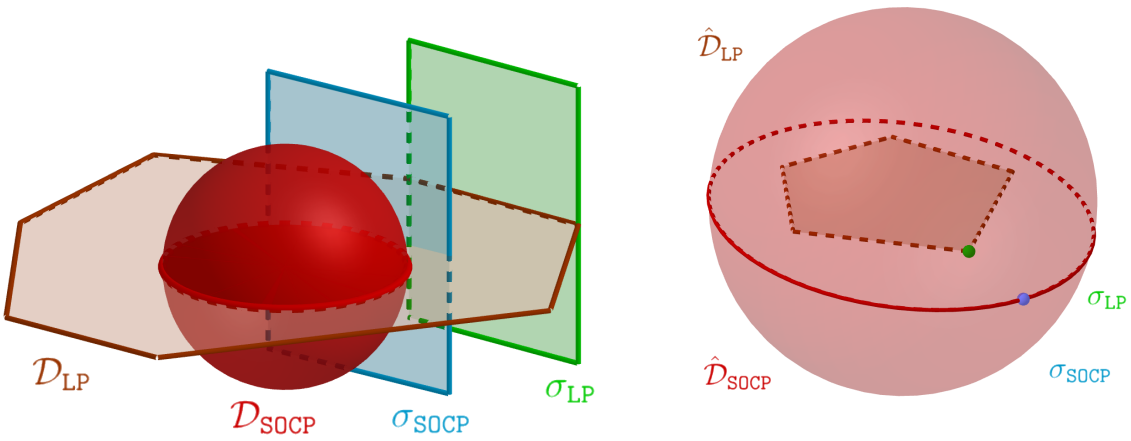
Using this σ_1, one can find an ϵ which ensures that the conic constraint is satisfied for any σ_2 and σ_3. □

This proposition is schematically represented on figure 3.16. In comparison to the primal domains, the inclusion relation is the opposite.

We can now conclude with an interesting geometrical interpretation of primal constraints, which provides a better understanding of propositions 1 and 2. The key property that we will apply is simply that each point of the dual domain corresponds to a constraint on the primal domain, and vice-versa. A point of the interior of the dual domain corresponds to a non-tight primal constraint, but a point of the closure of the dual domain corresponds to a tight primal constraint.

Knowing that, let us take a look at figures 3.17. $\sigma_{\texttt{LP}}$ being on the closure of $\hat{\mathcal{D}}_{\texttt{LP}}$, the corresponding primal constraint is tight for the LP domain, but is not tight for the SOCP domain since $\sigma_{\texttt{LP}}$ lies in the interior of $\hat{\mathcal{D}}_{\texttt{SOCP}}$. Similarly, $\sigma_{\texttt{SOCP}} \in \text{cl}(\hat{\mathcal{D}}_{\texttt{SOCP}}) \Leftrightarrow f_{\texttt{SOCP}}$ is tight for $\mathcal{D}_{\texttt{SOCP}}$, but $\sigma_{\texttt{SOCP}} \notin \hat{\mathcal{D}}_{\texttt{LP}} \Leftrightarrow f_{\texttt{SOCP}}$ is not satisfied for $\mathcal{D}_{\texttt{LP}}$.

Therefore, since $\hat{\mathcal{D}}_{\texttt{LP}} \subseteq \hat{P}_{\texttt{SOCP}\to\texttt{LP}}(\hat{\mathcal{D}}_{\texttt{SOCP}})$, all points of $\hat{\mathcal{D}}_{\texttt{LP}}$ also lie in $\hat{\mathcal{D}}_{\texttt{SOCP}}$, so all constraints defining $\mathcal{D}_{\texttt{LP}}$ are also satisfied by $\mathcal{D}_{\texttt{SOCP}}$, which yields $P_{\texttt{SOCP}\to\texttt{LP}}(\mathcal{D}_{\texttt{SOCP}}) \subseteq \mathcal{D}_{\texttt{LP}}$.



(a) Primal domains

(b) Dual domains

Figure 3.17: Link between the primal and dual domains: each dual point corresponds to a primal constraint. This allows us to deduce proposition 1 from proposition 2.

Chapter 4

Algorithm

The models derived in the previous section lie in the intersection of two reknown classes of optimization problems: on the one hand, convex non-differentiable optimization, typically tackled by cutting plane methods, and on the other hand, multi-stage optimization under uncertainty, typically tackled by approximate dynamic programming.

In 1991, Pereira and Pinto proposed the Stochastic Dual Dynamic Programming algorithm to solve multistage stochastic linear optimization problems [21]. By simulating the future state instead of enumerating all possibilities, they manage to avoid the well-known *curse of dimensionality* of dynamic programming. Combining dynamic programming, cutting plane methods and a decomposition of the mathematical program in order to exploit parallel computing, the `SDDP` algorithm has found great commercial success. It was initially developed in the framework of linear programming, and to the best of my knowledge, has not been applied to nonlinear programs so far.

In this chapter, we first present a gentle introduction to the algorithm, before discussing how it can be applied to convex programming. For a more comprehensive description of `SDDP`, the reader may refer to [66].

4.1 Stochastic Dual Dynamic Programming

4.1.1 Benders Decomposition

Before introducing more than two time stages and uncertainty, we first present the application of Kelley's cutting plane algorithm on a deterministic two-stage problem, in order to gain an insight of this concept, which is central to `SDDP`.

Consider a problem of the form

$$\begin{aligned} z^\star = \min_{x,y} \quad & c^T x + q^T y \\ \text{s.t.} \quad & x \in X \\ & y \in Y(x) \end{aligned}$$

where in our case, x denotes the decision variables of the first time stage, while y denotes the second one, X and Y correspond to their convex domain and $c^T x + q^T y$ stands for the total cost of operation on the two time stages. The main idea of the algorithm is to "relax the second-stage constraints, and to gradually account for their impact on the optimization problem by adding

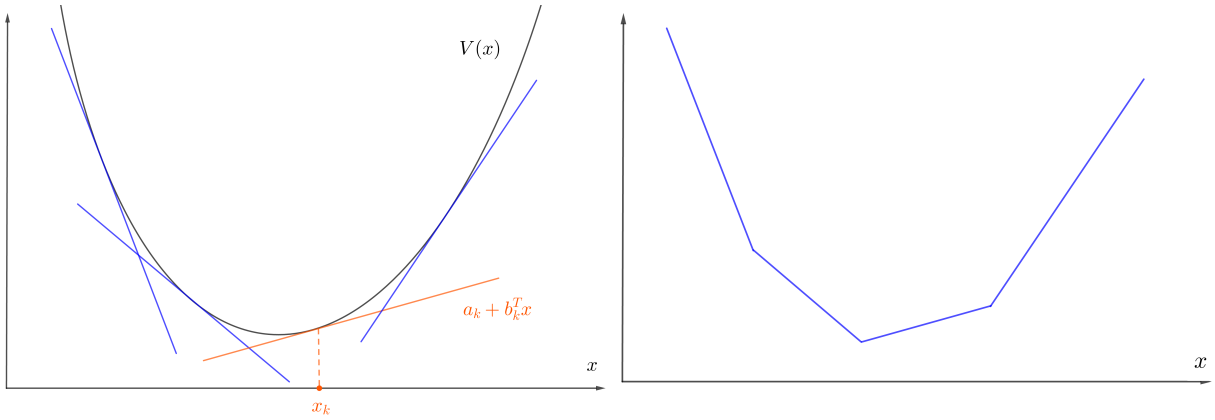


Figure 4.1: Illustration of the approximation of the value function by some of its supporting hyperplanes. The construction is represented on the left panel, while the resulting approximation is given on the right panel.

terms in the objective function of the problem that reflect the second-stage cost of a given first-stage trial decision" [66].

To this end, we decompose the problem into two subproblems, which leads us to the important concept of **value function** $V(x)$, that provides the best cost which can be achieved in the second stage given a first-stage decision x

$$\begin{aligned} V(x) = \min_y \quad & q^T y \qquad (\texttt{Slave}) \\ \text{s.t.} \quad & y \in Y(x). \end{aligned}$$

This allows us to rewrite the initial problem without the second-stage variables and constraints

$$\begin{aligned} z^\star = \min_x \quad & c^T x + V(x) \qquad (\texttt{Master}^\star) \\ \text{s.t.} \quad & x \in X. \end{aligned}$$

However, the challenge is that $V(x)$ is typically not available. The idea of the algorithm is to approximate it by a piecewise linear function, by employing supporting hyperplanes of V. To this end, given a first-stage variable x_k, we must find parameters a_k and b_k satisfying

$$\begin{aligned} V(x_k) &= a_k + b_k^T x_k \qquad \text{for some } x_k \in X \\ V(x) &\geq a_k + b_k^T x \qquad \forall\, x \in X. \end{aligned}$$

a_k and b_k can be computed for different trial decisions x_k by solving the `Slave` problem with $Y(x_k)$. Iteratively repeating this exercise yields a piecewise linear approximation of $V(x)$ as illustrated in Figure 4.1, and allows us to write the following problem

$$\begin{aligned} z_\kappa = \min_x \quad & c^T x + \theta \qquad (\texttt{Master}_\kappa) \\ \text{s.t.} \quad & x \in X \\ & \theta \geq a_k + b_k^T x \qquad k = 1, \ldots, \kappa. \end{aligned}$$

The constraints $\theta \geq a_k + b_k^T x$ are referred to as optimality cuts. The Benders decomposition algorithm consists in successively solving the `Master`, feeding the `Slave` with the obtained solution and then coming back to the `Master` with the resulting optimality cut. Moreover, the resolution of $\texttt{Master}_\kappa$ provides a lower bound z_κ to the optimal objective value of the initial

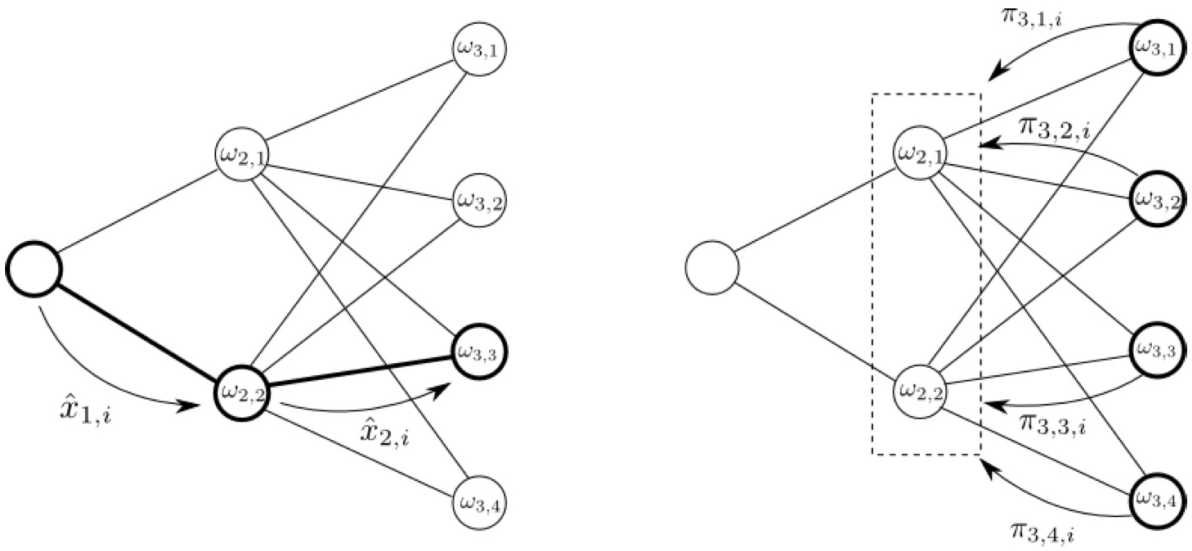


Figure 4.2: Lattice where each node stands for a different NLDS. Forward (left panel) and backward (right panel) pass in SDDP. The forward pass of Monte Carlo sample i generates trial decisions $\hat{x}_{t,i}$. The backward pass generates the optimality cut parameters $\pi_{t,\omega,i}$ that can be used for the previous time stage. Taken from [66].

problem $z^\star$. In the context of linear programming, i.e. if X and $Y(x)$ are polyhedra made of linear inequalities, the Benders decomposition algorithm converges finitely, i.e.

$$\exists\ \kappa^\star \in \mathbb{N} \text{ such that } z_{\kappa^\star} = z^\star.$$

A proof can be found in [66].

4.1.2 Adding multiple time stages and uncertainty

The Benders decomposition algorithm provides the skeleton of SDDP. We can now refine it by adding multiple time stages and uncertainty to the problem.

The concept of value function is thus extended to the **Nested L-shaped Decomposition Subproblem** at (t, ω) (NLDS(t, ω)), which is the subproblem that models time stage t under the realization ω of uncertainty. For the sake of clarity, we introduce a new graphical representation of uncertainty where each node corresponds to a different NLDS. Thus, each node stands for a time stage and a realization of uncertainty: this representation is called a **lattice** and although it differs slightly to the scenario tree presented in section 3.1.1, they are equivalent [66]. A graphical example of lattice is presented in Figure 4.2.

The idea of the algorithm is the same as Benders decomposition:

1. First, we propagate forward through the lattice by generating trial solutions at each NLDS and feeding a descendant NLDS with it. It consists in the **forward pass**.
2. Then, we proceed from the last time stages to the early ones by computing optimality cuts and adding them to the ancestor NLDSs. This composes the **backward pass**.

This process is illustrated in Figure 4.2, and the SDDP algorithm is composed of a succession of forward and backward passes that eventually converge.

Furthermore, two important features improve significantly the efficiency of SDDP, and allow it to outperform a related algorithm called nested decomposition:

- Instead of enumerating all possible outcomes, the forward pass relies on **Monte Carlo simulations**: each forward pass consists in K samples of the random process $(\omega_1, \dots, \omega_H)$, and each sample i provides a set of trial decisions $\hat{x}_{t,i}$. Although it yields probabilistic outcomes, simulating the future states instead of enumerating all possibilities allows us to avoid the well-known curse of dimensionality of dynamic programming.
- The optimality cut parameters generated during the backward pass are shared between all the `NLDS`s of the previous time stage. In the example of Figure 4.2, the four parameters $\pi_{t,\omega,i}$ are provided to both previous `NLDS`s. In this way, we make the best use of these parameters, and this allows a faster learning of the future time stages.

In light of that, we can now describe formally the `SDDP` algorithm as follows [66]:

Definition 5. `SDDP` algorithm.

`Forward pass`

- Solve `NLDS`(1). Let x_1 be the optimal solution. Initialize $\hat{x}_{1,i} = x_1$ for $i = 1, \dots, K$
- Repeat for $t = 2, \dots, H-1$, $i = 1, \dots, K$
  - Sample an outcome $\omega_{t,i}$ from the set Ω_t
  - Solve `NLDS`$(t, \omega_{t,i})$ with trial decision $\hat{x}_{t-1,i}$
  - Store the optimal solution as $\hat{x}_{t,i}$

`Backward pass`

- Repeat for $t = H, H-1, \dots, 2$
  - Repeat for $i = 2, \dots, K$
    - Repeat for $\omega \in \Omega_t$
      - ♢ Solve `NLDS`(t, ω) with trial decision $\hat{x}_{t,i}$
      - ♢ Store the optimality cut parameters resulting from the solution
    - Using these parameters, add an optimality cut to every `NLDS`$(t-1, \psi)$ for $\psi \in \Omega_{t-1}$

We can calculate that the forward pass needs to solve the initial `NLDS` once, before solving $H-2$ `NLDS`s for each Monte Carlo simulation. Given K Monte Carlo simulations, this results in a total of $1 + K \cdot (H-2)$ programs that need to be solved at each forward pass. Considering the backward pass, the resolution of $\sum_{t=2}^{H} |\Omega_t|$ programs is needed per sequence of trial decisions. Thus, over K Monte Carlo trials, the total backward pass requires the resolution of $K \cdot \sum_{t=2}^{H} |\Omega_t|$ programs. Therefore, the computational effort needed by each iteration scales linearly with K.

On the other hand, a larger K provides a better learning of the future states for the algorithm. Indeed, "the role of the forward pass is to determine the point $\hat{x}_{t,i}$ at which the value function should be explored" [66]. Hence, a larger K allows convergence in fewer iterations.

As a result, the choice of K involves a trade-off and is actually an important feature of the design of the algorithm. This will be addressed on a practical test case in section 5.4, and we will discuss the influence of uncertainty on the optimal choice of K.

4.2 Application on Convex Programming

One step still remains unclear in definition 5: how to extract the optimality cut parameters from the solution of `NLDS`(t, ω) in the backward pass ? Similarly, coming back to Benders

decomposition, how do we compute a_k and b_k from the solution of $V(x_k)$?

In the linear case, it is straightforward. If we have

$$Y(x) = \{Wy = h - Tx,\ y \geq 0\},$$

the dual of the `Slave` can be expressed as

$$\begin{aligned} \max_{\pi} \quad & \pi^T(h - Tx) \qquad (\texttt{Slave Dual}) \\ & \pi^T W \leq q^T \end{aligned}$$

and the supporting hyperplane of $V(x)$ at x_k is given by

$$\pi_k^T(h - Tx),$$

where π_k^T denotes the dual optimal multiplier of the `Slave` (i.e. the optimal solution of the `Slave Dual`) fed with the trial decision x_k. In addition, "since the `Slave Dual` has a finite number of optimal solutions, only a finite number of supporting hyperplanes exist for $V(x)$, therefore the function must be **piecewise linear** convex" [66].

However, for `SOCP` and `SDP`, this result does not hold. But Kelley's cutting plane method is designed for convex programming, and linearity has not been assumed in the derivation of the `SDDP` algorithm in the last section. It should be mentioned that Pereira had hinted at this idea back in 1991 [21]. In particular, he mentions briefly in paragraph 4.3: *The algorithm can be extended to the nonlinear case. Naturally, convergence to the global optimum can only be ensured under the usual convexity conditions for nonlinear problems.*

In order to compute the supporting hyperplanes in the nonlinear convex case, a new notion and a proposition are needed.

Definition 6. (From [66]). Consider a function g, π is a **subgradient** of g at u if

$$g(w) \geq g(u) + \pi^T(w - u) \text{ for all } w.$$

Proposition 3. (From [66]). Define $c(u)$ as the optimal value of the following mathematical program

$$\begin{aligned} c(u) = \min_{x} \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq u_i \qquad i = 1, \ldots, m \\ & x \in \text{dom} f_0, \end{aligned}$$

and suppose that $\text{dom} f_0$ is a convex set and f_0, f_i are convex functions for $i = 1, \ldots, m$.

- Then $c(u)$ is a convex function.
- Suppose strong duality holds, and denote $\lambda^\star$ as the maximizer of the dual function

$$\begin{aligned} \min_{x \in \text{dom} f_0} \quad & f_0(x) - \lambda^T(f(x) - u) \\ \text{s.t.} \quad & \lambda \leq 0, \end{aligned}$$

  Then $\lambda^\star$ is a subgradient of $c(u)$.

In the `SOCP` case, the value function of stage t is given by

$$V_t(x_{t-1}) = \min \quad \sum_{i \in \mathcal{N}} MC_{i,t} \cdot p_{i,t} + PL \sum_{i \in \mathcal{E}} R_i \cdot \ell_{i,t}$$

$$(\pi_i) \qquad \text{s.t.} \quad x_{i,t} = x_{i,t-1} + \eta c_i \cdot bc_{i,t} - \frac{bd_{i,t}}{\eta d_i} \qquad\qquad i \in \mathcal{N}$$

Other convex constraints of stage t

The equality constraint can be rewritten equivalently with two inequality constraints of opposite sign. Proposition 3 can then be applied, and yields

- $V_t(x_{t-1})$ is a convex function.
- The optimal multiplier $\pi^\star_{\text{trial}} = (\pi^\star_{\text{trial},1}, \ldots, \pi^\star_{\text{trial},n})$ is a subgradient of $V_t(x_{t-1})$ at the given value $x_{t-1,\text{trial}}$.

Then, merely writing down the definition of a subgradient yields the **optimality cut** for the `SOCP` case, given a trial value $x_{t-1,\text{trial}}$

$$V_t(x_{t-1}) \geq V_t(x_{t-1,\text{trial}}) + \pi^T_{\text{trial}}(x - x_{\text{trial}}).$$

Finally, a similar derivation can be performed for the `SDP` case. Since it is exactly the same developments but for three phases, it is not redone for the sake of concision.

4.3 Shape of the Value Function

As mentioned in the previous section, the value function is piecewise linear in the linear case, and this is due to the fact that the `Slave Dual` admits a finite number of solutions since it is a linear program. A relevant question is whether this result extends to the convex case or not. In order to see whether the same reasoning as the linear case can be done, we must derive the dual of the second-order cone problem.

Proposition 4. (From [7]). The dual of the `SOCP`

$$\begin{aligned} \min_x \quad & f^T x \\ \text{s.t.} \quad & ||A_i x + b_i||_2 \leq c_i^T x + d_i \qquad i = 1, \ldots, m \end{aligned}$$

with variables $x \in \mathbb{R}^n$, can be expressed as

$$\begin{aligned} \max_{u,v} \quad & \sum_{i=1}^m b_i^T u_i - d_i v_i && \text{(SOCP Dual)} \\ \text{s.t.} \quad & \sum_{i=1}^m (A_i^T u_i - c_i v_i) + f = 0 \\ & ||u_i||_2 \leq v_i && i = 1, \ldots, m \end{aligned}$$

with variables $u_i \in \mathbb{R}^{n_i}, v_i \in \mathbb{R}, i = 1, \ldots, m$. The problem data are $f \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{n_i \times n}, b_i \in \mathbb{R}^{n_i}, c_i \in \mathbb{R}$ and $d_i \in \mathbb{R}, i = 1, \ldots, m$.

Then, we are interested in whether the `SOCP Dual` admits a finite number of solutions as in the linear case. The answer is negative, at least in the general case since the domain of the `SOCP Dual` is the intersection of a second-order cone an a polyhedron, and not only a polyhedron.

More specifically, in order to gain an insight about the shape of the value function, we consider a perturbation fo the right-hand side d_i. Then what we obtain is a perturbation in the objective function of the dual, at the term $d_i v_i$. In order to understand the geometry of the value function, it is necessary to understand how the dual optimal multipliers behave in response to a

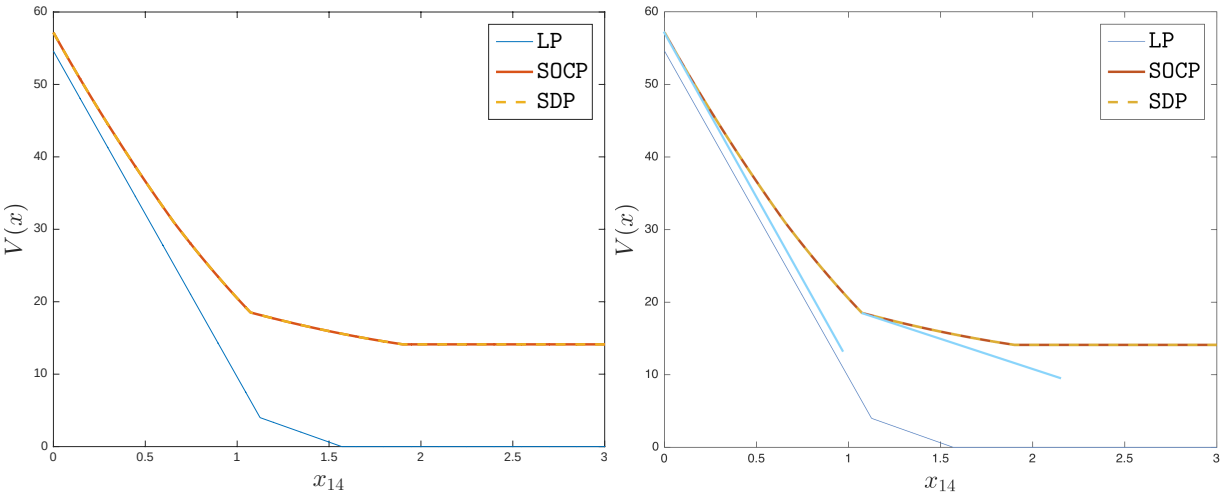


Figure 4.3: Numerical experiment on the shape of the value function for the three formulations. We can stress that SOCP and SDP provide the same solution. Green linear segments have been added to the right figure in order to highlight the nonlinearity of the SOCP and SDP value function, since it was not easily identifiable on the left figure. Otherwise, the two figures are identical.

perturbation in d_i. Since the dual feasible region is an intersection of a second-order cone and a polyhedron, there is nothing to preclude the possibility that the dual optimal multiplier changes smoothly. So as a general principle, the value functions will be convex, but their shape will not follow any specific functional form, e.g. piecewise affine. This does not pose a problem (unless one is interested in finite convergence proofs), because as long as the value function is convex we can anyways apply Kelley's method (and by extension SDDP) in order to attack the problem.

In order to confirm that theoretical analysis, we tried to see whether we would be able to experimentally capture nonlinearities in the value function of SOCP and SDP. As presented in Figure 4.3, the answer is positive, and this single counter-example proves our intuition that the value function is *not necessarily piecewise linear* when addressing nonlinear problems.

This figure has been obtained on the network presented in the next chapter. All batteries are initially empty except the one at bus 14, which varies between 0 and 3. The maximum battery discharge constraint is relaxed such that bus 14 can discharge all its power, and supply the whole network. Further comments can be stressed.

- The LP solution achieves tu fulfill all the demand from $x_{14} \gtrsim 16$. Note the two different slopes of the value function. The steepest is when we are reducing the power produced at the expensive source ($MC = 50$), and then the second one is when we do not produce anymore at this source, but we are reducing the power produced at the cheap source ($MC = 10$).
- For SOCP and SDP, on the other hand, an interesting behavior arises. Bus 14 can discharge its power all over the network, and not only in a local neighbourhood. And as this flowing power expands wider, Kirchhoff's laws come into play more significantly, introducing nonlinearities. At the end, SOCP and SDP are not able to reach 0 because of the voltage constraint.
- While the above theoretical discussion has only been done for SOCP, we observe that this experiment extends the conclusion to SDP as well. It convinces us that a similar discussion can be done for SDP.

A last comment should be mentioned about our implementation of SDDP. In the present chapter, we did not consider infeasibilities. But in the complete SDDP algorithm, when a NLDS is

infeasible due to a poor trial decision, a feasibility cut is added to the ancestor in order to make sure that it does not occur again. However, in the considered test case that will be presented in the next chapter, each `NLDS` is feasible so the need for feasibility cuts is avoided. This is further discussed in section 5.7.2, along with other further improvements on this subject.

Chapter 5

Results

5.1 Optimization Software

If SDDP has not been applied to nonlinear programming so far, there is a reason: it requires to solve a significant amount of subproblems repeatedly, wich makes it hardly tractable for nonlinear programs. Therefore, the efficiency was a crucial concern in the implementation of SDDP, particularly when applying it on SDP. Before our final Julia implementation, we implemented the algorithm in three other optimization languages, but none of them suited our modeling and efficiency requirements. In order to grasp the dominance of Julia over the latter, we must keep in mind an important classification of programming languages.

- In **interpreted languages**, the execution is directly done step-by-step from source code, without any pre-runtime translation. "They must be parsed, interpreted, and executed each time the program is run, thereby greatly adding to the cost of running the program. For this reason, interpreted programs are usually less efficient than compiled programs" [74]. Interpreted languages include AMPL, Python, R, Matlab to name a few.
- In **compiled languages**, the source code is first translated into machine code – this step is called the compilation. Then, the resulting machine code can be executed very efficiently, and any number of times without repeating the compilation. Compiled languages include C/C++, Julia, Pascal, Fortran to name a few.
- In a compiled language, the overhead is incurred just once, during compilation, then it only needs to be loaded and executed. However, each line of an interpreted program must be translated each time it is executed, leading to a higher overhead [74]. For example, if a piece of code is used N times in an algorithm, the translation will be incurred only once by a compiled language, instead of N times for an interpreted one.

5.1.1 Convex Modeling Toolboxes

As mentioned above, we have implemented the algorithm in four different reknown convex modeling languages/toolboxes. However, only one of them suited our modeling and efficiency requirements. An illustration of our progression is presented in Figure 5.1 and the drawbacks of the first three languages/toolboxes are given hereafter. Note that since SDP is formulated in complex variables, a language supporting them would be convenient, but otherwise SDP can be reformulated in real variables (see below).

1. AMPL does not support complex variables, and cannot handle semi-definite constraints as

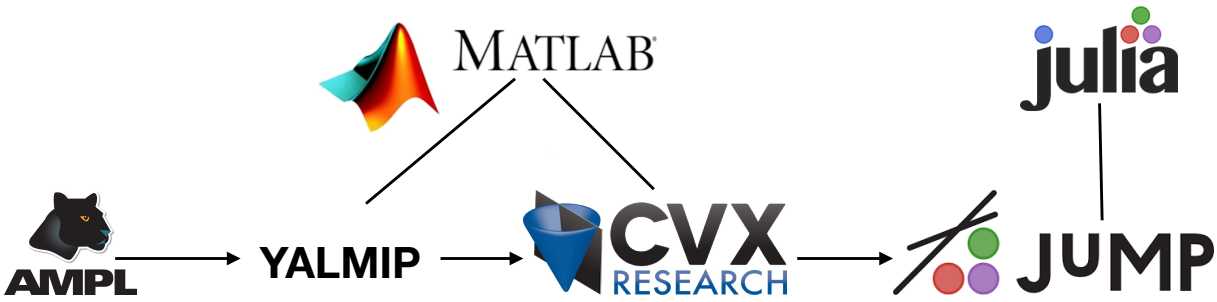


Figure 5.1: Illustration of our pursuit of a suitable convex modeling toolbox.

of today's date. If it was possible, solvers such as `Mosek` would update their `AMPL` version such that it could be used to solve `SDP`.

2. `YALMIP` is a `Matlab` toolbox for optimization modeling developed and maintained predominantly by Johan Löfberg from Linköping University, Sweden. Built on `Matlab`, it has the convenience to support complex variables. It can also handle `SDP`s, but unfortunately, it is not possible to extract the optimal dual multipliers of an `SDP` problem, which renders the computation of optimality cuts impossible.

3. `CVX` is "a popular modeling framework for disciplined convex programming that turns `Matlab` into a modeling language, allowing constraints and objectives to be specified using standard `Matlab` expression syntax" [75]. It supports complex variables, solves `SDP`s and their optimal dual multipliers can be extracted. However, solving one single `SDP` takes 2 seconds on a personal computer, rendering the `SDDP` implementation not tractable.

4. `JuMP` is "an open-source modeling language that allows users to express a wide range of optimization problems (linear, mixed-integer, quadratic, conic-quadratic, semidefinite, and nonlinear) in a high-level, algebraic syntax. `JuMP` takes advantage of advanced features of the `Julia` programming language to offer unique functionality while achieving competitive performance" [76]. `JuMP` remedies to `CVX` efficiency issue for the following reasons:

   (a) Since it is written in `Julia`, it is compiled instead of interpreted.

   (b) It achieves performance on par with commercial modeling tools, as benchmarks show in [76]. As a result, it has recently received the INFORMS Computing Society (ICS) Awards 2016 Prize[1].

   (c) `JuMP` allows us to modify the parameters of a problem and solve it with this modification without having to generate a whole new model. This feature is particularly relevant to `SDDP`, since the same `NLDS`s are solved at each Monte Carlo sample, they only differ from the trial decision that is seen as a parameter by the language. This is exploited in [77], and we have added this improvement to our code.

   (d) After all, `JuMP` supports *warm starts*, i.e. the possibility, when solving a sequence of similar problems, to start the resolution with the solution of the previous problem in order to save time by not starting from scratch every time. This is also relevant to `SDDP`, since the same `NLDS`s, only differing from additional cuts and different trial decisions, are solved a substantial amount of times. Our code also implements warm starts.

However, while displaying these advantages, `JuMP` does not support complex variables in its current version. Hence, `SDP` must be rewritten in real variables in order to be solved via `JuMP`.

[1] `https://juliacomputing.com/press/2016/12/15/jump-prize.html`

The constraints defining can be cast into three categories.

- The inequalities already involve only real variables and can remain unchanged;
- The complex equalities can simply be split into the equality of the real parts and the equality of the imaginary parts;
- The semi-definite constraint seems more challenging, but Boyd and Vandenberghe help us to overcome this task.

Proposition 5. (From [7]). Let $\mathbf{X} \in \mathbb{C}^{n\times n}, \mathbf{X} = \mathbf{X}^H$. We have

$$\mathbf{X} \succeq 0 \qquad \Leftrightarrow \qquad \begin{bmatrix} \Re\mathbf{X} & -\Im\mathbf{X} \\ \Im\mathbf{X} & \Re\mathbf{X} \end{bmatrix} \succeq 0.$$

Applying this proposition to our semi-definite constraint yields

$$\begin{aligned}
\mathbf{W} = \begin{bmatrix} \mathbf{V} & \mathbf{S} \\ \mathbf{S}^H & \mathbf{I} \end{bmatrix} \in \mathbb{S}_+ \Leftrightarrow \quad & \begin{bmatrix} \mathbf{V} & \mathbf{S} \\ \mathbf{S}^H & \mathbf{I} \end{bmatrix} \succeq 0, && \mathbf{W} = \mathbf{W}^H \\
\Leftrightarrow \quad & \begin{bmatrix} \Re\mathbf{W} & -\Im\mathbf{W} \\ \Im\mathbf{W} & \Re\mathbf{W} \end{bmatrix} \succeq 0, && \begin{cases} \Re\mathbf{W} = \Re\mathbf{W}^T \\ \Im\mathbf{W} = -\Im\mathbf{W}^T \end{cases} \\
\Leftrightarrow \quad & \begin{bmatrix} \Re\mathbf{V} & \mathbf{P} & -\Im\mathbf{V} & -\mathbf{Q} \\ \mathbf{P}^T & \Re\mathbf{I} & \mathbf{Q}^T & -\Im\mathbf{I} \\ \Im\mathbf{V} & \mathbf{Q} & \Re\mathbf{V} & \mathbf{P} \\ -\mathbf{Q}^T & \Im\mathbf{I} & \mathbf{P}^T & \Re\mathbf{I} \end{bmatrix} \succeq 0, && \begin{cases} \Re\mathbf{V} = \Re\mathbf{V}^T \\ \Re\mathbf{I} = \Re\mathbf{I}^T \\ \Im\mathbf{V} = -\Im\mathbf{V}^T \\ \Im\mathbf{I} = -\Im\mathbf{I}^T, \end{cases}
\end{aligned}$$

which involve only real variables and can be modeled in `JuMP`.

5.1.2 Solvers

Given its very good reputation in convex programming and competitive performances on semi-definite programming [19], we opted for `Mosek`. This solver implements a state-of-the-art primal-dual interior point method and is widely employed in the financial, energy and forestry industry [20]. `Mosek` uses its same reknown interior-point optimizer for our `SDP`, `SOCP` and even the `LP`.

However, `Mosek` performed poorly when solving `SDDP` on the `LP` formulation of our test case, while it encountered no issue for `SOCP` and `SDP`. We may attribute this behaviour to limited numerical precision. Thus, we resorted to `Gurobi` for `LP`.

Another drawback of `Mosek` is that it cannot be warm-started. Indeed, "since there are no known generally reliable ways to hot-start interior-point methods. This is an open research topic" [78]. Therefore, another solver which supports warm starts could be tested in future works. For instance, `SCS` solves `SDP`s using `ADMM` and can be warm-started. Other performant `SDP` solvers can be found in [19].

At last, all the tests of this work were realized on a MacBook (Retina, 12-inch, Early 2015) with a 1,3 GHz Intel Core M processor and 8 GB of memory 1600 MHz DDR3. All the presented results were obtained by `JuMP` code running on the versions of `Julia` packages given in Table 5.1, and the 8.1.0.47 release of `Mosek`.

Package	Version	Branch if not released
`Julia`	0.6.2	
`JuMP.jl`	0.17.1+	master
`MathOptInterface.jl`	0.3.0	
`MathOptInterfaceMosek.jl`	0.0.3+	bl/moi0.3
`Mosek.jl`	0.8.3	
`Gurobi.jl`	0.3.3+	odow/moi

Table 5.1: Versions of `Julia` packages used in this work.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
R_i	.001	.0883	.1384	.0191	.0175	.0482	.0523	.0407	.01	.0241	.0103	.001	.1559	.0953
X_i	.12	.1262	.1978	.0273	.0251	.0689	.0747	.0582	.0143	.0345	.0148	.12	.1119	.0684

Table 5.2: Impedance of the lines composing the test case network, given in Ω. Taken from [8].

5.2 Test Case

5.2.1 Description

The test case network is composed of 15 buses radially connected and rooted at node 0, the swing bus. The structure of the network and the impedance of the lines are taken from [8] and are given in Table 5.2 and Figure 5.2, respectively. It is a MATPOWER[2] case provided by N-SIDE[3], and reproduces a network from Liège. We aim at optimizing a residential microgrid with a significant penetration of distributed energy resources, as illustrated in figure 1.4. Therefore, each bus models a house equipped with solar panels and a local storage device, e.g. a TESLA Powerwall. Apart from bus 0, which stands for the substation linking the microgrid to the rest of the network. At this node, power can be bought from the electric utility company, e.g. ENGIE, at the market price. In addition, the network is provided with a micro CHP generator connected to bus 11.

The market price depends on the period of the day: it is cheaper during off-peak hours (i.e. 10 pm – 6 am) than during rush hours, i.e. the rest of the day, as in Belgium [79]. However, the price of generating power from the micro CHP is constant over the day since it just depends on the fuel price. We assume that it is cheaper than rush market price, but more expensive than off-peak market price

$$MC_{0,\text{off-peak}} = 10 \qquad MC_{0,\text{rush}} = 50 \qquad MC_{11,t} = 20 \quad \forall\, t.$$

We model bus 0 as an unlimited source of electricity, whereas the micro CHP can only deliver a limited amount of power

$$p_{\max,0,t} = \infty \qquad p_{\max,11,t} = 2.5 \qquad [\text{kW}],\ \forall\, t.$$

Note that this unlimited source of electricity ensures that the problem will always be feasible, for each time stage.

The objective is to operate this network over a day. However, we cannot afford an hour-precision since 24 time periods would not be tractable on a personal computer when optimizing the `SDP` formulation. We opt for 5 time periods of 4 hours, starting at 6 am and ending at 10

[2] `http://www.pserc.cornell.edu/matpower/`
[3] `https://www.n-side.com/`

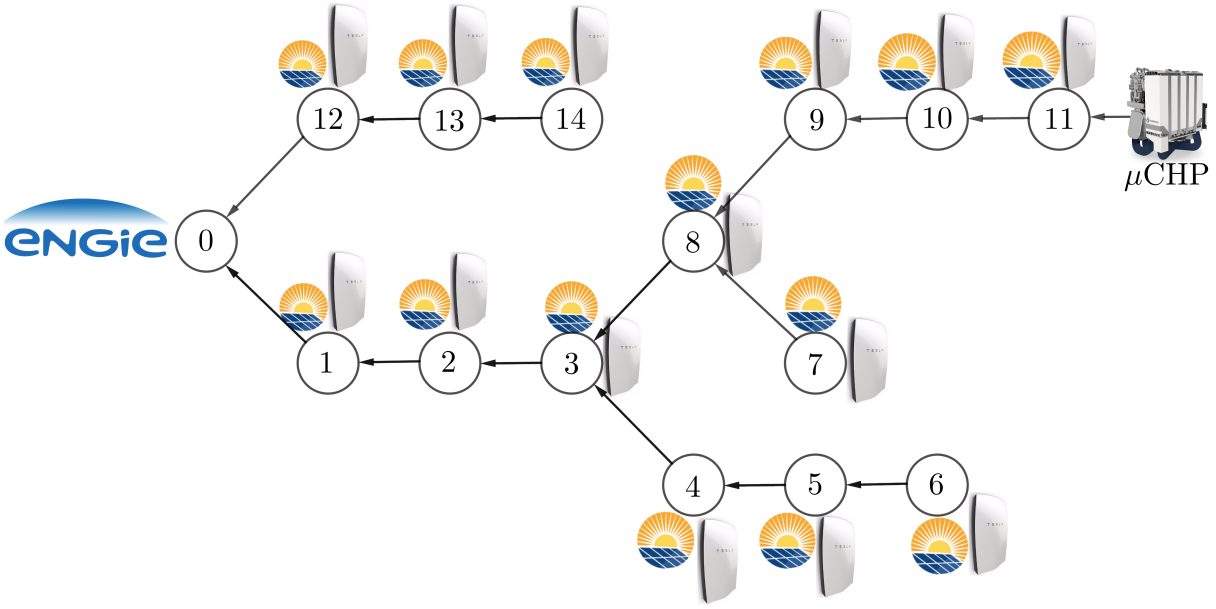


Figure 5.2: Radial structure of the test case network.

pm. In this way, it is tractable, matches the market price hours and only ignores the dark part of the day, when solar panels do not produce. Therefore,

$$\mathcal{T} = \{1, 2, 3, 4, 5\} \sim \{6 \text{ am}, 10 \text{ am}, 2 \text{ pm}, 6 \text{ pm}, 10 \text{ pm}\}, \qquad H = 5.$$

The power demand is not constant during the day. There is a peak of consumption in the morning and one in the afternoon, around 6.30 pm. In order to account for this, we adopt the daily load variation characteristic given in [80], depicted in Figure 5.3a. Further, the solar generation also varies over daytime. The total solar-PV power generation of Elia in Belgium at the 14th March 2018 is presented in Figure 5.3b. It was a bright day, and the profile exhibits a Gaussian form. From these two profiles, we approximate for the five time periods

$$p_{\text{load}} = \begin{bmatrix} 1 & 4/3 & 7/6 & 3/2 & 4/3 \end{bmatrix} \qquad p_{PV} = \begin{bmatrix} 0 & 1/3 & 1 & 1/2 & 0 \end{bmatrix},$$

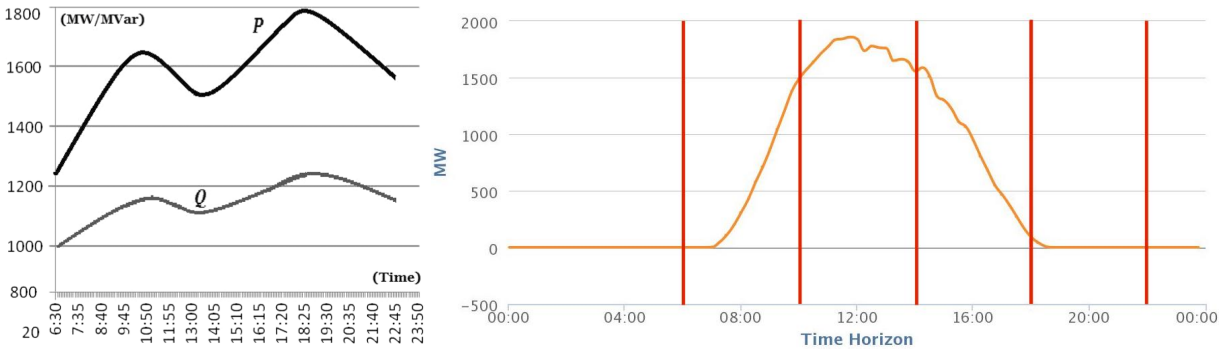
both given in $\left[\frac{\text{kW}}{4}\right]$, p_{load} is per bus and p_{PV} is per solar panel. Indeed, p_{load} sums to $19/3 \approx 6.4$ kW h, the average household electricity consumption in France [81]. Regarding solar generation, we assume that houses are equipped with standard solar panels of dimensions 1.64 m^2 and 250 W maximal production. Therefore, in the ideal case of 4 hours of full sun between 10 am and 2 pm, 1 kW h is produced per solar panel.

However, the solar production must be lowered when the weather is not bright. Moreover, this reduction is uncertain since it depends on the weather. We model it by a random scaling factor WF which can take four possible values

$$P(WF = 0) = 0.1 \qquad P(WF = 1/4) = 0.4 \qquad P(WF = 1/2) = 0.4 \qquad P(WF = 1) = 0.1,$$

respectively standing for rainy, cloudy, partly cloudy and sunny. Moreover, we assume this probability distribution to hold for every time period, i.e. the current weather has no influence on the next forecast. This results in the stochastic process represented in Figure 5.4. The net demand of a bus can then be computed

$$pD = \begin{bmatrix} 1 & 4/3 & 7/6 & 3/2 & 4/3 \end{bmatrix} - PV \cdot WF \cdot \begin{bmatrix} 0 & 1/3 & 1 & 1/2 & 0 \end{bmatrix},$$



(a) (b)

Figure 5.3: Daily load and solar profiles used in this document. Left: daily load variation characteristic, from [80]. Right: total solar-PV power generation of Elia in Belgium at the 14th March 2018. The monitored solar PV capacity is 3 369.05 MW and measures are taken every 15 minutes. The red segments delimit the considered time periods. Taken from [82].

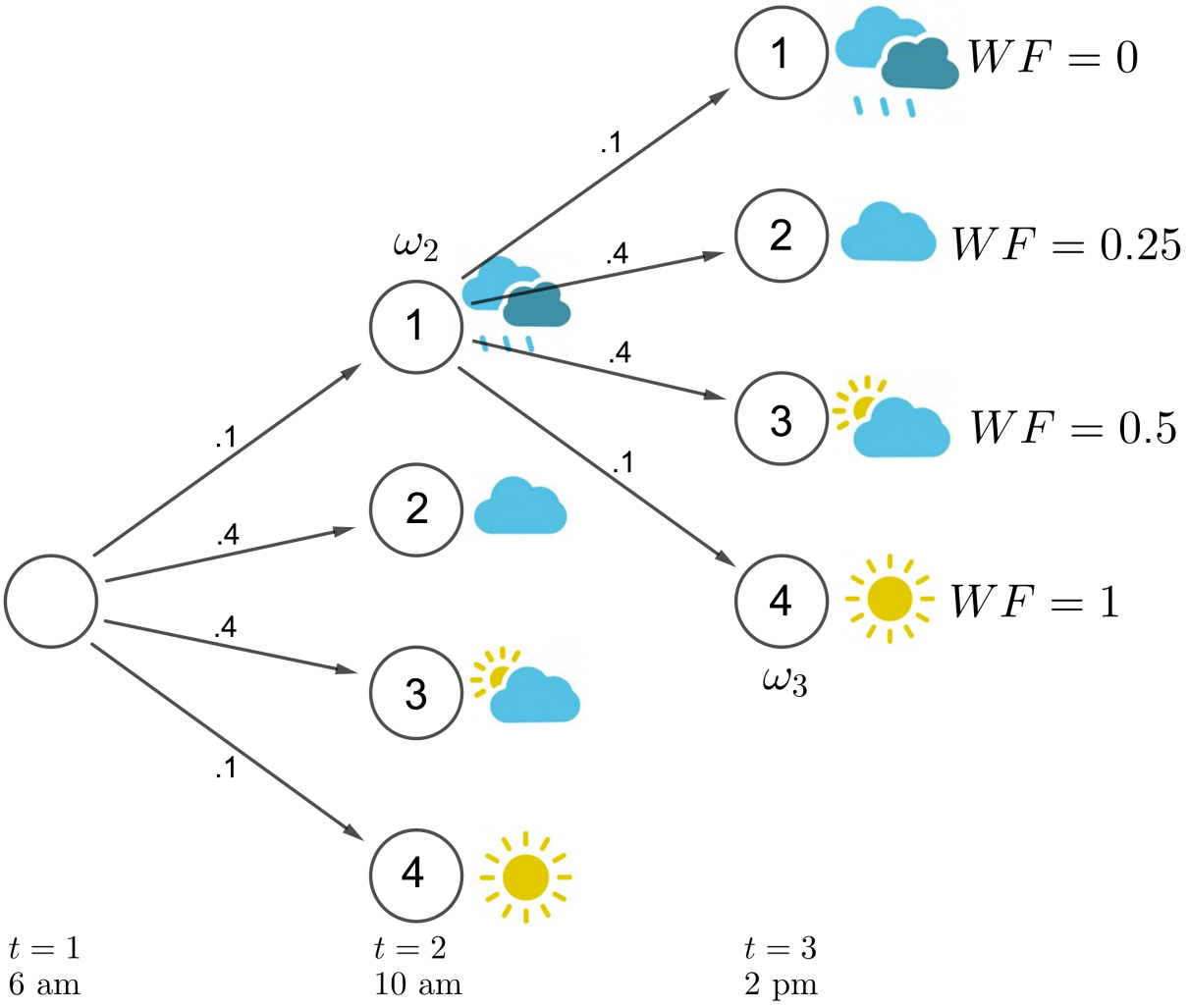


Figure 5.4: Representation of the model of uncertainty adopted for the test case.

where PV stands for the number of solar panels equipping each house.

Furthermore, we assume that each bus is equipped with a Tesla Powerwall. From [83], we have the technical specifications

$$x_{\max} = 13.5\,\text{kW h} \qquad bc_{\max} = bd_{\max} = 5\,\text{kW} \qquad \eta c_{\max} = \eta d_{\max} = 0.9.$$

Finally, we allow voltage magnitude to vary up to 10 % from its nominal value and assume that we start the day with empty batteries

$$v_{\min} = 0.81 \qquad v_{\max} = 1.21 \qquad x_{\text{init}} = 0,$$

where v stands for the squared voltage magnitude. Also, reactive power parameters are obtained as a third of real power ones, and shunt susceptance and conductance effects are neglected. While basic and simplified, this test case has the advantage to be tractable and permits a physical understanding of the solution, as presented in the following section.

5.2.2 Linear Solution

The solution of this test case given by the linear formulation is presented in Table 5.3 and provides an optimal cost of 614.629. Only the first three time stages are listed for ease of presentation. We can make the following observations

- At $t = 1$, we take advantage of the lowest power price of the day to buy an important amount of power from the electric utility company and store 64 % of it in the batteries, i.e. about 2 kWh per bus.
- Thanks to these provisions, power does not need to be bought at $t = 2$. The load is fulfilled only by discharging the batteries, 2/3 of the total storage in case of rain. In case of full sun however, the solar production suffices to balance the demand and the surplus is stored, even though the solar intensity is only at the third of its maximum.
- At the third time stage, we must resort to turn on the micro CHP in case of rain. In three cases, it is used at its maximal production capacity while storage could contribute instead. It is due to the fact that in subsequent time stages, power will need to be bought at the rush hour market price (50) in case of bad weather because the micro CHP will not be sufficient to satisfy the demand. Therefore, power produced now at bus 11 is cheaper than the market power, even though it undergoes a 0.81 factor due to charge and discharge efficiency. Also, note that the sun is at its zenith, and charges the batteries for the three other realizations of uncertainty.

5.2.3 Conic Solution

While the linear program provides a meaningful solution, it does not take the physical constraints governing power flows into account: Kirchhoff's laws. In order to consider them, we must resort to more complete formulations. `SOCP` and `SDP`, when tight, respect Kirchhoff's laws. Moreover, they provide the same solution since the considered network is assumed to be balanced, as discussed in section 3.1.3. Then, let us refer to it as the 'conic solution'. Besides, as the `LP` feasible set is larger, the linear solution is a lower bound of the conic one, as presented in section 3.5.1.

The conic solution of the test case with 5 solar panels per bus and $PL = 50$ is presented in Table 5.4. It yields an optimal generation cost of 659.861, which corresponds to an increase

$t=1$				$t=2$				$t=3$			
ω	p_0	p_{11}	$\sum_i x_i$	ω	p_0	p_{11}	$\sum_i x_i$	ω	p_0	p_{11}	$\sum_i x_i$
								1	0	10	3.580
				1	0	0	10.617	2	0	0.514	12.130
								3	0	0	27.417
								4	0	0	57.503
								1	0	10	10.061
				2	0	0	17.098	2	0	0	18.148
								3	0	0	33.898
								4	0	0	63.443
1	48.842	0	31.357					1	0	10	16.543
				3	0	0	23.580	2	0	0	24.630
								3	0	0	40.380
								4	0	0	69.909
								1	0	3.998	21.852
				4	0	0	35.557	2	0	0	36.607
								3	0	0	51.502
								4	0	0	82.520
p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}
0	10	20	1	1/3	50	20	4/3	1	50	20	7/6

Table 5.3: The first three time stages of the `LP` solution for the test case, with 5 solar panels per bus. p is given in [kW/4] and x in kWh.

of 7.36 % of the linear optimal cost. In other words, Kirchhoff's laws 'charge' 7.36 % of the objective. Indeed, they prevent us from buying as much as the linear solution at the market off-peak price. Therefore, power must be bought later on at a higher price, as displayed in Table 5.4.

However, the objective to minimize has changed from the linear formulation. Resistive power losses $PL \cdot \sum_{i \in \mathcal{E}} R_i \cdot |I_i|^2$ are added to the generation cost. The power losses account for 58.875, so the total cost, including generation cost and power losses, is given by 718.736. Thus, the 7.36 % increase of generation cost is also due to the presence of this penalty term: a higher amount of power bought at the market off-peak price would have implied larger currents in order to store it in the network batteries, which would have resulted in more important power losses. Nevertheless, if PL is too low the relaxations are not tight and the obtained solution not physical. This issue is discussed in the next section.

5.3 Exactness of the Relaxations

5.3.1 Theoretical Conditions

Second-Order Cone Program

As stressed in section 3.3.3, a sufficient condition for the tightness of the `SOCP` relaxation consists in

1. $PL > 0$, in this way we aim at minimizing the current, and therefore tighten the second-order cone constraint;

$t=1$				$t=2$				$t=3$			
ω	p_0	p_{11}	$\sum_i x_i$	ω	p_0	p_{11}	$\sum_i x_i$	ω	p_0	p_{11}	$\sum_i x_i$
								1	0	10	0.762
				1	0	8.751	7.905	2	0	3.801	12.355
								3	0	0	24.701
								4	0	0	55.538
								1	0	10	2.593
				2	0	4.530	9.737	2	0	1.738	12.343
								3	0	0	26.531
1	36.142	0	19.033					4	0	0	57.356
								1	0	10	4.030
				3	0	0	11.173	2	0	0.144	12.350
								3	0	0	27.965
								4	0	0	58.758
								1	0	10	16.086
				4	0	0	23.222	2	0	0	24.271
								3	0	0	40.016
								4	0	0	70.383
p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}
0	10	20	1	1/3	50	20	4/3	1	50	20	7/6

Table 5.4: The first three time stages of the `SOCP` and `SDP` solution for the test case, with 5 solar panels per bus and $PL = 50$. p is given in [kW/4] and x in kWh.

2. an infinite battery capacity for all buses, allowing the solution to get rid of extra power in case of oversupply.

The first condition is easily satisfied, even though PV distorts the solution, as discussed in the experimental discussion. For the second one, the value 13.5 kW h taken from [83] suffices for the considered test case. However, when experimentally checking the exactness of the relaxation, we observed that the slack of the `SOCP` constraint was rising when $\omega_t = 3, 4$ and $\omega_{t-1} = 3, 4$. In other words, it happened in case of consecutive sunny time periods, i.e. oversupply. It is actually for the exact same reason as before: get rid of extra power by artificially creating resistive power losses. Except that here, it is for *reactive* power. Indeed, batteries can only store real power, so the problem remains for reactive oversupply. In order to address that issue, we introduced the possibility for the solution to get rid of extra reactive power by relaxing the reactive power balance equality into an inequality.

Semi-Definite Program

A sufficient condition for the tightness of the `SDP` relaxation was given in section 3.4.2 and can be summarized as

1. a constraint involving the maximum injection power of each bus and impedance, difficult to interpret but Zhang et al. showed that it is expected to be satisfied by most networks [48];
2. $PL > 0$, otherwise the solution could artificially increase the current to create resistive power losses in case of oversupply, just like in the `SOCP` case;
3. a lower bound on $\overline{qD}_i$ for each bus i, for the same reason as cited above: ensuring that reactive power can be shed in case of oversupply.

First, we check that the constraint on the network is satisfied by our test case [4]. Then, the second condition is discussed in the next section for both SOCP and SDP relaxations. Eventually, the lower bounds of the last condition can be computed using the expression provided by Theorem 2, given in section 3.4.2. This yields

$$\overline{qD}_0 > -0.0518 \qquad \overline{qD}_{11} > 0.1389 \qquad \overline{qD}_i > 0 \quad \forall\, i,$$

which would be fullfilled if there was distributed generation. However, from a certain number of solar panels (e.g. 5), in case of bright weather the loads become negative: the solar production is larger than the demand. The same issue as the SOCP thus arises: an oversupply of reactive power, and no possibility to shed the surplus. Similarly to the SOCP case, we relax the reactive power balance equality into an inequality to correct that issue.

The theoretical conditions of these two convex relaxations share many similarities. This particularity will be experimentally confirmed in the following discussion.

5.3.2 Experimental Discussion

Our test case satisfies the theoretical conditions, so both relaxations should be tight. In the present section, we experimentally discuss the influence of two parameters on their tightness: PV, the number of solar panels per bus and PL, the weighting factor of resistive power losses in the objective. In order to assess tightness, the following criteria are used

$$\text{slack}_{i,t,\omega_{[t]}} = \ell_{i,t,\omega_{[t]}} - \frac{P^2_{i,t,\omega_{[t]}} + Q^2_{i,t,\omega_{[t]}}}{v_{i,t,\omega_{[t]}}} \qquad \text{ratio}_{i,t,\omega_{[t]}} = \left| \frac{\lambda_{2,i,t,\omega_{[t]}}}{\lambda_{1,i,t,\omega_{[t]}}} \right|,$$

where $\lambda_{j,i,t,\omega_{[t]}}$ is the jth largest eigenvalue of $\begin{bmatrix} \mathbf{V}_{i,t,\omega_{[t]}} & \mathbf{S}_{i,t,\omega_{[t]}} \\ \mathbf{S}^H_{i,t,\omega_{[t]}} & \mathbf{I}_{i,t,\omega_{[t]}} \end{bmatrix}$. If $\text{slack}_{i,t,\omega_{[t]}} = 0 \ \forall\ i,t,\omega_{[t]}$, the SOCP relaxation is exact. Similarly, a rank-1 solution to the SDP relaxation is ensured if $\text{ratio}_{i,t,\omega_{[t]}} = 0 \ \forall\ i,t,\omega_{[t]}$. However, due to finite numerical precision, these criteria will never be exactly equal to 0 even if the relaxation is tight. Therefore, we rely on their maximum and mean, taken over all $i,t,\omega_{[t]}$. The smaller the maximum and mean, the more tight is the relaxation.

We first study the influence of the number of solar panels, PV, on tightness. Note that this parameter actually stands for the magnitude of uncertainty in our model: the greater PV, the more stochastic is the problem. From Figure 5.5, we observe that tightness worsens while increasing PV. It should be stressed that the solutions presented on this Figure do not benefit from the possibility to shed reactive power, i.e. the reactive power balance has not been relaxed to an inequality. Therefore, a larger PV yields more serious reactive oversupply in case of good weather, forcing the SOCP and SDP solutions to artificially generate reactive power losses, and move away from tightness. In order to address that issue, the reactive power balance will be relaxed to an inequality for the following tests. In addition, all the following numerical tests will be conducted with $PV = 1$ and 5, respectively standing for low and high uncertainty.

Then, we address the influence of the weighting factor of resistive power losses in the objective, PL, on tightness. As shown in Figure 5.6, tightness improves while increasing PL. However, theoretical conditions state that a strictly positive value of PL should ensure tightness, no matter its value. This is due to limited numerical precision: a too low PL does not sufficiently penalizes untight solutions to the eye of the IPM solver. This numerical behavior appears at its worst for the SDP case with 5 solar panels per bus, where $PL = 1000$ is needed to ensure a ratio of 10^{-2} for all matrices.

[4] For bus 0, $p_{\max} = 10$ yields $-1.5625 < \bar{\theta}_0 = 0.1202 < 1.5625$. For bus 11, we have $-0.9628 < \bar{\theta}_{11} = 0.0044 < 0.9628$ and for the other buses i, $p_{\max,i} = 0$ so $\bar{\theta}_i = 0$ and $-\tan^{-1}\left(\frac{B_i}{G_i}\right) < \bar{\theta}_i = 0 < \tan^{-1}\left(\frac{B_i}{G_i}\right)$ is satisfied since $B_i > 0 \quad \forall\, i$.

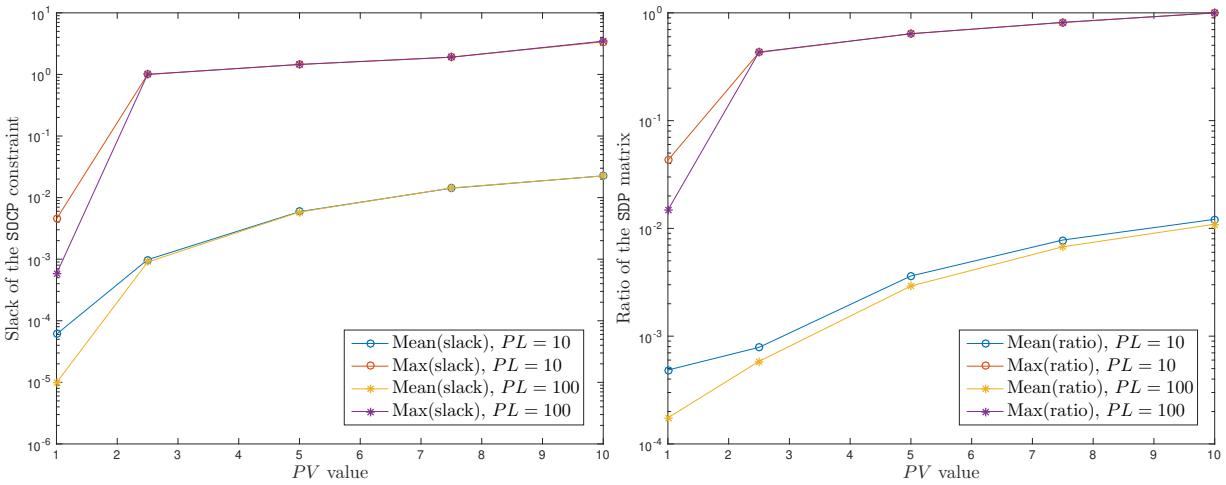


Figure 5.5: Study of the influence of PV on tightness of convex relaxations. The left figure illustrates the SOCP relaxation, while the right figure considers the SDP relaxation. Results are presented for $PL = 10$ and 100, without the possibility to shed reactive power.

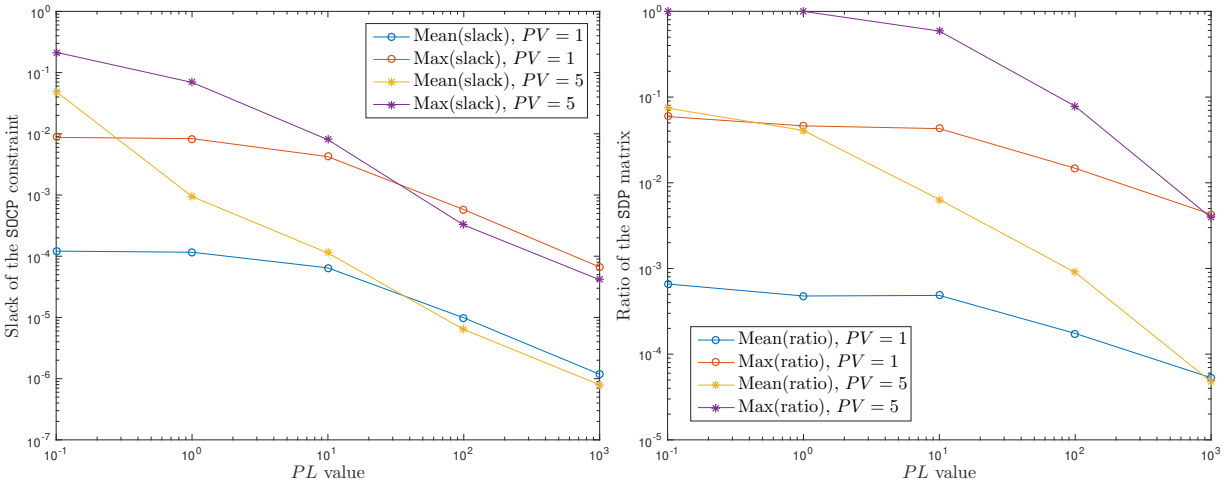


Figure 5.6: Study of the influence of PL on tightness of convex relaxations. The left figure illustrates the SOCP relaxation, while the right figure considers the SDP relaxation. Results are presented for $PV = 1$ and 5, with the possibility to shed reactive power.

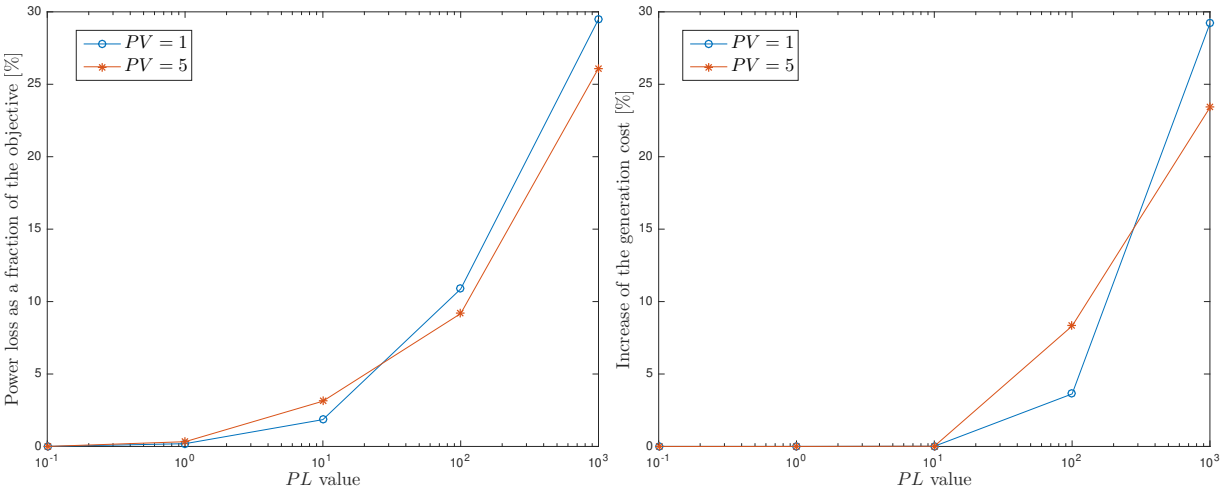


Figure 5.7: Study of the influence of PL on the objective of convex relaxations. The left figure illustrates the evolution of the power loss fraction of the total objective, while the right figure considers the increase of generation cost as a fraction of the initial one, i.e. $(GC_{PL} - GC_0)/GC_0$, where GC_{PL} stands for the generation cost with PL. Results are presented for $PV = 1$ and 5, with the possibility to shed reactive power.

However, PL modifies the objective, and therefore the solution. With a too large value of PL, the problem shifts from optimizing generation cost to minimizing power losses. The resulting solution buys less power at the market off-peak price (at $t = 1$) since it would have implied larger power losses in order to store it in the network batteries, as discussed in section 5.2.3. Figure 5.7 shows that if $PL = 1000$, power losses account for up to 30 % of the objective. By adjusting to this shifted objective, the optimal generation cost increases by 30 %. Note that in Figure 5.7, the curves represent both `SOCP` and `SDP` relaxations. Indeed, they provide the same solution since the considered network is assumed to be balanced, as discussed in section 3.1.3.

As a result, we end up with a trade-off to choose the value of PL. On one side, it improves tightness, but on the other side, it distorts the solution. After examining Figures 5.6 and 5.7, we opt for $PL = 100$ for the ensuing numerical tests. In this way, the worst-case `SDP` ratio is 0.1 but 10^{-3} on average, and the `SOCP` slack is always lower than 10^{-3}. Besides, the increase of generation cost does not exceed 10 %, thus the solution is not considerably modified.

5.4 Computational Results

In the present section, the `SDDP` algorithm is run on the three formulations of the test case introduced in section 5.2. We discuss the number of iterations and time of convergence of the algorithm, and the influence of K, the number of Monte Carlo simulations performed at each iteration. As mentioned in the previous section, the tests are run with $PL = 100$ and $PV = 1, 5$. The choice of K is an important feature of the design of the algorithm, and a relevant concern is to find the optimal $K^\star$ that minimizes the time of convergence. Interestingly, $K^\star$ depends on PV and it can be intuitively understood, as explained at the end of this section.

First, we must choose the termination criterion of the algorithm. Usually, `SDDP` is terminated when it reaches a confidence interval based on an estimation of the optimal objective. However, here, since the test case is not too large, we can solve it in its **Extensive form** (i.e. solve the whole problem at once, as a unique optimization problem), and obtain the exact optimal solution. Actually, it is what has been done in the two previous sections. Therefore, we will stop the

algorithm when it reaches an objective *sufficiently* close to the optimum $f^\star$, i.e. when

$$\frac{f^\star - f}{f^\star} \leq \texttt{tol},$$

where the value of `tol` still needs to be defined. Given the large difference in efficiency of the three considered formulations, different gap tolerances will be chosen. For `SDP`, we settle to `tol` = 2 or 3 %, as usual stochastic unit commitment models [84]. However, `SOCP` and `LP` are solved significantly faster, allowing us to afford gap tolerances of 0.01 % and 0.001 %, respectively.

Figure 5.8 presents a study of the performance of the `SDDP` algorithm on the three formulations, for different values of K. On the left figures, the evolution of the objective value is presented as the algorithm progresses for 4 values of K. A higher number of Monte Carlo simulations per iteration increases the rate of convergence over the number of iterations. Indeed, a larger K accelerates the learning of the value function, resulting in more efficient iterations.

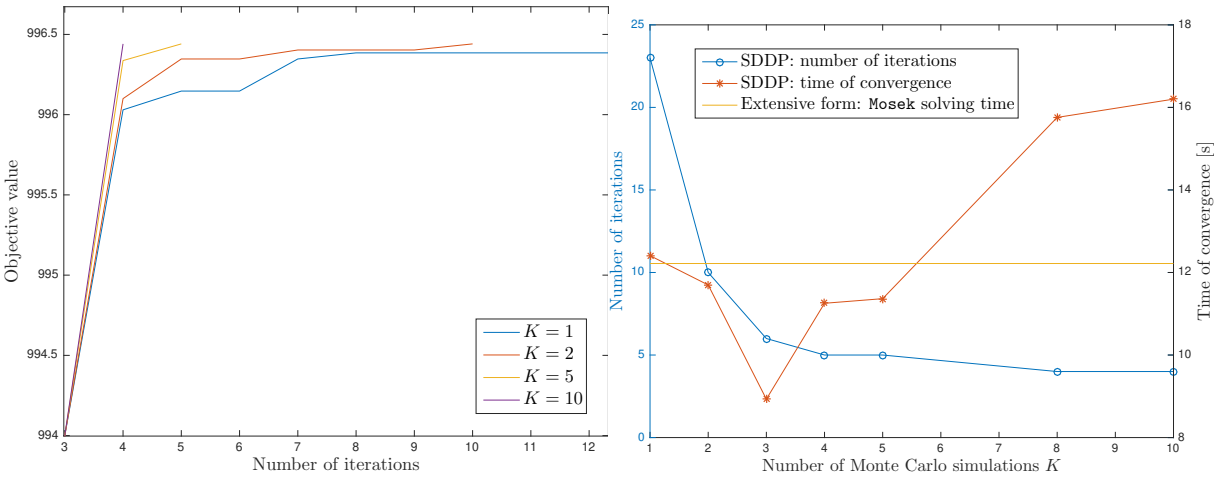
The number of iterations needed to reach the gap tolerance thus decreases with K, as illustrated by the blue curves on the right figures. We can further observe that this decrease seems to follow an exponential shape. However, when K is larger, each iteration takes more time. Indeed the forward pass requires to solve $1 + K(H-2) = 3K + 1$ problems, and $K \cdot \sum_{t=2}^{H} |\Omega_t| = 16K$ problems are solved during the backward pass. Thus, each iteration requires to solve $19K + 1$ problems.

Therefore, the choice of K involves a trade-off: a larger value ensures a lower number of iterations before convergence, but at the cost of increasing the number of problems that need to be solved at each iteration. In addition, a larger K increases the size of the NLDS subproblems more rapidly because of a greater number of cuts [66]. This trade-off is well depicted on Figure 5.8b: when K changes from 2 to 3, the number of iterations drop from 10 to 6. Gaining 4 iterations is worth charging each iteration with 19 additional linear problems to solve, so the time of convergence drops to 8.93 s. However, when $K = 4$ we only save one iteration, so the time of convergence increases to 11.25 s. In this case, the optimal number of Monte Carlo is given by $K^\star = 3$.

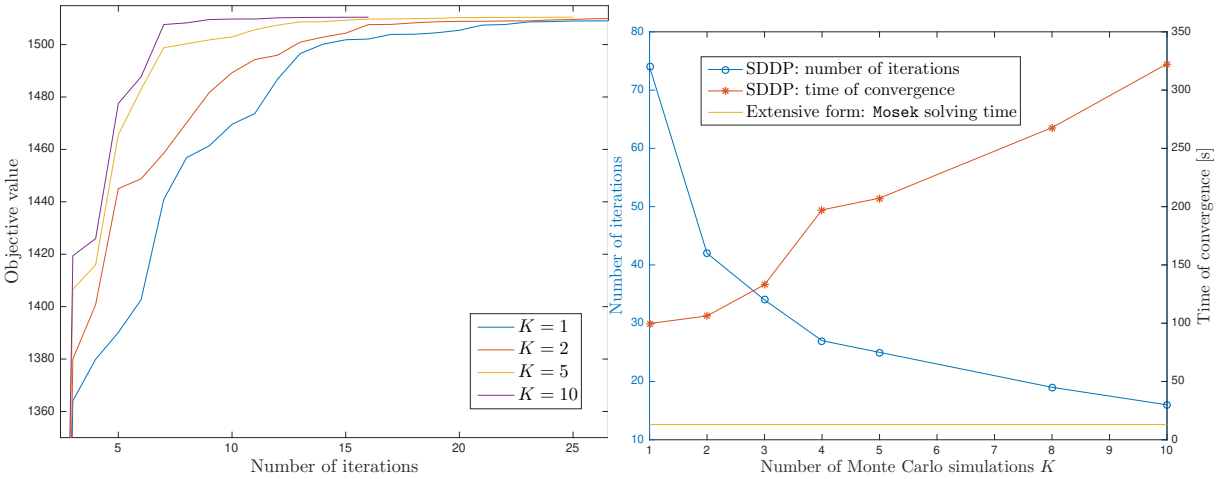
Nevertheless, this optimal choice of K does not extend to `SOCP` and `SDP`. Conic programs require a more important computational effort to solve, and the iterations saved by a larger K do not balance the $19K + 1$ additional conic programs that must be solved at each iteration. As a result, we observe on Figures 5.8d and 5.8f that the optimal number of Monte Carlo is given by $K^\star = 1$ for `SOCP` and `SDP` when $PV = 1$.

As a comparison, the time needed for `Mosek` to solve the whole problem at once (i.e. the extensive form) is also represented on the right figures. This allows us to assess whether the implementation of `SDDP` is worth the computational effort gained. For $K_{\texttt{LP}} \in [2, 5]$ and $K_{\texttt{SDP}} \in \{1, 2\}$, `SDDP` outperforms the extensive optimization. Although we do not save an important amount of time, it should be recalled that the considered test case is a small problem. On practical instances, this saving could be more significant, especially if we parallelize the algorithm.

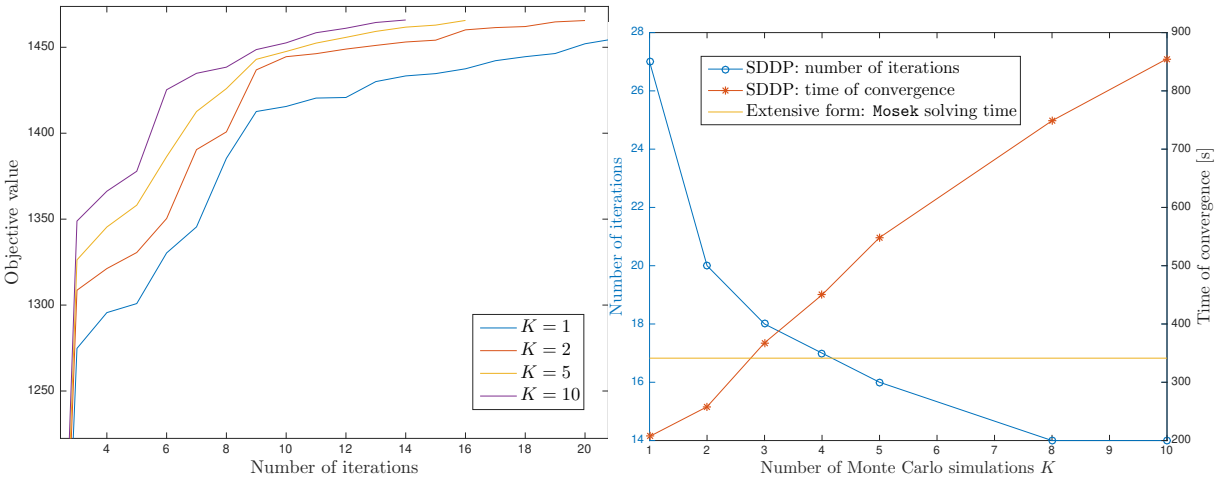
Furthermore, we ran the same tests with $PV = 5$ instead of 1. It yields a problem with a larger influence of uncertainty, and we observe in Figure 5.9 that it affects significantly the optimal choice of K. When $PV = 1$, cursory simulations in the forward pass were sufficient to achieve the prescribed gap tolerance, leading to $K^\star_{\texttt{LP}} = 3$ and $K^\star_{\texttt{SOCP}} = K^\star_{\texttt{SDP}} = 1$. But with $PV = 5$, a more accurate representation of uncertainty is required since it has more weight. In particular, the forward passes do not often simulate the rainy and full-sun realizations of uncertainty, i.e. $\omega = 1, 4$. However, they have an important influence since they correspond to either none or full solar generation. More simulations per forward pass help to capture the uncertainty model more accurately, and therefore reach convergence faster.



(a) LP, tol = 0.001 % (b) LP, tol = 0.001 %

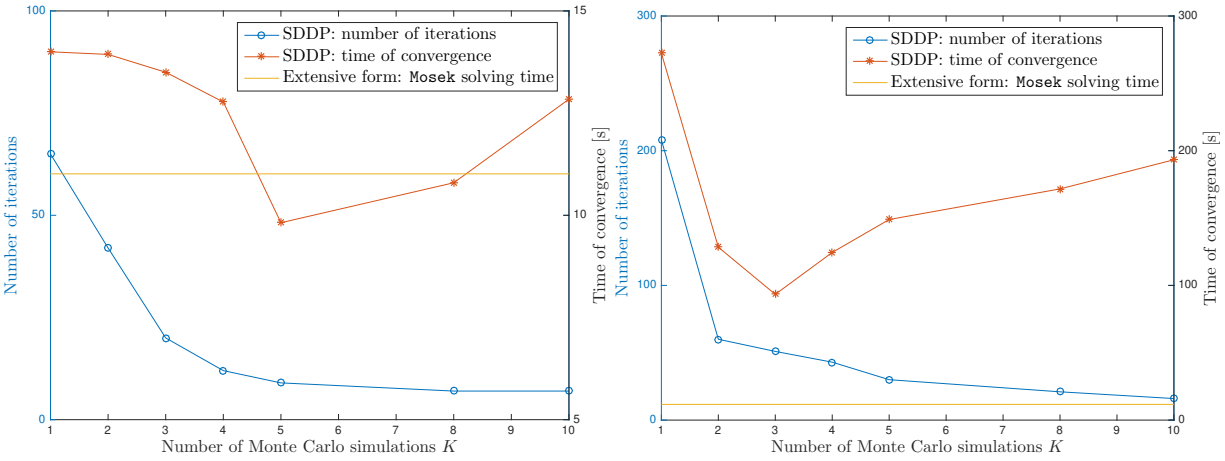


(c) SOCP, tol = 0.01 % (d) SOCP, tol = 0.01 %



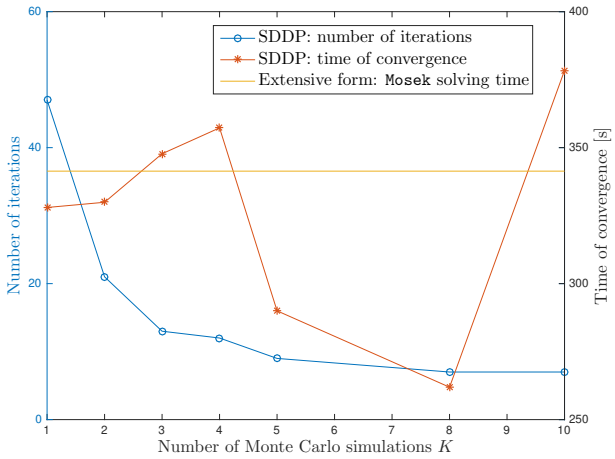
(e) SDP, tol = 3 % (f) SDP, tol = 3 %

Figure 5.8: Study of the performance of the SDDP algorithm on the three formulations, for different values of K. The left figures show the evolution of the objective as the algorithm progresses, while the right figures consider the number of iterations and time of convergence for different values of K. As a comparison, the time needed for Mosek to solve the whole problem at once (i.e. the extensive form) is also represented. These tests are performed for $PL = 100$ and $PV = 1$.



(a) LP, tol = 0.001 %

(b) SOCP, tol = 0.01 %



(c) SDP, tol = 2 %

Figure 5.9: Study of the performance of the SDDP algorithm on the three formulations, for different values of K. The number of iterations and time of convergence for different values of K are presented. As a comparison, the time needed for Mosek to solve the whole problem at once (i.e. the extensive form) is also considered. These tests are performed for $PL = 100$ and $PV = 5$.

In this way, it yields $K^\star_{\texttt{LP}} = 5, K^\star_{\texttt{SOCP}} = 3$ and $K^\star_{\texttt{SDP}} \in (5, 10)$ when $PV = 5$. Concerning Figure 5.9c, up to $K = 4$ the decrease of iterations does not balance the additional time required by each iteration. But when $K = 5$, the number of iterations drops from 12 to 9, and it pays off regarding the time of convergence. However, $K = 8$ manages to save 2 more iterations and decreases again the time of convergence, although 8 Monte Carlo simulations are performed at each forward pass.

5.5 Network Imbalance

5.5.1 Unbalanced Test Case

The present section studies the case of an unbalanced network. In order to be able to compare the unbalanced solution with the balanced one, we consider and unbalanced version of the test case examined so far. To this end, we connect each load to only one phase, like it is the case in real distribution networks. The phase assigned to each load is illustrated in Figure 5.10, and the

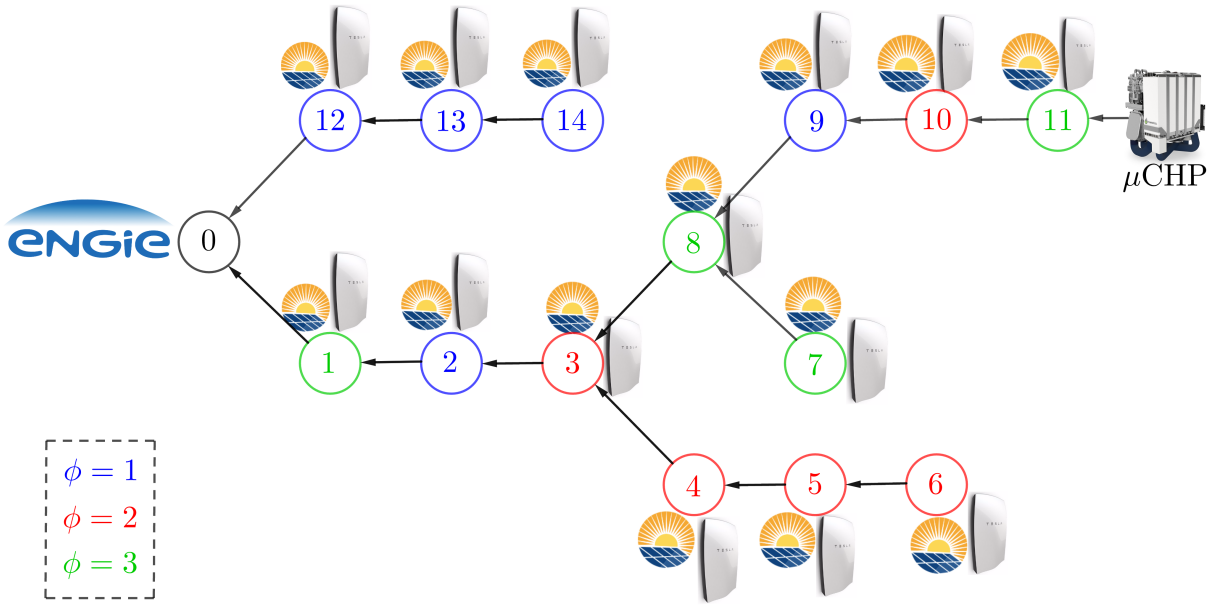


Figure 5.10: Unbalanced test case network. Each color corresponds to a different phase, while black stands for the three phases. The micro CHP and power substation are connected to the three phases.

geometry of this attribution is inspired from real data [85].

5.5.2 Unbalanced Solution

On this unbalanced setup, even though we have been working on the same network, no solution computed so far can be used. For example, power cannot be exchanged between loads 1, 2 and 3 even though they are connected next to each other, because they are connected to three different phases. Therefore, in order to operate this network efficiently, the single-phase formulations are of no use, and we have no other choice than resort to the semi-definite formulation.

Apart from connecting the loads to single phases instead of the three of them, no modification needs to be performed on the `SDP` formulation and implementation. The solution is presented in Table 5.5 as the sum on the three phases. In this way, it can be compared to the balanced solutions previously presented in the same format. In addition, solution for each different phase is given in appendix B.

This solution reaches a generation cost of 972.026 and power losses of 283.06, for a total optimal objective of 1255.086. This can be compared to the balanced conic solution of section 5.2.3, that achieved a generation cost of 659.861 and power losses of 58.875, for a total optimal objective of 718.736. Apart from the network imbalance, these two solutions perform on the exact same network ($PV = 5$, $PL = 50$).

First, it can be stressed that network imbalance is costly to the solution: the optimal objective increases by 74.62 %. Yet, by looking at the two components of the objective, we observe that this increase is mostly due to power losses: they increase by 380.78 % while generation cost increases by 47.30 %. It can be explained by the fact that power losses increase quadratically with current, and thus flowing power, since they are given by $PL \cdot \sum_{i \in \mathcal{E}} R_i \cdot |I_i|^2$. Therefore, in the balanced setup, power is evenly shared through the three phases in order to minimize power losses. However, when for example bus 6 needs to be supplied in the unbalanced case, all the needed power, and therefore current, has to be routed through the second phase. This results in

$t=1$				$t=2$				$t=3$			
ω	$\sum p_0^\phi$	$\sum p_{11}^\phi$	$\sum\sum x_i^\phi$	ω	$\sum p_0^\phi$	$\sum p_{11}^\phi$	$\sum\sum x_i^\phi$	ω	$\sum p_0^\phi$	$\sum p_{11}^\phi$	$\sum\sum x_i^\phi$
				1	0.518	9.617	14.309	1	15.319	9.965	19.980
								2	2.400	7.673	23.144
								3	0.039	6.230	35.587
								4	0.142	3.697	62.320
				2	0.009	6.461	16.376	1	5.662	9.976	13.825
								2	0.013	5.846	22.063
								3	0.074	1.016	33.479
1	36.693	6.870	24.917					4	0.078	0.196	62.159
				3	0.051	3.649	19.627	1	4.314	9.974	15.552
								2	0.051	4.904	24.307
								3	0.074	0.273	35.989
								4	0.080	0.175	65.267
				4	0.079	6.075	33.216	1	8.229	9.925	31.806
								2	0.026	6.264	38.369
								3	0.029	1.649	49.990
								4	0.074	1.177	78.271
p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}
0	10	20	1	1/3	50	20	4/3	1	50	20	7/6

Table 5.5: The first three time stages of the SDP solution for the unbalanced test case, with 5 solar panels per bus and $PL = 50$. p is given in [kW/4] and x in kWh. In this table, the sum of the three phases is presented.

more important power losses when network imbalance is taken into account.

At last, Tables 5.4 and 5.5 allow us to compare the influence of network imbalance on the optimal operation. The unbalanced solution seems to follow the same heuristic as the balanced one, taking advantage of cheap power sources to fill the batteries. However, more power is bought at each time stage, and the batteries are filled more generously. Once again, it can be attributed to the larger power losses: since more power is wasted because of its circulation, more power has to be generated and stored for the ensuing time periods.

5.5.3 Exactness

Finally, we can conclude this study on network imbalance by testing the exactness of SDP when applied on an unbalanced setup. As mentioned in section 3.4.2, no theoretical condition ensuring tightness is known in case of imbalance. However, Dall'Anese et al. still expect a rank-1 solution even in the unbalanced setup, and support their intuitive claim with a 2-bus and a 3-bus example [16].

The mean and maximal ratios of the unbalanced SDP are presented and compared with the balanced case in Figure 5.11, for $PV = 1$ and various values of PL. We observe that imbalance worsens significantly tightness: a factor 10^2 increases the mean of ratios while the maximal ratio, although already large, reaches 0.23 and almost ignores the variation of PL.

5.6 Discussion

After observing all these computational results, some conclusions can be drawn.

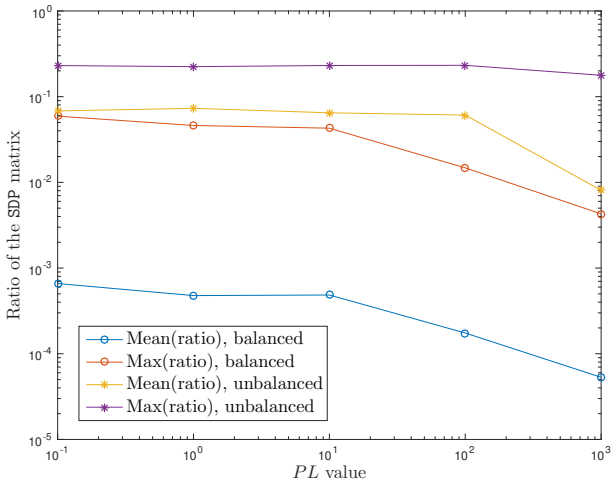


Figure 5.11: Study of the influence of imbalance on tightness of SDP, for various values of PL. Results are presented for $PV = 1$, with the possibility to shed reactive power.

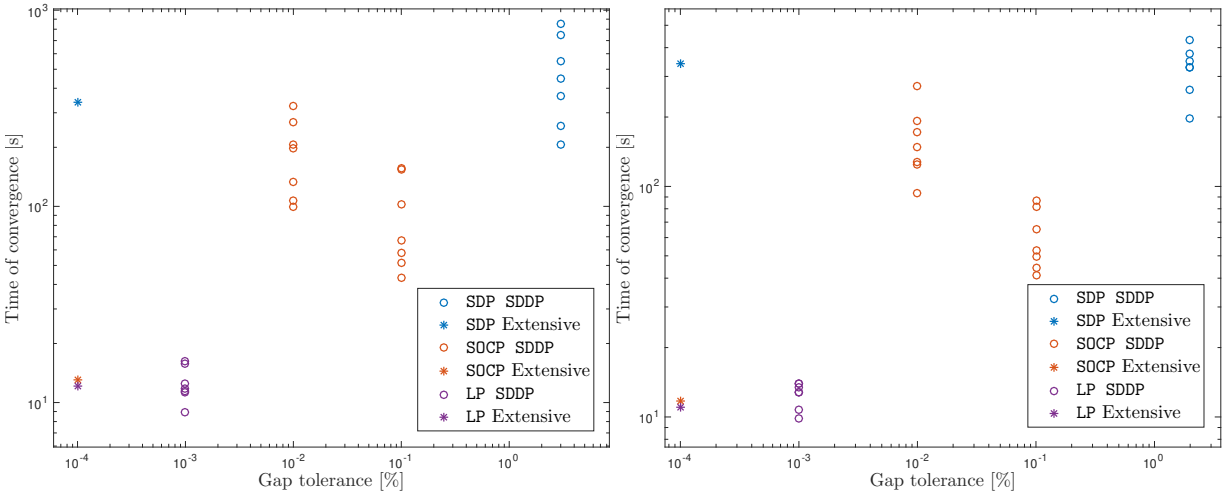
First of all, in addition to providing the same solution when tight, the second-order cone and semi-definite programs undergo the same issue in case of oversupply. The solution tends to shed real and reactive power by creating artificial resistive power losses in order to satisfy the power balance. In both formulations, this behavior is detected by the tightness criterion (i.e. the slack of the SOCP inequality or the eigenvalue ratio of the SDP matrix) and leads to an untight solution. In order to ensure the absence of that behavior, theoretical conditions are provided for SOCP [6] and SDP [48]. Although these conditions are derived in different contexts and formulated in two distinct manners, they appear to involve noticeably similar consequences, and translate into the same conditions on the considered test case.

Moreover, the efficiency properties of the three studied formulations have not be compared between each other yet. In order to remedy this, we plotted the time of convergence of the three problems, for all tested values of K and for $PV = 1$ and 5 in Figure 5.12. Since the same gap tolerance required for convergence has not been applied on the programs, it is represented by the abscissae. Thus, the lower left corner of the figure indicates high efficiency while the upper right corner stands for low efficiency.

As we would expect, the linear program is undoubtedly the most efficient, since it reaches the lowest gap in the shortest time. However, as stressed before, this formulation does not account for Kirchhoff's laws, the well-known nonlinear rules governing power flows. For a more detailed formulation, one must resort to SOCP and SDP. Although these two convex problems provide the same solution, SOCP is significantly more efficient. It was not obvious with the time needed to reach a 0.01 % gap tolerance, so we reran it for a 0.1 % gap. Note that solving the whole SOCP at once (i.e. its extensive form) outperforms all SDDP executions, but parallelization would rectify this on a larger instance.

However, in low voltage networks, different houses are connected to each phase, which results in network imbalance. In this case, LP and SOCP are of no use since they rely on a single-phase model of the optimal power flow. Nevertheless, SDP can be applied on the unbalanced setup without any further implementation effort. Although less efficient than the two others, SDP can be solved in polynomial time by interior-point solvers like Mosek [86].

Furthermore, SDDP outperforms the optimization of the extensive form on our small test case when K is well chosen, as depicted in Figure 5.12. This promises good performance on larger instances, with the support of parallelization. It also stresses that even with the computational effort gained by the implementation of SDDP, the resolution of a small instance already requires



(a) $PV = 1$ (b) $PV = 5$

Figure 5.12: Comparison of the efficiency of the three considered formulations. The presented results were obtained for $K \in \{1, 2, 3, 4, 5, 8, 10\}$, $PL = 100$ and $PV = 1$ (left panel) or 5 (right panel). In the legend, "Extensive" stands for the time required by `Mosek` to solve the whole problem (i.e. its extensive form) at once.

a non-negligible amount of time. Therefore, it confirms the relevance of a distributed approach to operate distribution networks.

Nonetheless, network imbalance seems to worsen tightness as observed on the test case in the previous section. Actually, no theoretical condition ensuring tightness is known in case of imbalance although Dall'Anese et al. still expect a rank-1 solution [16]. This question could be further studied in future works. Otherwise, we could resort to an approach similar to the one proposed in [58]. In this publication, the authors recover a feasible power flow solution from an inexact one, provided by an untight `SOCP` model. Even though their method relies on a single-phase model of the `OPF`, a similar approach could be attempted in the unbalanced setup.

5.7 Further Improvements

5.7.1 Test Case

The test case considered in this document was basic and small in order to ensure a physical understanding of the solution and tractability on a personal computer. However, future works could address more detailed and larger instances in order to study the application of `SDDP` on `SOCP` and `SDP` more practically.

First of all, future test cases could cover more time stages: one per hour would be good. Then, loads could differ from each other, instead of having the same load at each bus. Also, a more detailed model of uncertainty would be more practical: real solar data could be sampled to this end. In addition, one could account for shunt susceptance and conductance effects in the model, while they were neglected for the test case. Moreover, larger networks could be considered. Eventually, in order to model more accurately unbalanced networks, non-diagonal impedance matrices could be used. All these improvements would result in a more practical review of the studied method, but would require more significant computational resources.

5.7.2 Stochastic Dual Dynamic Programming

Concerning the algorithm SDDP itself, an important improvement would be to add feasibility cuts. Indeed, for the considered test case, they were not needed since there was no bound on the power injected at bus 0. In order to verify this, we implemented a script which checked that each NLDS was feasible without any battery to charge or discharge from. But if all power injections are bounded, or under some technical constraints, subproblems could reach infeasibility and feasibility cuts would be required. We implemented them on LP and tried to extend them on the conic formulations, but without success: the dual values returned by Mosek were not practical. We also tried to avoid that issue by adding slack variables instead, and a penalty term in the objective. Nevertheless, these slack variables lead to a bad numerical behaviour of Mosek, and we had to abandon the possibility to take infeasibilities into account in our implementation. However, we remain confident about the feasibility of implementing feasibility cuts on the conic formulations.

Besides, it can be stressed that the number of Monte Carlo simulations, K, could change over iterations of the algorithm to accelerate the convergence of the algorithm, while we restricted ourselves to constant values of K in this document. "For example, the parameter may increase over iterations, since early iterations will anyways lead to short-sighted trial decisions, and there is little gain from exploring the value function in the neighborhood of such poor decisions" [66].

In addition, the performance of SDDP could be improved by using a solver that can be *warm-started*, i.e. use the last solution of a NLDS as a starting iterate. Although the NLDS has been changed by cuts and/or different trial decisions, it remains quite similar and warm start could significantly decrease the time of convergence, especially for SDP. However, Mosek, which is one of the best SDP solvers today, cannot be warm-started. On the other hand, SCS solves SDPs using ADMM and can be warm-started.

Moreover, since SDDP relies on random Monte Carlo simulations, the algorithm can converge faster or slower over different executions with the same parameter K. Hence, the algorithm should be executed several times for each K in order to have a significant sample of the outcomes. The discussion of section 5.4 could then be performed on a statistical version of Figures 5.8 and 5.9. Unfortunately, we did not have the time to perform such statistical study, but we believe that the main consequences drawn in this discussion are relevant.

At last, SDDP can be accelerated by sorting the optimality and feasibility cuts generated through the iterations, and only keeping the "active" ones. Indeed, the algorithm generates a significant amount of cuts, and as it progresses, some former cuts may be deprecated and slow down the resolution of the subproblems at each iteration. [87] implements a method that estimates the "trust" of each cut by a decaying function over iterations, which gets a bonus if it used when solving a subproblem. When this trust falls under a threshold, the corresponding cut is pruned.

Chapter 6

Conclusion

6.1 Main Results

This thesis focused on the optimal operation of distribution networks over multiple time periods and under uncertainty caused by renewables, which can be mitigated by the presence of local batteries. This problem is challenging due to the imbalance arising in low-voltage networks, but a recent convex relaxation, SDP, can be applied to manage it. Moreover, semi-definite has been an active research field over the last decade, and modern interior-point solvers as Mosek can solve SDPs in polynomial time. In order to deal with the multistage stochastic character of the problem, the SDDP algorithm, which has found great commercial success in linear programming, is implemented on the nonlinear SOCP and SDP.

In the first chapter of this thesis, a broad literature review is realized on the optimal power flow and its relaxations, which have been extensively studied. Then, the three addressed models are derived from the very beginning, and the relation between their feasibility sets is discussed. After that, the stochastic dual dynamic programming is presented, along with its application on our convex programs. After all, the derived methods are tested on a simple test case and compared by mean of tightness and performance. Some important conclusions were drawn from these results, and can be summarized as follows:

- In addition to provide the same answer when tight, SOCP and SDP relaxations exhibit the same issue of oversupply, and their respective theoretical conditions to ensure tightness translate into the same consequences on the studied test case.
- The SDDP algorithm performs well on SOCP and SDP, and shows promising efficiency performances compared to a commercial solver, especially by recalling that SDDP can be parallelized.
- Network imbalance is shown to imply substantial modifications to the solution, rendering LP and SOCP impractical. Therefore, we must resort to SDP, the least efficient of the considered models. However, it can still be solved in polynomial time.
- Network imbalance worsens tightness of the SDP relaxation, and no theoretical condition ensuring tightness is known in this case.
- This convinces us that optimizing low-voltage networks remains challenging regarding computational resources, which stresses the relevance of a distributed approach.

The three questions asked in the introduction are thus answered by these highlighted results. In this way, this work tries to progress in the broader objective to analyze the possibility of a hierarchical organization of electricity markets, as mentioned in the introduction.

6.2 Future Perspectives

As this thesis lies in the intersection of diverse active research fields, future works on the subject can explore various directions.

First of all, larger and more detailed test cases can be considered, at the cost of more significant computational resources requirements. The algorithm can be applied on diverse larger networks, and over more time stages. In addition, loads and solar data can be sampled at a better precision to model uncertainty more accurately, which would result in a more practical study of the algorithm.

Moreover, an important improvement to the `SDDP` algorithm itself would be to add feasibility cuts to the implementation. The test case considered in this thesis did not suffer from infeasibilities, but it does not reflect practical instances. Although we did not manage to implement feasibility cuts on the conic formulations, we remain confident about the feasibility of this task.

Besides, `SDDP` can be accelerated by increasing the number of Monte Carlo simulations over iterations, since a large K is not valuable in the early iterations. Also, pruning redundant feasibility and optimality cuts might avoid the slowdown caused by a substantial amount of constraints in the late iterations.

In addition, parallelizing an algorithm allows it to execute several instructions simultaneously, and therefore substantially improves its performances. It also allows to solve instances that would not be tractable without parallelism. Thanks to its structure, `SDDP` can be easily parallelized, and it has already been done extensively in the literature. In particular, parallelization could help us to achieve better performances when applying `SDDP` on `SDP`.

As stressed above, network imbalance worsens tightness of the `SDP` relaxation, and no theoretical condition ensuring tightness is known in this case. However, Dall'Anese et al. still expect a rank-1 solution [16]: this question could be studied in future works. Otherwise, one could resort to a feasible-recovering approach like the one proposed in [58] for single-phase networks.

Furthermore, it has been argued by Stephen Boyd and Bill Hogan that the use of `SDDP` for tackling this problem is an overkill, and that one should instead resort to model predictive control. The weakness of model predictive control is that it does not secure the system well against low-probability, catastrophic outcomes. It would therefore be interesting to see whether MPC can be shown to perform significantly worse than stochastic programming for this problem.

Eventually, although the state-of-the-art interior-point solver of `Mosek` has a really good reputation when it comes to convex programming, and rank among the most performant semi-definite solvers [19], other solvers may be tested in order to take advantage of the warm start feature that we implemented, and could potentially significantly accelerate `SDDP` on `SDP`.

* *
*

Bibliography

[1] Anthony Papavasiliou. *Optimization Models in Electricity Markets*.

[2] CREG. Nota over de opvallende evoluties op de belgische groothandelsmarkten voor elektriciteit en aardgas in 2017. `https://www.creg.be/sites/default/files/assets/Publications/Notes/Z1719NL.pdf`. Accessed: 2018-07-31.

[3] Engie. Microgrids to bring electricity to remote rural areas. `https://www.engie.com/en/businesses/microgrids-decentralized-energy/`. Accessed: 2018-07-29.

[4] Caruna. Electricity production and distribution. `http://vuosiraportti2015.caruna.fi/en/report/trustworthy-operator/the-electricity-market-in-finland/`. Accessed: 2018-07-31.

[5] Wikipedia. Three-phase electric power. `https://en.wikipedia.org/wiki/Three-phase_electric_power`. Accessed: 2018-07-31.

[6] Masoud Farivar and Steven H Low. Branch flow model: Relaxations and convexification. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 3672–3679. IEEE, 2012.

[7] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[8] Anthony Papavasiliou. Analysis of distribution locational marginal prices. *IEEE Transactions on Smart Grid*, 2017.

[9] Benjamin Franklin, Peter Collinson, Ebenezer Kinnersley, Guillaume Mazéas, Sir William Watson, David Colden, John Canton, William Maine, John Winthrop, Philip Syng, et al. *Experiments and observations on Electricity made at Philadelphia in America*. D. Henry, 1769.

[10] Cyril Dion. *Petit manuel de résistance contemporaine*. Actes Sud, domaine du possible edition, 2018.

[11] ACEEE. Combined heat and power (chp). `http://aceee.org/topics/combined-heat-and-power-chp`. Accessed: 2018-07-31.

[12] European Commission. Renewable energy: Moving towards a low carbon economy. `https://ec.europa.eu/energy/en/topics/renewable-energy`. Accessed: 2018-07-31.

[13] Damien Ernst. Smart grids versus microgrids. `https://fr.slideshare.net/damienernst5/smart-grids-versus-microgrids`. Accessed: 2018-07-31.

[14] Ignacio Pérez-Arriaga and Christopher Knittel. Utility of the future. *MIT Energy Initiative*, December 2016.

[15] John Farrell. Institute for local self-reliance. https://ilsr.org/john-farrell-talks-energy-decentralization-thom-hartmanns-big-picture/. Accessed: 2018-07-31.

[16] Emiliano Dall'Anese, Hao Zhu, and Georgios B Giannakis. Distributed optimal power flow for smart microgrids. *IEEE Trans. Smart Grid*, 4(3):1464–1475, 2013.

[17] William H Kersting. *Distribution system modeling and analysis*. CRC press, 2001.

[18] Jianfeng Liu, Anya Castillo, Jean-Paul Watson, and Carl D Laird. Global solution strategies for the network-constrained unit commitment (ncuc) problem with nonlinear ac transmission models. *Available on Optimization Online*, 2017.

[19] Hans D. Mittelmann. Several sdp-codes on sparse and other sdp problems. http://plato.asu.edu/ftp/sparse_sdp.html. Accessed: 2018-08-01.

[20] Mosek. Optimization done right. https://www.mosek.com/. Accessed: 2018-08-01.

[21] Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, 52(1-3):359–375, 1991.

[22] Hans D. Mittelmann. Benchmarks for optimization software. http://plato.asu.edu/bench.html. Accessed: 2018-08-01.

[23] Joshua Adam Taylor. *Convex optimization of power systems*. Cambridge University Press, 2015.

[24] J Carpentier. Contribution a l'etude du dispatching economique. *Bulletin de la Societe Francaise des Electriciens*, 3(1):431–447, 1962.

[25] M Huneault and FD Galiana. A survey of the optimal power flow literature. *IEEE transactions on Power Systems*, 6(2):762–770, 1991.

[26] James A Momoh, Rambabu Adapa, and ME El-Hawary. A review of selected optimal power flow literature to 1993. i. nonlinear and quadratic programming approaches. *IEEE transactions on power systems*, 14(1):96–104, 1999.

[27] James A Momoh, ME El-Hawary, and Ramababu Adapa. A review of selected optimal power flow literature to 1993. ii. newton, linear programming and interior point methods. *IEEE Transactions on Power Systems*, 14(1):105–111, 1999.

[28] Hua Wei, Hiroshi Sasaki, Junji Kubokawa, and R Yokoyama. An interior point nonlinear programming for optimal power flow problems with a novel data structure. *IEEE Transactions on Power Systems*, 13(3):870–877, 1998.

[29] Brian Stott and Of Alsac. Fast decoupled load flow. *IEEE transactions on power apparatus and systems*, (3):859–869, 1974.

[30] O Alsac, J Bright, M Prais, and B Stott. Further developments in lp-based optimal power flow. *IEEE Transactions on Power Systems*, 5(3):697–711, 1990.

[31] Konrad Purchala, Leonardo Meeus, Daniel Van Dommelen, and Ronnie Belmans. Usefulness of dc power flow for active power flow analysis. In *Power Engineering Society General Meeting, 2005. IEEE*, pages 454–459. IEEE, 2005.

[32] Brian Stott, Jorge Jardim, and Ongun Alsaç. Dc power flow revisited. *IEEE Transactions on Power Systems*, 24(3):1290–1300, 2009.

[33] Steven H Low. Convex relaxation of optimal power flow—part i: Formulations and equivalence. *IEEE Transactions on Control of Network Systems*, 1(1):15–27, 2014.

[34] Rabih A Jabr. Radial distribution load flow using conic programming. *IEEE transactions on power systems*, 21(3):1458–1459, 2006.

[35] Joshua Adam Taylor. *Conic optimization of electric power systems*. PhD thesis, Massachusetts Institute of Technology, 2011.

[36] Joshua A Taylor and Franz S Hover. Convex models of distribution system reconfiguration. *IEEE Transactions on Power Systems*, 27(3):1407–1413, 2012.

[37] Mesut E Baran and Felix F Wu. Optimal capacitor placement on radial distribution systems. *IEEE Transactions on power Delivery*, 4(1):725–734, 1989.

[38] Mesut Baran and Felix F Wu. Optimal sizing of capacitors placed on a radial distribution system. *IEEE Transactions on power Delivery*, 4(1):735–743, 1989.

[39] Na Li, Lijun Chen, and Steven H Low. Demand response in radial distribution networks: Distributed algorithm. In *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*, pages 1549–1553. IEEE, 2012.

[40] Qiuyu Peng and Steven H Low. Distributed algorithm for optimal power flow on a radial network. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 167–172. IEEE, 2014.

[41] Céline Gérard. *Coordinated operation of electric power transmission and distribution systems*. PhD thesis, Université Catholique de Louvain, 2017.

[42] Burak Kocuk, Santanu S. Dey, and X. Andy Sun. Strong socp relaxations for the optimal power flow problem. *Operations Research*, 64(6):1177–1196, 2016.

[43] Xiaoqing Bai, Hua Wei, Katsuki Fujisawa, and Yong Wang. Semidefinite programming for optimal power flow problems. *International Journal of Electrical Power & Energy Systems*, 30(6-7):383–392, 2008.

[44] Ilan Adler and Farid Alizadeh. *Primal-dual interior point algorithms for convex quadratically constrained and semidefinite optimization problems*. Rutgers University. Rutgers Center for Operations Research [RUTCOR], 1995.

[45] Michael J Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[46] Javad Lavaei and Steven H Low. Zero duality gap in optimal power flow problem. *IEEE Transactions on Power Systems*, 27(1):92, 2012.

[47] Javad Lavaei, David Tse, and Baosen Zhang. Geometry of power flows in tree networks. In *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–8. IEEE, 2012.

[48] Baosen Zhang, Albert YS Lam, Alejandro D Domínguez-García, and David Tse. An optimal and distributed method for voltage regulation in power distribution systems. *IEEE Transactions on Power Systems*, 30(4):1714–1726, 2015.

[49] Daniel K Molzahn and Ian A Hiskens. Convex relaxations of optimal power flow problems: An illustrative example. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(5):650–660, 2016.

[50] H Kersting William et al. Distribution system modeling and analysis. *Ed.: CRC Press, USA*, 2002.

[51] Zhi-Quan Luo, Wing-Kin Ma, Anthony Man-Cho So, Yinyu Ye, and Shuzhong Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, 2010.

[52] Lingwen Gan and Steven H Low. Convex relaxations and linear approximation for optimal power flow in multiphase radial networks. In *Power Systems Computation Conference (PSCC), 2014*, pages 1–9. IEEE, 2014.

[53] Qiuyu Peng and Steven H Low. Distributed algorithm for optimal power flow on an unbalanced radial network. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 6915–6920. IEEE, 2015.

[54] Daniel K Molzahn, Jesse T Holzer, Bernard C Lesieutre, and Christopher L DeMarco. Implementation of a large-scale optimal power flow solver based on semidefinite programming. *IEEE Transactions on Power Systems*, 28(4):3987–3998, 2013.

[55] Subhonmesh Bose, Steven H Low, Thanchanok Teeraratkul, and Babak Hassibi. Equivalent relaxations of optimal power flow. *IEEE Transactions on Automatic Control*, 60(3):729–742, 2015.

[56] Steven H Low. Convex relaxation of optimal power flow—part i: Formulations and equivalence. *IEEE Transactions on Control of Network Systems*, 1(1):15–27, 2014.

[57] Steven H Low. Convex relaxation of optimal power flow—part ii: Exactness. *IEEE Transactions on Control of Network Systems*, 1(2):177–189, 2014.

[58] Wei Wei, Jianhui Wang, Na Li, and Shengwei Mei. Optimal power flow of radial networks and its variations: A sequential convex optimization approach. *IEEE Transactions on Smart Grid*, 8(6):2974–2987, 2017.

[59] Balho H Kim and Ross Baldick. Coarse-grained distributed optimal power flow. *IEEE Transactions on Power Systems*, 12(2):932–939, 1997.

[60] TA Rotting and A Gjelsvik. Stochastic dual dynamic programming for seasonal scheduling in the norwegian power system. *IEEE Transactions on Power Systems*, 7(1):273–279, 1992.

[61] Maria Elvira Pineiro Maceira, VS Duarte, DDJ Penna, LAM Moraes, and ACG Melo. Ten years of application of stochastic dual dynamic programming in official and agent studies in brazil-description of the newave program. *16th PSCC, Glasgow, Scotland*, pages 14–18, 2008.

[62] Anders Gjelsvik, Birger Mo, and Arne Haugstad. Long-and medium-term operations planning and stochastic modelling in hydro-dominated power systems based on stochastic dual dynamic programming. In *Handbook of power systems I*, pages 33–55. Springer, 2010.

[63] Tito Homem-de Mello, Vitor L De Matos, and Erlon C Finardi. Sampling strategies and stopping criteria for stochastic dual dynamic programming: a case study in long-term hydrothermal scheduling. *Energy Systems*, 2(1):1–31, 2011.

[64] MEP Maceira and JM Damázio. Use of the par (p) model in the stochastic dual dynamic programming optimization scheme used in the operation planning of the brazilian hydropower system. *Probability in the Engineering and Informational Sciences*, 20(1):143–156, 2006.

[65] BC Flach, LA Barroso, and MVF Pereira. Long-term optimal allocation of hydro generation for a price-maker company in a competitive market: latest developments and a stochastic dual dynamic programming approach. *IET generation, transmission & distribution*, 4(2):299–314, 2010.

[66] Anthony Papavasiliou. *Stochastic Dual Dynamic Programming.*

[67] Andrew B Philpott and Ziming Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36(4):450–455, 2008.

[68] Alexander Shapiro. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, 209(1):63–72, 2011.

[69] Bose Subhonmesh, Steven H Low, and K Mani Chandy. Equivalence of branch flow and bus injection models. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 1893–1899. IEEE, 2012.

[70] Georg Simon Ohm. *Die galvanische Kette, mathematisch bearbeitet*. TH Riemann, 1827.

[71] Javier Gusatvo Herrera Murcia. Self and mutual transmission line impedance estimation by means of the non-linear least squares method. In *Simposio Internacional sobre la Calidad de la Energía Eléctrica-SICEL*, volume 7, 2013.

[72] RC Silva, ECM Costa, S Kurokawa, and J Pissolato. Mutual coupling modeling in transmission lines directly in the phase domain. In *Electrical Power and Energy Conference (EPEC), 2011 IEEE*, pages 374–379. IEEE, 2011.

[73] Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.

[74] IBM Knowledge Center. Compiled versus interpreted languages. `https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zappldev/zappldev_85.htm`. Accessed: 2018-08-03.

[75] Inc. CVX Research. Software for disciplined convex programming. `http://cvxr.com/`. Accessed: 2018-08-03.

[76] Iain Dunning, Joey Huchette, and Miles Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

[77] Benoît Legat Joaquim Dias Garcia. Parameterjump.jl. `https://github.com/JuliaStochOpt/ParameterJuMP.jl`. Accessed: 2018-08-03.

[78] MOSEK ApS. Mosek faq. `https://docs.mosek.com/8.1/faq/faq.html`.

[79] Engie. Heures pleines, heures creuses. `https://particuliers.engie.fr/electricite/conseils-electricite/prix-electricite/heures-pleines-heures-creuses.html`. Accessed: 2018-06-9.

[80] Anjia Mao and M Reza Iravani. A trend-oriented power system security analysis method based on load profile. *IEEE Transactions on Power Systems*, 29(3):1279–1286, 2014.

[81] Shrink that footprint. Average household electricity consumption. `http://www.shrinkthatfootprint.com/average-household-electricity-consumption`. Accessed: 2018-06-09.

[82] Elia. Solar-pv power generation data. `http://www.elia.be/en/grid-data/power-generation/solar-power-generation-data/graph`. Accessed: 2018-06-09.

[83] Tesla. Technical specs. `https://www.tesla.com/powerwall?redirect=no`. Accessed: 2018-06-09.

[84] Anthony Papavasiliou, Shmuel S Oren, and Barry Rountree. Applying high performance computing to transmission-constrained stochastic unit commitment for renewable energy integration. *IEEE Transactions on Power Systems*, 30(3):1109–1120, 2015.

[85] Alejandro Navarro-Espinosa, Tuba Gozel, L Ochoa, R Shaw, and Dan Randles. Data analysis of lv networks: key parameters from one year of monitoring over hundreds ok uk lv feeders. *CIRED 2015*, 2015.

[86] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[87] Benoît Legat, François Pacaud, and Oscar Dowson. Cutpruners.jl. `https://github.com/JuliaPolyhedra/CutPruners.jl`. Accessed: 2018-08-03.

Appendix A

Intermediate Formulations

A.1 Single-Phase OPF after the Angle Relaxation

The single-phase optimal power flow after the angle relaxation is given by the following optimization problem.

$$\min_{P,Q,p,q,\ell,v,x,bc,bd} \quad \sum_{i\in\mathcal{N}} MC_{i,1}\cdot p_{i,1} + \mathbb{E}\left[\sum_{t=2\dots H}\sum_{i\in\mathcal{N}} MC_{i,t}\cdot p_{i,t,\omega_{[t]}}\right] \qquad \text{(OPF1-ar)}$$

$$\text{s.t.} \quad v_{i,t,\omega_{[t]}} - 2\cdot(R_iP_{i,t,\omega_{[t]}} + X_iQ_{i,t,\omega_{[t]}}) + (R_i^2+X_i^2)\cdot\ell_{i,t,\omega_{[t]}} = v_{A_i,t,\omega_{[t]}}$$
$$i\in\mathcal{E}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}$$

$$p_{i,t,\omega_{[t]}} - pD_{i,t,\omega_t} + \sum_{j\in C_i}(P_{j,t,\omega_{[t]}} - R_j\ell_{j,t,\omega_{[t]}}) - P_{i,t,\omega_{[t]}} - G_iv_{i,t,\omega_{[t]}} + bd_{i,t,\omega_{[t]}} - bc_{i,t,\omega_{[t]}} = 0$$
$$i\in\mathcal{N}^+, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}, \omega_t\in\Omega_t$$

$$p_{0,t,\omega_{[t]}} + \sum_{j\in C_0}(P_{j,t,\omega_{[t]}} - R_j\ell_{j,t,\omega_{[t]}}) - G_0v_{0,t,\omega_{[t]}} + bd_{0,t,\omega_{[t]}} - bc_{0,t,\omega_{[t]}} = 0$$
$$t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}$$

$$q_{i,t,\omega_{[t]}} - qD_{i,t,\omega_t} + \sum_{j\in C_i}(Q_{j,t,\omega_{[t]}} - X_j\ell_{j,t,\omega_{[t]}}) - Q_{i,t,\omega_{[t]}} + B_iv_{i,t,\omega_{[t]}} = 0$$
$$i\in\mathcal{N}^+, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}, \omega_t\in\Omega_t$$

$$q_{0,t,\omega_{[t]}} + \sum_{j\in C_0}(Q_{j,t,\omega_{[t]}} - X_j\ell_{j,t,\omega_{[t]}}) + B_0v_{0,t,\omega_{[t]}} = 0 \qquad t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}$$

$$\ell_{i,t,\omega_{[t]}} = \frac{P_{i,t,\omega_{[t]}}^2 + Q_{i,t,\omega_{[t]}}^2}{v_{i,t,\omega_{[t]}}} \qquad i\in\mathcal{E}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}$$

$$x_{i,1} = x_{\text{init},i} + \eta c_i\cdot bc_{i,1} - \frac{bd_{i,1}}{\eta d_i} \qquad i\in\mathcal{N}$$

$$x_{i,t,\omega_{[t]}} = x_{i,t-1,\omega_{[t]}} + \eta c_i\cdot bc_{i,t,\omega_{[t]}} - \frac{bd_{i,t,\omega_{[t]}}}{\eta d_i} \qquad i\in\mathcal{N}, t=2\dots H, \omega_{[t]}\in\Omega_{[t]}$$

$$0\le p_{i,t,\omega_{[t]}}\le p_{\max,i,t} \qquad i\in\mathcal{N}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}$$

$$0\le q_{i,t,\omega_{[t]}}\le q_{\max,i,t} \qquad i\in\mathcal{N}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}$$

$$0\le \ell_{i,t,\omega_{[t]}} \qquad i\in\mathcal{N}, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}$$

$$v_{\min,i}\le v_{i,t,\omega_{[t]}}\le v_{\max,i} \qquad i\in\mathcal{N}^+, t\in\mathcal{T}, \omega_{[t]}\in\Omega_{[t]}$$

$$
\begin{aligned}
& 0 \leq x_{i,t,\omega_{[t]}} \leq x_{\max,i} && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& 0 \leq bc_{i,t,\omega_{[t]}} \leq bc_{\max,i} && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& 0 \leq bd_{i,t,\omega_{[t]}} \leq bd_{\max,i} && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& v_{0,t,\omega_{[t]}} = 1 && t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}.
\end{aligned}
$$

A.2 The equivalent formulation of OPF3 with the SDP and rank constraints

The equivalent formulation of OPF3 with the SDP and rank constraints is given by the following optimization problem.

$$
\begin{aligned}
\min_{\mathbf{S},\mathbf{s},\mathbf{I},\mathbf{V},\mathbf{x},\mathbf{bc},\mathbf{bd}} \quad & \sum_{i\in\mathcal{N}} \sum_{\phi\in\Phi_i} MC^{\phi}_{i,1} \cdot \Re(s^{\phi}_{i,1}) + \mathbb{E}\left[\sum_{t=2\ldots H} \sum_{i\in\mathcal{N}} \sum_{\phi\in\Phi_i} MC^{\phi}_{i,t} \cdot \Re\left(s^{\phi}_{i,t,\omega_{[t]}}\right) \right] && \text{(OPF3rank)} \\
\text{s.t.} \quad & \mathbf{V}_{i,t,\omega_{[t]}} - \mathbf{S}_{i,t,\omega_{[t]}} \mathbf{Z}^H_{i,t,\omega_{[t]}} - \mathbf{Z}_{i,t,\omega_{[t]}} \mathbf{S}^H_{i,t,\omega_{[t]}} + \mathbf{Z}_{i,t,\omega_{[t]}} \mathbf{I}_{i,t,\omega_{[t]}} \mathbf{Z}^H_{i,t,\omega_{[t]}} = \mathbf{V}_{A_i,t,\omega_{[t]}} \\
& && i \in \mathcal{E}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& \mathbf{s}_{i,t,\omega_{[t]}} + \sum_{j\in C_i} \operatorname{diag}\left(\mathbf{S}_{j,t,\omega_{[t]}} - \mathbf{Z}_j \mathbf{I}_{j,t,\omega_{[t]}}\right) + \mathbf{bd}_{i,t,\omega_{[t]}} \\
& = \mathbf{sD}_{i,t,\omega_{[t]}} + \operatorname{diag}\left(\mathbf{S}_{i,t,\omega_{[t]}} + \mathbf{V}_{i,t,\omega_{[t]}} \mathbf{Y}^H_i\right) + \mathbf{bc}_{i,t,\omega_{[t]}} \\
& && i \in \mathcal{N}^+, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]}, \omega_t \in \Omega_t \\
& \mathbf{s}_{0,t,\omega_{[t]}} + \sum_{j\in C_0} \operatorname{diag}\left(\mathbf{S}_{j,t,\omega_{[t]}} - \mathbf{Z}_j \mathbf{I}_{j,t,\omega_{[t]}}\right) + \mathbf{bd}_{0,t,\omega_{[t]}} = \operatorname{diag}\left(\mathbf{V}_{0,t,\omega_{[t]}} \mathbf{Y}^H_0\right) + \mathbf{bc}_{0,t,\omega_{[t]}} \\
& && t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& \begin{bmatrix} \mathbf{V}_{i,t,\omega_{[t]}} & \mathbf{S}_{i,t,\omega_{[t]}} \\ \mathbf{S}^H_{i,t,\omega_{[t]}} & \mathbf{I}_{i,t,\omega_{[t]}} \end{bmatrix} \in \mathbb{S}_+ && i \in \mathcal{E}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& \operatorname{rank} \begin{bmatrix} \mathbf{V}_{i,t,\omega_{[t]}} & \mathbf{S}_{i,t,\omega_{[t]}} \\ \mathbf{S}^H_{i,t,\omega_{[t]}} & \mathbf{I}_{i,t,\omega_{[t]}} \end{bmatrix} = 1 && i \in \mathcal{E}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& \mathbf{x}_{i,1} = \mathbf{x}_{\text{init},i} + \eta c_i \cdot \mathbf{bc}_{i,1} - \frac{\mathbf{bd}_{i,1}}{\eta d_i} && i \in \mathcal{N} \\
& \mathbf{x}_{i,t,\omega_{[t]}} = \mathbf{x}_{i,t-1,\omega_{[t]}} + \eta c_i \cdot \mathbf{bc}_{i,t,\omega_{[t]}} - \frac{\mathbf{bd}_{i,t,\omega_{[t]}}}{\eta d_i} && i \in \mathcal{N}, t = 2\ldots H, \omega_{[t]} \in \Omega_{[t]} \\
& 0 \leq \Re\left(\mathbf{s}_{i,t,\omega_{[t]}}\right) \leq \mathbf{p}_{\max,i,t} && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& 0 \leq \Im\left(\mathbf{s}_{i,t,\omega_{[t]}}\right) \leq \mathbf{q}_{\max,i,t} && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& \mathbf{v}_{\min,i} \leq \operatorname{diag}\left(\mathbf{V}_{i,t,\omega_{[t]}}\right) \leq \mathbf{v}_{\max,i} && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& 0 \leq \mathbf{x}_{i,t,\omega_{[t]}} \leq \mathbf{x}_{\max,i} && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& 0 \leq \mathbf{bc}_{i,t,\omega_{[t]}} \leq \mathbf{bc}_{\max,i} && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& 0 \leq \mathbf{bd}_{i,t,\omega_{[t]}} \leq \mathbf{bd}_{\max,i} && i \in \mathcal{N}, t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]} \\
& \mathbf{V}_{0,t,\omega_{[t]}} = \mathbf{V}_{\text{swing}} && t \in \mathcal{T}, \omega_{[t]} \in \Omega_{[t]},
\end{aligned}
$$

where $\mathbf{V}_{\text{swing}} = \mathbf{v}_{\text{swing}} \cdot \mathbf{v}_{\text{swing}}^{H} = \begin{bmatrix} 1 & \frac{-1-\sqrt{3}}{2} & \frac{-1+\sqrt{3}}{2} \\ \frac{-1+\sqrt{3}}{2} & 1 & \frac{-1-\sqrt{3}}{2} \\ \frac{-1-\sqrt{3}}{2} & \frac{-1+\sqrt{3}}{2} & 1 \end{bmatrix}$.

Appendix B

Complete Solution of the Unbalanced Test Case

The solution of the unbalanced test case for each phase is presented in Figures B.1, B.2 and B.3.

$t=1$				$t=2$				$t=3$			
ω	p_0^ϕ	p_{11}^ϕ	$\sum_i x_i^\phi$	ω	p_0^ϕ	p_{11}^ϕ	$\sum_i x_i^\phi$	ω	p_0^ϕ	p_{11}^ϕ	$\sum_i x_i^\phi$
								1	6.605	3.312	8.168
				1	0.079	3.178	5.482	2	0.055	2.724	8.041
								3	0.014	2.200	13.089
								4	0.003	1.731	22.713
								1	1.865	3.321	4.880
				2	0.019	1.748	6.144	2	0.021	2.225	8.305
								3	0.030	0.075	12.018
1	15.152	1.321	9.659					4	0.036	0.081	22.201
								1	1.196	3.321	5.351
				3	0.024	0.704	7.336	2	0.023	1.336	8.721
								3	0.026	0.064	13.186
								4	0.035	0.074	23.365
								1	2.662	3.301	11.920
				4	0.004	2.374	12.972	2	0.003	1.566	14.290
								3	0.019	0.180	18.616
								4	0.008	0.622	29.021
p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}
0	10	20	1	1/3	50	20	4/3	1	50	20	7/6

Table B.1: The first three time stages of the SDP solution for the unbalanced test case, with 5 solar panels per bus and $PL = 50$. p is given in [kW/4] and x in kWh. This table presents the first phase of the solution, i.e. $\phi = 1$.

$t=1$				$t=2$				$t=3$			
ω	p_0^ϕ	p_{11}^ϕ	$\sum_i x_i^\phi$	ω	p_0^ϕ	p_{11}^ϕ	$\sum_i x_i^\phi$	ω	p_0^ϕ	p_{11}^ϕ	$\sum_i x_i^\phi$
								1	4.619	3.334	4.218
				1	0.383	3.334	2.872	2	2.313	2.629	7.151
								3	0.057	2.100	10.429
								4	0.081	0.497	19.531
								1	3.298	3.331	4.174
				2	0.029	3.053	4.057	2	0.030	2.507	6.529
								3	0.015	0.894	10.677
1	9.493	3.334	6.506					4	0.020	0.045	20.597
								1	2.807	3.330	5.076
				3	0.004	2.439	5.549	2	0.006	2.591	7.935
								3	0.019	0.163	11.478
								4	0.021	0.039	22.024
								1	3.787	3.325	8.786
				4	0.076	1.300	8.574	2	0.037	2.882	11.045
								3	0.005	1.342	15.351
								4	0.041	0.181	24.697
p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}
0	10	20	1	1/3	50	20	4/3	1	50	20	7/6

Table B.2: The first three time stages of the SDP solution for the unbalanced test case, with 5 solar panels per bus and $PL = 50$. p is given in [kW/4] and x in kWh. This table presents the second phase of the solution, i.e. $\phi = 2$.

$t=1$				$t=2$				$t=3$			
ω	p_0^ϕ	p_{11}^ϕ	$\sum_i x_i^\phi$	ω	p_0^ϕ	p_{11}^ϕ	$\sum_i x_i^\phi$	ω	p_0^ϕ	p_{11}^ϕ	$\sum_i x_i^\phi$
				1	0.056	3.105	5.954	1	4.095	3.320	7.595
								2	0.032	2.319	7.953
								3	0.004	1.929	12.069
								4	0.064	1.470	20.076
				2	0.019	1.659	6.175	1	0.499	3.324	4.771
								2	0.022	1.114	7.229
								3	0.029	0.048	10.785
1	12.047	2.215	8.752					4	0.023	0.069	19.360
				3	0.023	0.505	6.742	1	0.312	3.324	5.125
								2	0.022	0.977	7.652
								3	0.029	0.045	11.325
								4	0.024	0.063	19.878
				4	0.007	2.400	11.670	1	1.781	3.299	11.100
								2	0.007	1.815	13.034
								3	0.015	0.128	16.022
								4	0.040	0.374	24.554
p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}	p_{PV}	MC_0	MC_{11}	p_{load}
0	10	20	1	1/3	50	20	4/3	1	50	20	7/6

Table B.3: The first three time stages of the `SDP` solution for the unbalanced test case, with 5 solar panels per bus and $PL = 50$. p is given in [kW/4] and x in kWh. This table presents the third phase of the solution, i.e. $\phi = 3$.

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve **www.uclouvain.be/epl**