NTNU
Norwegian University of
Science and Technology

Master's thesis

2021

Petter Engblom Nordby

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Energy and Process Engineering

Petter Engblom Nordby

Optimization of flexible renewable energy systems under uncertainty

June 2021

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Optimization of flexible renewable energy systems under uncertainty

Petter Engblom Nordby

Energy and Environmental Engineering

Submission date:	June 2021
Supervisor:	Truls Gundersen
Co-supervisor:	Johannes Jäschke Avinash Subramanian

Norwegian University of Science and Technology
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

Department of Energy
and Process Engineering

MASTER'S THESIS

for

Petter Engblom Nordby

Spring 2021

Optimization of Flexible Renewable Energy Systems under Uncertainty

Supervisor:	Truls Gundersen
Co-supervisors:	Johannes Jäschke
	Avinash Subramanian

Problem Description

While significant investments in renewable energy have led to a sharp decline in the related costs, uncertain and intermittent behavior complicates the process of finding optimal renewable energy system (RES) designs. Designing a flexible renewable power plant is imperative for coping with inherent fluctuations for instance in weather patterns and power demand.

One way of dealing with the uncertainty in the design process is by the use of stochastic programming, which includes a representation of the uncertain parameters in the optimization problem. Increased RES flexibility can also be achieved by incorporating energy storage technologies, such as batteries or hydrogen, aiming to effectively mitigate the effect of fluctuating demand and power production. However, RES both have a long project lifetime and requires a high temporal resolution to represent the substantial short-term parameter variation. In turn, the complexity of these optimization problems rapidly increases, and careful consideration is needed for both modeling and solving. In addition, developing accurate and realistic models typically result in introducing nonconvexities.

Through the novel optimization software GOSSIP and a scenario representation of uncertainty, these types of problems can be solved. GOSSIP provides a C++ framework for the modeling of two-stage stochastic programs and their effective solution using a decomposition-based approach. Specifically, GOSSIP has the inclusion of the Non-convex Generalized Benders Decomposition (NGBD) algorithm, which allows for non-convex optimization problems and scales favorably with the number of scenarios.

Thus, the main objective of this master´s thesis is to use the GOSSIP software to develop and solve the cost-optimal design and operation problem of flexible renewable energy systems under uncertainty. The following tasks are to be considered:

1. Investigate and implement a renewable energy system model for the flexible design that includes short-term energy storage using batteries.
2. Investigate approaches to account for and model uncertainty with hourly temporal resolution. This would involve investigation of suitable scenario generation and methods.
3. Investigate the use of nonlinear models to improve model accuracy, for instance by accounting for degradation of the battery with use.
4. Suggestions for further work for the flexible design problem of renewable energy systems will be made.

Abstract

Flexibility of energy systems is a term used for the ability to react to changes in circumstances and to cover varying demand loads. For the renewable energy systems (RES), sufficient flexibility can become an issue due to inherent intermittency and fluctuations of renewable energy sources. A *flexible design* of a system mitigates these variations by increased production capacities, combination of different energy sources, and/or installation of energy storage technologies. The growing need for flexible plants requires investigation of system environments and behaviours during the design stage. Finding the optimal design is a complex process subject to large uncertainties due to long project lifetimes and a need for high temporal resolution.

This work studies the flexible design problem of renewable energy systems in the novel optimization software GOSSIP, which provides a framework for formulation and efficient solution of large and complex two-stage stochastic programs. Two-stage stochastic programming is an optimization approach that can account for uncertainty by separation of decision-variables into two stages. Thus, first-stage variables represent the design decisions and second-stage variables represent the operational decisions. Twelve 24-hour scenarios are used to represent the uncertainty through typical design days. A mixed-integer linear programming (MILP) model for studying an energy system with an end user, solar and wind energy sources, and a battery for hourly energy storage is constructed using the GOSSIP framework. Further, the model is developed to be compatible with a decomposition algorithm embedded in GOSSIP, providing a base for further problem investigation.

The flexible design strategy using the two-stage stochastic programming framework is shown to turn a project profitable by increasing flexibility through battery energy storage and adjusting the installed capacity of renewable sources. While being unprofitable with a design based on average values, the flexible design obtained a value of the stochastic solution of 197 k$ annually. Further, constraining construction cost is shown to affect the design decisions for achieving flexibility. For battery energy storage, assumptions and modelling choices are found to potentially cause suboptimal designs. Therefore, it is argued for a need for further investigation of energy storage modelling in the GOSSIP framework for the renewable energy system flexible design problem. Further work with an extensive case study with rigorous uncertainty modelling, expansion of the model with more components, and more detailed models should be investigated. Finally, future work will include the creation of a user-friendly interface for making extensible RES models in GOSSIP.

Sammendrag

Fleksibilitet for energisystem er et begrep som brukes om evnen til å dekke en usikker og varierende etterspørsel av energi. For fornybare energisystemer kan tilstrekkelig fleksibilitet bli et problem på grunn av naturlige svingninger og periodevise brudd i energiproduksjon. Et fleksibelt design kan dekke etterspørselen av energi ved å øke produksjonskapasiteten, kombinere forskjellige energikilder og/eller benytte energilagring. Det økende behovet for fleksible system krever derimot en grundig undersøkelse av systemets miljø og operasjonelle aspekt i designfasen. Å finne den beste løsningen er en kompleks prosess med stor usikkerhet på grunn av lange tidshorisonter og behov for fin tidsoppløsning.

I følgende arbeid studeres designproblemet for å oppnå tilstrekkelig fleksibilitet med fornybare energisystemer ved bruk av GOSSIP. GOSSIP er en ny programvare for optimalisering av to-trinns stokastiske programmer som gir et rammeverk for formulering og effektive løsninger av store og komplekse problem. To-trinns stokastisk programmering er en optimaliseringsmetode som tar høyde for usikkerhet ved å separere variabler i to trinn. Typisk representerer variabler i første trinn avgjørelsene for design og andre trinns variabler de operasjonelle beslutningene. Tolv 24-timers scenarier brukes til å representere usikkerheten mellom trinnene ved å konstruere dager med typiske svingninger i fornybare system. Ved bruk av rammeverket for modellering i GOSSIP, er det bygd en lineær mixed-integer programmeringsmodell (MILP) av et energisystem med sluttbrukere, solenergi, vindkraft og batteri. Videre er modellen laget for å være kompatibel med en dekomponeringsalgoritme innebygd i GOSSIP, og gir et grunnlag for videre undersøkelse av problemet.

Den fleksible designstrategien ved bruk av det to-trinns stokastiske rammeverket er vist å kunne gjøre et prosjekt lønnsomt ved å øke fleksibiliteten gjennom energilagring og justering av produksjonskapasiteter. Selv om det viste seg å ikke være lønnsomt med et design basert på en deterministisk tilnærming til optimalisering av problemet, viste en stokastisk løsning at det potensielt kunne bli tilført en verdi av på 197 tusen dollar årlig. Videre er det vist at begrensning av byggekostnadene påvirker de beslutningene som gjøres i designfasen for å oppnå fleksibilitet. For batterilagring, er antagelser brukt for modellering vurdert til å kunne forårsake suboptimale design. Derfor argumenteres det for et behov for videre undersøkelse av modellering av energilagring i GOSSIP-rammeverket. Videre arbeid innebærer en omfattende casestudie med grundig usikkerhetsmodellering og videre utvidelse med flere komponenter, samt undersøkelse av detaljerte modeller. Fremtidig arbeid vil også inkludere å lage et brukervennlig grensesnitt for å lage utvidbare RES-modeller.

Acknowledgements

I would like to express my gratitude to my supervisor, Truls Gundersen, for his guidance and support over the course of this project. His availability and rigorous feedback on sections in this thesis have been highly treasured. Additionally, I would like to thank my co-supervisors, Avinash Subramanian and Johannes Jäschke, for supporting me with interesting discussions and valuable advice. Avinash´s suggestions and ideas are greatly acknowledged as being imperative to the outcome of the project. I am thankful for the support and commitment of all my supervisors, which has been an important motivating factor over course of my Master´s project. Finally, I would like to thank my fellow student, Mari Elise Rugland, for providing an excellent partnership and teamwork on the joint topics of our projects.

Nomenclature

Abbreviations

BD	Benders decomposition
CRF	Capital recovery factor
EEV	Evaluation of the expected value solution
$EEVP$	Problem of evaluating the expected value solution
EVP	Expected value problem
GBD	Generalized Benders decomposition
$GOSSIP$	Global optimization of non-convex two-stage stochastic mixed-integer programs
$LCOE$	Levelized cost of energy
LP	Linear program/problem
$MICP$	Convex mixed-integer non-linear program/problem
$MILP$	Mixed-integer linear program/problem
$MINLP$	Mixed-integer non-linear programming/problem
MIP	Mixed-integer program/problem
$NGBD$	Non-convex generalized Benders decomposition
P	Program/problem
RES	Renewable energy system
RP	Optimal value of recourse problem
$SOES$	Security of energy supply
SP	Stochastic program/problem

Subscripts and Superscripts

b	Binary
BAT	Battery
ch	Charging
d	Discrete
dis	Discharging
exp	Export

Imp	Import
LB	Lower bound
PV	Solar PV
UB	Upper bound
WT	Wind turbine

Symbols

η	Efficiency
$\mathbb{E}$	Expected value
ω	Probability
C	Construction cost
E	Energy
f	Power flow variable
FiT	Feed-in-tariff
IR	Solar irradiance
L	Demand load
M	Maintenance cost
OC	Occurred cost
Q	Charge
q	Power
SoC	Battery state of charge
T	Number of timesteps
W	Wind speed

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Impelled by significant investments, technological progress and economies of scale have led to a sharp decline in the cost of electricity production by renewable energy sources [1]. Photovoltaic solar energy has seen an 82% reduction in the past decade, outperforming the marginal cost per unit of energy produced by existing coal-fired plants. In fact, with a moderate price on carbon, even the cheapest new coal plant were undercut in costs by the larger part of renewable capacity additions in 2019. Replacing the 500 most expensive coal fired plants with solar and wind could potentially save 23 billion dollars annually according to IRENA [2].

While the levelized cost of energy (LCOE) of renewable sources are becoming increasingly competitive with fossil fuel driven plants, a technology's generating cost of electricity can vary significantly from its actual system value. Pure cost calculations neglect important differences in and interactions between market, system and technology specific characteristics. These aspects tend to decrease the value of renewable energy in the energy system [3].

Flexibility of energy systems is the ability to produce sufficient amounts of energy, i.e. match energy supply with the demand of end users. However, operational challenges arise for renewable energy systems due to the fluctuating nature of renewable energy input and energy demand. The inherent fluctuations, characterized by significant uncertainties, reduce the readily available energy, and consequently, the ability to cover demand loads. The capability of reacting swiftly to changes in operational circumstances are needed to mitigate the uncertain fluctuations and obtain energy security. Sufficient system flexibility therefore has to be considered at the design stage by optimal selection of system technologies and unit size.

The flexible design problem of renewable energy systems aims to increase flexibility by taking uncertainties and fluctuations into account when designing the system. Measures that improve the reliability to cover demand loads include:

- Additional energy production capacity to provide a general higher level of power on average.
- The use of different energy sources to exploit power generation under different weather conditions.
- Installation of energy storage, such as batteries or hydrogen technology, to offset power surpluses and deficits in the system.

It should be noted that there is a trade-off between an increase in capital expenses and revenue associated with increased robustness to uncertainty. Even though a flexible design with additional capacity increases investment costs, it is more likely that the system can cover the demand load.

For short time horizons in particular, battery energy storage has been posed as a solution to increase system flexibility. Batteries have the capability to quickly absorb and discharge electricity, and can be used to balance out the fluctuations in light of the rest of the system and market. This way, batteries can defer investments in production capacities and obtain a more flexible system without increasing production [4]. With the inclusion of energy storage, it is believed that substantial amounts of renewable energy sources can be integrated with the global energy mix [5].

Designing a renewable energy system can be a challenging task and may entail a substantial risk of sub-optimal decisions. Techniques for optimization under uncertainty should therefore be considered for this purpose. Specifically, two-stage stochastic programming is a commonly utilized framework [6]. In a two-stage stochastic program formulation, decision-making is performed sequentially in two separate stages. A set of first-stage decisions has to be made before any uncertainty is revealed, and a set of second-stage decisions can then be made for each realization of uncertainty.

However, optimization of the flexible design of an energy system is often characterized by significant complexity. Numerous technologies are often available, which are subject to many uncertain parameters with high variability. The rapidly fluctuating natures of energy markets and renewable energy sources require long time horizons and high resolutions from an operational modelling perspective, driving up the number of decision variables and constraints.

Furthermore, accurate modelling of system technologies, such as battery energy storage, may add non-linearities to the model. Non-linearities in an optimization model can cause a non-convex program, known for being hard to solve. Due to its reasonable size, Gabrielli et al. [7] proposed a deterministic mixed-integer linear programming approach for a multi-energy system that allows hourly resolution and account for seasonal changes. However, the approach places a limited focus on the uncertain variations.

In another study, uncertainty was included in the problem formulation through a two-stage stochastic mixed-integer linear programming (MILP) approach [8]. For two-stage stochastic programming, MILP formulations of the energy system design problems is the preferred approach due to their simplicity and the availability of releavnt optimization software, as seen in [9], [10], [11] and [12]. However, recent advances are unlocking the key for solving nonconvex two-stage stochastic problems [13]. Additionally, large scale problems would require the need for decomposition approaches.

In summary, optimization of the flexible design problem under uncertainty poses considerable challenges as complexity may end up restricting accurate modelling of both uncertainty and system technologies. Additionally, the inclusion of energy storage through batteries is expected to be the key for obtaining sufficient system flexibility, but it further increases the decision-making complexity. The novel optimization software GOSSIP, developed by researchers in the Process Systems Engineering Laboratory (PSEL) at MIT, has built-in functionality for decomposition-based approaches that can guarantee optimal solutions for large and non-linear two-stage stochastic problems [14]. Utilizing such decomposition strategies for allowing large scale and complex problems is hypothesized to be valuable for the study of the flexible design problem of renewable energy systems.

1.2 Objectives, scope and limitations

Thus, the objective of this thesis is three-fold:

- Firstly, we develop a multi-period optimization model for the flexible design problem of renewable energy systems under uncertainty. The model is used to obtain optimal flexible solutions in GOSSIP by using a two-stage stochastic programming approach compatible with a decomposition algorithm available in GOSSIP [15].
- The program is aimed to have hourly temporal resolution to be able to account for uncertain fluctuations and employ measures at the design stage for increasing flexibility of renewable energy systems.

- Finally, we investigate the inclusion of energy storage in the model and the effect of related modelling choices in a two-stage stochastic program.

A simple energy system with energy storage, solar energy and wind energy is used for the flexible design problem in this thesis. Uncertainty will be accounted for when designing the system, while operation will be optimized with perfect information. The scope of this project has certain limitations:

- Energy modelling is limited to linear relations.
- Energy storage is limited to a battery that works on hourly variations.
- Non-linear battery models and degradation effects have been limited to investigation of possible dangers of omitting these aspects when modelling energy storage.
- Modelling of the energy system is limited to energy production and storage technologies. Components such as transformers and grid connections are omitted.
- A simple representation of uncertainty and fluctuating patterns has been employed. Thus, results are limited to illustration of possible ways to obtain flexible designs.

1.3 Structure of report

In Chapter 2, a brief background on renewable energy systems, solar energy, wind energy and batteries are presented for the purpose of later technology modelling choices and implications. An overview and fundamental concepts of optimization, mixed-integer programming, stochastic programming and decomposition algorithms are presented in Chapter 3. Chapter 4 presents the method and the model used in this thesis. The modelling approach of flexible renewable energy system design is given in Sections 4.1 and 4.2, linking the background information in Chapters 2 and 3. Further, the resulting complete model formulations and the program inputs are given in Sections 4.3 and 4.4 respectively. Results and discussion of the results are then presented in Chapters 5 and 6. Firstly, the optimal solution and flexible design is presented together with an assessment of the battery model in Chapter 5. Chapter 6 will then present a discussion on the modelling approach and GOSSIP framework, before concluding remarks and suggestions for further work are given in Chapter 7.

1.4 Contributions

The main contributions of this work are:

- Developing an MILP model for the flexible design problem.
- Renewable energy system modelling approach for compatibility with a decomposition algorithm for solving non-convex problems.
- Performing a sensitivity analysis to investigate assumptions related to battery operation and behaviour in models for two-stage stochastic optimization programs.
- Models for testing during the development of an object-oriented framework for the flexible design problem of renewable energy system.

2 | Renewable Energy Systems

Two trends have become increasingly prominent in the energy sector over the last decades. Firstly, the transition from fossil to renewable energy sources have begun to speed up. 2020 is the first year renewable energy production has overtaken fossil fuels in the EU due to increases in renewable capacity, such as wind and solar power, and a falling share of coal [16]. Secondly, renewable energy sources are fuelling the deployment of decentralized and distributed energy systems with more and smaller electricity producers [17]. In this chapter, brief background information on three important technologies for the development of decentralized renewable energy systems will be presented. Additional background on the working principles of the technologies can be found in Appendix A.

2.1 Solar energy

Harvesting energy from the sun has been on the rise this past decade, and 2019 alone saw a 22% increase in solar energy production capacity [18]. However, challenges regarding solar PV are present due to mismatches between electricity production and demand load patterns. This affects the value of energy output from solar farms.

2.1.1 Solar irradiance

Solar PV has fluctuating power production due to the varying available intensity of solar radiation throughout the day. However, typical patterns are fairly predictable. The value of solar mean flux incident on a theoretical cross-section of the atmosphere is called the solar constant and is illustrated in Figure 2.1. The solar radiation incident on the earth's surface is, however, determined by atmospheric effects and solar angels.

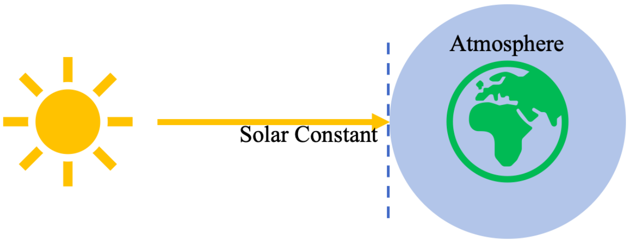


Figure 2.1: Solar constant on the atmosphere.

The atmospheric layer contains gases, dust, pollutants and other particles, causing a part of the radiation to be absorbed, scattered or reflected [19]. This radiation is called diffuse radiation, and the part of the radiation that has remained unaffected by the atmosphere is the beam radiation. Consequently, an unclear atmosphere will have a reducing effect on the solar intensity, as illustrated in Figure 2.2.

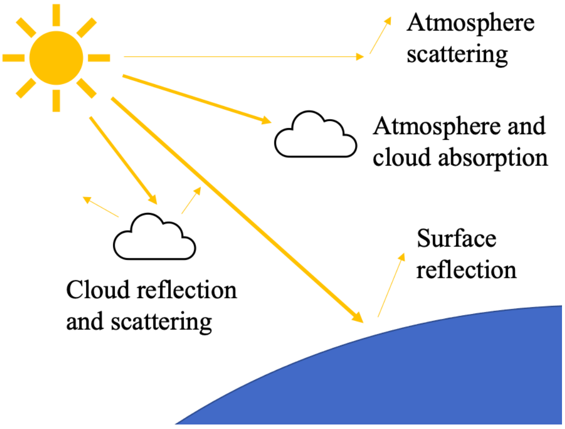

Atmosphere scattering
Atmosphere and cloud absorption
Surface reflection
Cloud reflection and scattering

Figure 2.2: Atmospheric effects on radiation.

Solar angles impact the radiation intensity on the earth surface. Air mass, illustrated in Figure 2.3, is a measure of the amount of air the sun rays have to pass through. It is normalized to the shortest possible distance through the atmosphere, which is when the sun rays are incident normal to the surface of the earth. The angle between the radiation path and the normal line to the earth's surface is called the solar zenith angle, denoted by θ_z, and defines the sun's altitude.

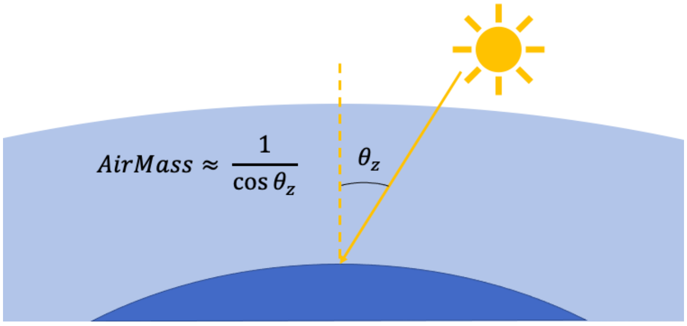


Figure 2.3: Zenith angle and air mass.

The radiation received on a solar panel is also affected by the angle of which the solar panel has been mounted. The tilt of the solar panel is meant to counter-act the fact that the zenith angle causes a reduction in the horizontal solar intensity as earth surface area per unit of radiation energy is increased. However, the panel will have a fixed tilt and position. Radiation incident on the panel will therefore depend on the solar azimuth angle, which defines the sun's relative location on the horizon. For a tilted surface, radiation reflected off the earth's surface or other objects may be incident on the surface.

This reflected radiation depends on the object's properties, where the fraction of light reflected is defined as the albedo of the object. For instance, while the reflectivity of the earth´s surface is varying, it is often assumed that it has an albedo of 0.2 [20], meaning that 20% of the radiation incident on the surface is reflected. The global solar irradiance on a receiver, IR_g, is the total radiation incident on its surface. In summary, it can be assumed to consist of three components:

- Beam radiation, IR_b: The direct solar radiation.
- Diffuse radiation, IR_d: Scattered solar radiation that hits the solar panel.
- Albedo, IR_a: The reflected solar radiation that a solar panel may receive after being reflected from the ground.

Calculating the three components above through-out the day can be used for estimation of solar radiation patterns. This is imperative in evaluation of solar power in renewable energy systems. Tools, which also includes weather data for average atmosphere clearness, are available for this purpose [21].

2.1.2 Solar cells

Solar cells can convert radiation directly into electricity by exploiting the photovoltaic effect [22]. A brief overview of the concept of photovoltaics can be found in Appendix A.1.

The IV characteristic curve of a solar cell, illustrated in Figure 2.4, is the key for determining its performance and efficiency [22]. The open circuit voltage occurs when there is no current, and remain fairly constant under constant temperature. The short circuit current on the other hand, depends on the intensity of the solar radiation.

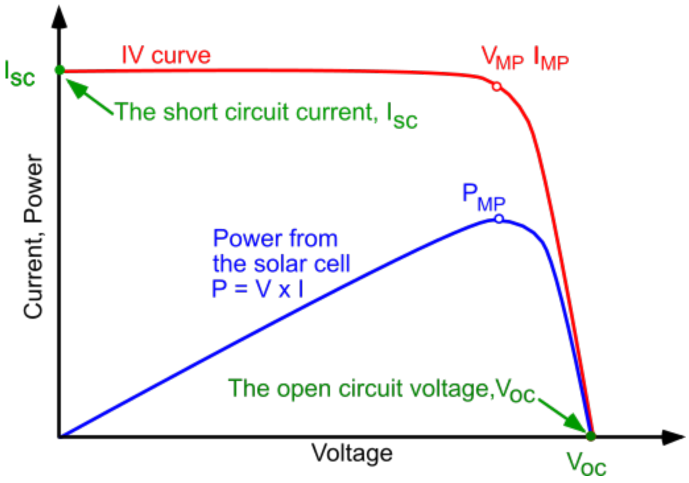


Figure 2.4: IV-curve of a solar cell [23].

Ideal operation of a solar cell is at the maximum power point, V_{MP} and I_{MP}, illustrated by P_{MP} in Figure 2.4. This point determines the efficiency of the solar cell, linking the power in the solar radiation to the power output of the solar cell, as given by Equation 2.1.

$$\eta = \frac{V_{MP}I_{MP}}{IR} = \frac{q_{out}}{IR} \tag{2.1}$$

where IR is the solar radiation incident on the solar cell and q_{out} is the power output. Single solar cells are connected in series or parallel to create photovoltaic modules called solar panels, shown in Figure 2.5. PV farms in a RES would require large areas of interconnected solar panels in strings and arrays to achieve a sufficient power output.

In addition to the design and construction, weather conditions play an important role for cell efficiency:

- Low temperatures are preferred as high temperatures will decrease open circuit voltage and the module efficiency.
- A higher solar intensity improves efficiency through altering the maximum power point.

While temperature can play a significant role in cell efficiency, the differences in normalized efficiencies due to irradiance are small [24]. Commercial PV modules usually have efficiencies around 15-20 % [25].

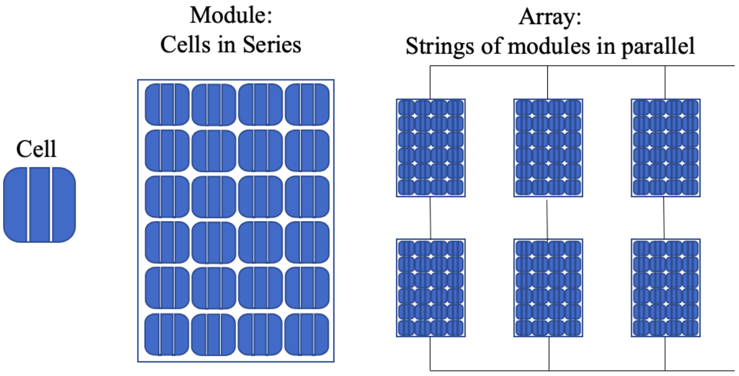


Figure 2.5: Connection of solar cells.

However, the efficiency of solar panels will decrease with time as radiation and weather exposure cause degradation to materials, for instance by deterioration of antireflective coating will reduce the efficiency [26]. Lifetime of the technology is given as the time before a certain efficiency loss is reached. PV manufacturers typically guarantee 80% of initial efficiency after 25-30 years of operation [27].

2.2 Wind energy

Contrary to solar energy, wind energy does not follow a typical pattern of production throughout the day. Rather, wind energy from single turbines tend to have more uncertain fluctuations that correlates more with wind turbines in the same area. As the distances between wind turbines increase, the fluctuations tend to be uncoupled. However, this requires large land ares and nature interference [28]. While placing wind turbines off-shore have been proposed as a future solution for mitigating downsides of wind energy, on-shore wind turbines are currently seen as the most economically viable option for harvesting the wind energy [29].

2.2.1 Wind speeds

The wind speed, determining the power output of wind turbines, does not follow typical patterns. Instead, the patterns are known for being more random than solar irradiation patterns.

Wind speeds tend to be measured close to the ground, whereas wind turbines are usually at higher altitudes. The boundary layer effect causes wind speed at higher altitudes to increase relative to the distance to the ground [30]. The increasing wind velocity with altitude is illustrated in Figure 2.6, where h is the height and v is the velocity vector of the wind.

The wind speed at higher altitudes can be estimated by relating the ratios to a surface roughness factor [30], as illustrated in Equation 2.2.

$$W = W_0 \cdot \left(\frac{h}{h_0}\right)^{\frac{1}{7}} \tag{2.2}$$

where W is the wind speed at altitude h, W_0 is the wind speed at reference altitude h_0, and the coefficient $\frac{1}{7}$ represent the surface roughness factor.

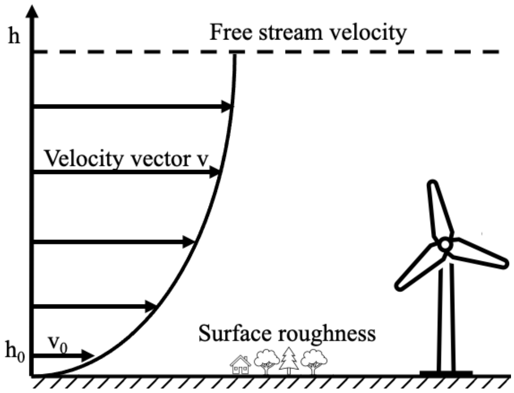


Figure 2.6: Boundary layer effect on wind speed.

2.2.2 Wind turbines

While a wind turbine's power output at given wind speeds depends on the size of the generator and the strengths of its transmission, it will often be described by its rated power and rated wind speed [31]. Rated power is the power produced by the wind turbine at the optimal wind conditions, called rated wind velocity, i.e. the rotor operates at desired rotational speed.

The wind speed is highly fluctuating and may be higher than the rated velocity. Despite the opportunity to obtain extra power by increasing the rated power and velocity to accommodate high wind speeds, this will usually lead to an over-designed system [31]. On the other hand, exceeding the optimal revolutions per minute (rpm) increases the risk for failure or overloading the wind turbine. Restriction of the blade velocity is therefore needed, achievable through pitch controlled wind turbines. The principles of generation and control of the rotational motion of blades on a wind turbine are explained in Appendix A.2.

Wind turbines with pitch control will start to control the blade velocity when the turbine approaches its rated power, keeping the turbine at its optimal rpm [31]. The relationship of the power output of a pitch-controlled wind turbine to wind speed is referred to as a power curve. While the actual observed power output can be described as a scatter plot, it can be approximated by an idealized power curve or manufacturer power curve, illustrated by MPC in Figure 2.7.

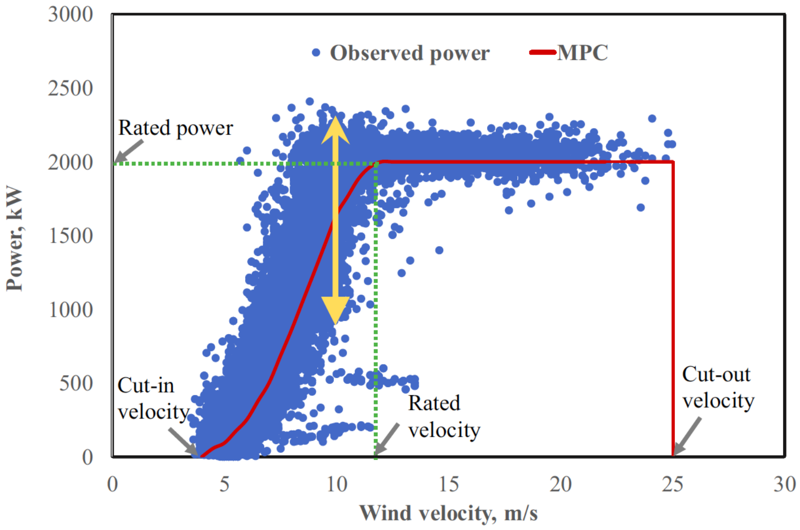


Figure 2.7: Real data from a pitch controlled wind turbine and its idealized power curve [32].

From the power curve, it is seen that the wind turbine can be characterized into four different performance regions by the cut-in velocity, rated velocity and cut-out velocity:

1. Below the cut-in velocity, the wind turbine is idle and will not generate power.
2. Between the cut-in velocity and the rated velocity, the power output will increase with increasing wind speeds.
3. From the rated velocity, the power output will remain at the rated power until the wind speed reaches the cut-out velocity.
4. At the cut-out velocity, the system shuts down as the wind speed becomes too high for safe operation.

An idealized power curve proposed by Matthew [31], given in Equation 2.3, can be used for power output between the cut-in velocity and the rated velocity.

$$q = q_d \cdot \frac{W^3 - W_{min}^3}{W_d^3 - W_{min}^3} \tag{2.3}$$

where q is the power output, q_d is the rated power, W is the wind speed, W_{min} is the cut-in velocity, and W_d is the rated wind speed.

Degradation driven by time and weather conditions, however, causes power output losses of around 1.5% per year for wind turbines due to fouling of blades and other components efficiency losses [33]. The lifetime of wind turbines can be estimated to be about 20 years due to this wind turbine degradation, in addition to fatigue and stress on the wind turbine materials.

2.3 Batteries

Utility-scale batteries are stationary batteries of several megawatt-hours connected to a grid or power production unit. While many different battery technologies are available, the largest part of the growth has been largely due to the cost reduction of Lithium-ion (Li-ion) batteries in recent years [4]. Li-ion batteries are viewed as one of the most mature and prevalent alternatives for large scale storage, providing low self-discharge losses, high energy and power density, decent expected lifetime and reasonable costs [34]. The working principle of Li-ion batteries is presented in Appendix A.3.

State of charge

A battery's state of charge is the percentage of available capacity to the maximum capacity of the battery, indicating the level of energy in the battery. The battery capacity indicates the amount of electrical charge stored in a battery in ampere hours (Ah). This number represents how long the battery can discharge a certain current. While actual available energy stored in the battery will depend on the battery cell voltage level, this is often assumed to be constant for evaluating the amount of energy that can be stored in the battery.

Voltage losses

The theoretical maximum power is given by the open circuit voltage of the battery, which is a measure of the electrode potential with no current flow. Thereby, no losses are associated with the amount of current that flows out of or in to the battery. The open circuit voltage is dynamic in the sense that it depends on the battery state of charge. Further, the open circuit voltage depends on whether the state was reached during charging or discharging. In Figure 2.8, typical charge-discharge curves and the effects on open circuit voltage of a battery are presented.

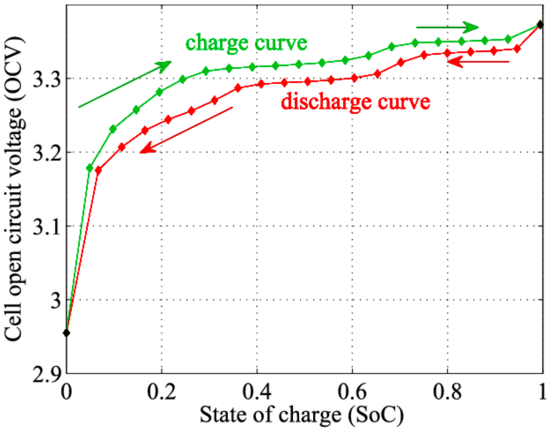


Figure 2.8: Charge - discharge profile [35].

The voltage losses will cause the actual battery voltage to differ from the open circuit voltage. During charging operation, the battery voltage will have to be greater than the open circuit voltage. Likewise, discharging the battery will give a battery terminal voltage that is less than the open-circuit voltage. Furthermore, the voltage losses will increase in magnitude with an increasing current, as shown in Figure 2.9: Increasing current will induce greater ohmic losses due to internal resitances, activation polarization, and concentration polarization.

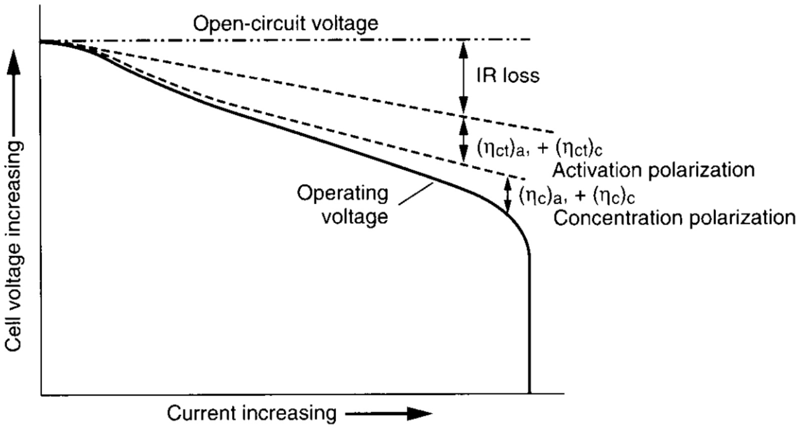


Figure 2.9: Voltage losses when increasing current [36].

Energy efficiency

The voltage losses are important in order to estimate the energy efficiency of the battery, η_{energy}, given by Equation 2.4.

$$\eta_{energy} = \eta_{voltage} \cdot \eta_{coulombic} = \frac{V^{dis}}{V^{charge}} \cdot \frac{Q^{dis}}{Q^{charge}} \tag{2.4}$$

Where: $\eta_{voltage}$ is the voltage efficiency; V^{dis} and V^{charge} are the battery voltage during discharging and charging respectively; and $\eta_{coulombic}$ is the Coulombic efficiency, given by the ratio of the total charge released during discharge, Q^{dis}, and put into the battery during charge, Q^{charge}. While the Coulombic efficiency is very high in lithium-ion batteries, the voltage efficiency can cause significant losses [36].

The amount of current drawn or fed into the battery is referred to as the C-rate. This gives a view of how high the battery power flow is relative to its capacity. Specifically, the C-rate indicates how many theoretical full capacities it discharges in an hour. For instance, a C-rate of 0.5 will indicate that the entire capacity is withdrawn in two hours, while a C-rate of 2 will discharge it in 30 minutes. Higher C-rates represent more power and faster cycles, but will increase the voltage losses due to increased current. Thereby, higher C-rates will decrease the energy efficiency.

Battery degradation

Various forms of degradation of batteries are well known phenomena: The battery experiences both capacity loss and power fading due to degradation mechanisms such as parasitic reactions, stress and volume changes [37]. Additionally, degradation can affect the lifetime of a battery, referred to as the battery cycle life, i.e. how many times the battery can be charged and discharged until required functions cannot be performed. The rate of power fading and capacity loss can be elevated by numerous aspects of how the battery is operated, including:

- High C-rates
- Low states of charge
- High states of charge

In addition, increased temperatures are well known for speeding up battery capacity loss and power fading [37]. However, proper maintenance and operation can slow down the performance drop. For instance, the battery cycle life has been shown to be extended by avoiding high battery states of charge [38]. The Tesla powerwall, an example of a Li-ion battery unit designed for complementary use to solar and wind power, is expected to last approximately 15 years with daily cycling [39].

3 | Mathematical Programming

In this chapter, the mathematical background is provided for basic understanding of relevant optimization and mathematical programming concepts. A focus will be placed on relevant problem formulations for this thesis and important concepts that are central for describing the characteristics and structure of optimization problems and stochastic programs. An overview and fundamental concepts of a decomposition strategy for solving such problems will also be presented.

3.1 Basic optimization concepts

Estimation of either maximum and minimum points is always the goal of mathematical optimization. The core concept is to find the optimal point of a defined domain corresponding to the maximum or minimum value of what is referred as an objective function. In this thesis, the minimization problem will be used to explain the concepts of mathematical optimization. Barton's course notes on Mixed-Integer and Non-Convex Optimization [40] has been used as inspiration in this section.

3.1.1 Optimality

In the standard optimization problem formulation (P.1), the goal is to find the point in a given set $D \in \mathbb{R}^n$ that obtains the minimum value of an objective function $f : \mathbb{R}^n \to \mathbb{R}$.

$$\min_{\boldsymbol{x} \in D} \quad f(\boldsymbol{x}) \tag{P.1}$$

The set D provides the feasible set for the problem, meaning that all feasible points exist in this set. Any point $\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} \notin D$ is infeasible and therefore cannot be the minimum for the optimization problem. The point corresponding to the global minimum, $\boldsymbol{x}^*$, is called the optimal solution of the optimization problem and has to satisfy Equation 3.1.

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*),\ \forall \boldsymbol{x} \in D \tag{3.1}$$

For optimization problems where a global minimum does not exist, a definition of optimality can alternatively be provided through defining an infimum of the problem. The infimum of a set $S \subset \mathbb{R}^n$, $\inf(S)$, is defined as the greatest lower bound on a set of real numbers, meaning that $z \geq \inf(S), \forall z \in S$ and $\inf(S) \geq \beta$, where β is any other lower bound for S. The optimzation problem on a given feasible set $D \in \mathbb{R}$ with an objective function f can then be formulated by

$$\inf_{\boldsymbol{x} \in D} \quad f(\boldsymbol{x}) \tag{P.2}$$

The set $S = \{f(\boldsymbol{x}) : \boldsymbol{x} \in D\}$ is then the set of objective function values at feasible points. However, the set S does not have to contain the optimal objective function value, $\inf(S)$. In contrast to the minimization problem P.1, it is not necessary for the infimum of problem (P.2) to be reached at any point $\boldsymbol{x} \in D$. Conversely, the least upper bound to the set S is defined as the supremum, $sup(S)$.

3.1.2 Relaxation and restriction

Relaxations and restrictions often play central roles in global optimization by providing bounds on the global solution. From the previous definitions in Section 3.1.1 and for a set $D \subset \mathbb{R}^n$, any point $x \in D$ in a function $f(x)$ naturally becomes an upper bound for the minimization problem $\min\{f(x) : x \in D\}$. If we let $D, E \subset \mathbb{R}^n$ and $D \subset E$, then the solution for the minimization of a function on the set E will give a lower bound for the solution of minimizing its subset:

$$\min\{f(\boldsymbol{x}) : x \in E\} \leq \min\{f(\boldsymbol{x}) : \boldsymbol{x} \in D\} \tag{3.2}$$

In the above formulation, the set E has to contain every point existing in D, while the set D only can contain points in E. An optimal solution to the problem on the left hand will therefore be an optimal solution to the problem on the right hand of Equation 3.2 if the set D also contains the optimal solution.

For the same two sets and the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$, the following two optimization problems can be created:

$$\min\{g(\boldsymbol{x}) : \boldsymbol{x} \in E\} \quad \text{and} \quad \min\{f(\boldsymbol{x}) : \boldsymbol{x} \in D\}$$

The left hand problem is then called a relaxation of the right hand problem if D is a subset of E and g is an underestimator of f, i.e. $D \subset E$ and $g(x) \leq f(x), \forall \boldsymbol{x} \in D$. The right hand problem can also be said to be a restriction of the problem on the left side. A solution to relaxations or restrictions can therefore provide lower bound or upper bounds respectively to the original optimization problem.

3.1.3 Constrained optimization

The domain on which the objective function will be optimized plays an important role in the field of mathematical optimization. If the feasible set D equals the entire real domain of $\mathbb{R}^n$ in the standard optimization problem formulation in Problem (P.1), the problem is called unconstrained and any point in $\mathbb{R}^n$ can be the optimal solution provided that a minimum exists. A constrained optimization problem seeks to find a minimum on a defined feasible set of values for each variable. In this case, the domain D will not be equal to $\mathbb{R}^n$ and the optimal solution does not strictly depend on the objective function. The general form of a constrained minimization problem with an objective function $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ is shown in Problem (P.3).

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ s.t \quad & \boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0} \\ & \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0} \\ & \boldsymbol{x} \in X \subset \mathbb{R}^n \end{aligned} \tag{P.3}$$

In the above problem formulation, the vector $\boldsymbol{x}$ is a vector of decision variables to be optimized and X can explicitly describe the feasible points for the problem. The scalar output of the objective function will in addition be affected by the set of inequality and equality constraints, $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^{m_I}$ and $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^{m_E}$ respectively. For a point to be the optimal solution, the point has to be the minimum value of the domain specified by X and the two sets of constraints. Figure 3.1 illustrates how the global minimum of a simple problem $\min\{f(x_1, x_2) : c_1(x_1, x_2) \leq 0\ ,\ c_2(x_1, x_2) \leq 0\ ,\ x_1, x_2 \in X \subset \mathbb{R}^2\}$ is affected when the objective function is subject to constraints. The contour of f illustrates the objective function's minimum value and how the optimal solution to the problem, $\boldsymbol{x}^*$, is constrained from this area by the two constraints.

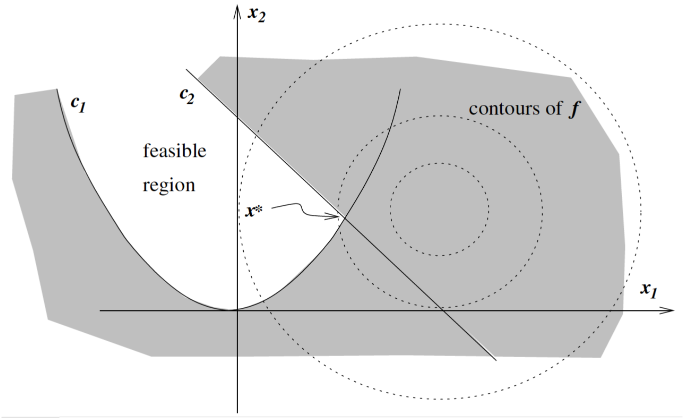


Figure 3.1: Geometrical representation of the feasible set, objective function and optimal solution of a constrained optimization problem [41].

These types of constraints are referred to as hard constraints, meaning that they are requirements for the problem, and as mentioned previously, a solution cannot lie outside of the region. The other types of constraints are said to be soft, which means that they are not definite requirements. Instead the associated variables can be penalized in the objective function if a condition is not met. Soft constraints will therefore not affect the feasible set, but rather how the feasible points are valued.

3.1.4 Convexity

In mathematical optimization, convexity is a central concept for finding global optimal solutions and classifying optimization problems. Characterized by the properties of their feasible sets and objective function, convex optimization problems have the very useful property that every local minimum is guaranteed to be a global minimum.

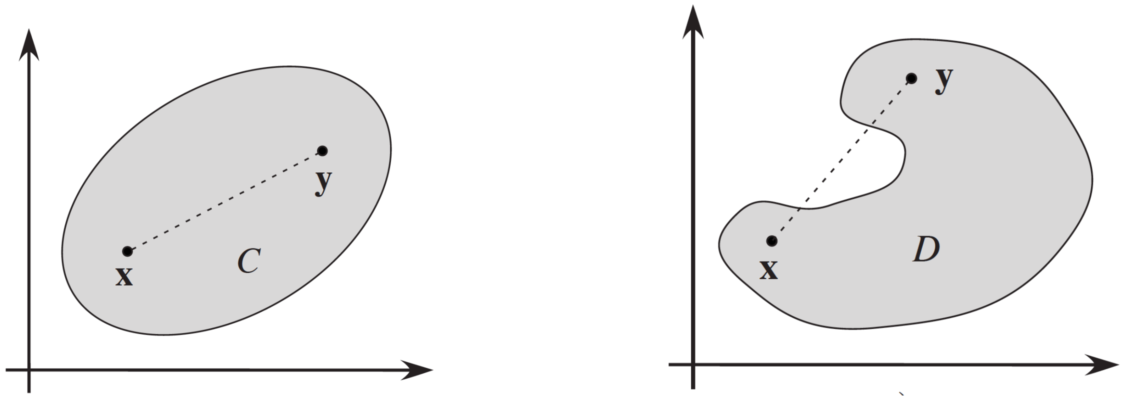
The formulation of a convex optimization problem does not change from a general constrained optimization problem. It is rather determined by the properties of constraints and objective function. For an optimization problem to be convex, both the objective function and the feasible set have to be convex. If either the objective function or the feasible set are non-convex, the problem is said to be non-convex.

Convex sets

Any convex set, C, satisfies that for all $\boldsymbol{x}, \boldsymbol{y} \in C$ and $\boldsymbol{z}$ defined in Equation 3.3, every point in $\boldsymbol{z}$ is also in the set C.

$$\boldsymbol{z} = \lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}, \ \forall \lambda \ \in \ [0, 1] \tag{3.3}$$

A set, C, is called convex if, for any two points in the set, every point on the straight line connecting the two points are also in the set, as shown in Figure 3.2a. Figure 3.2b shows a non-convex set, D, where some of the points on the straight line connecting two points in the set, are outside D. In turn, these non-convex features produce non-convex optimization problems.



(a) Convex set

(b) Non-convex set

Figure 3.2: Convexity of sets [40].

The feasible set in the constrained optimization problem formulation in Problem P.3 is said to be convex if the set D of feasible points, given in Equation 3.4, is convex. This means that all functions creating the set, i.e. $g_j(\boldsymbol{x})\forall j = 1, ..., m_I$ and $h_i(\boldsymbol{x}), \forall i = 1, ..., m_E$, have to be convex.

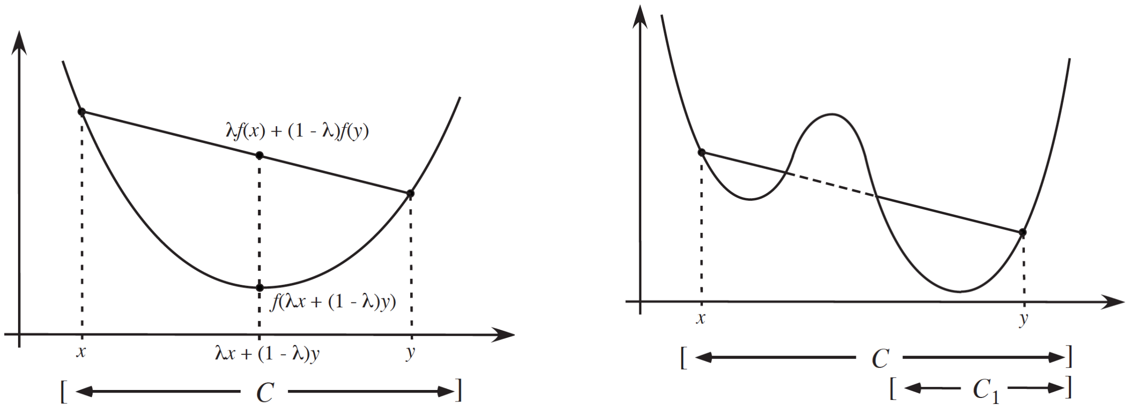
$$D = \{\boldsymbol{x} \in X : g_j(\boldsymbol{x}) \leq 0, \forall j = 1, ..., m_I; h_i(\boldsymbol{x}) = 0, \forall i = 1, ..., m_E\} \tag{3.4}$$

Convex functions

The curvature of the graph is what defines if a function is convex or non-convex. A function is convex if, for any two points on the graph, the straight line connecting the points lies entirely above or on the graph, as shown in Equation 3.5.

$$f(\lambda\boldsymbol{x} + (1 - \lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1 - \lambda)f(\boldsymbol{y}), \quad \forall\lambda \in [0, 1], \quad \forall\boldsymbol{x}, \boldsymbol{y} \in C \tag{3.5}$$

Where $f : C \to \mathbb{R}$ and $C \subset \mathbb{R}^n$ is a convex set. A strictly convex function uses a hard inequality in the same equation for $\boldsymbol{x} \neq \boldsymbol{y}$. Figure 3.3a shows a strictly convex function where the straight line lies entirely above the graph. A non-convex function is illustrated in Figure 3.3b, where a part of the straight line lies below the graph of the function. It is worth noting that a non-convex function can be convex on a subset of its original set, provided that the subset is chosen correctly. As an example, the set in Figure 3.3a can be viewed as a subset of the set in Figure 3.3b, if both figures illustrates the graph of the same function.



(a) Convex function

(b) Non-convex function

Figure 3.3: Convexity of functions [40].

Hyperplanes

Hyperplanes are useful concepts when dealing with non-convex global optimization. A hyperplane, H, is defined by a normal vector, $\boldsymbol{a} \in \mathbb{R}^n$, and a constant, $b \in \mathbb{R}$. For vectors of variables, $\boldsymbol{x} \in \mathbb{R}^n$, the affine equation defining a hyperplane is given in Equation 3.6.

$$H = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} = b\} \tag{3.6}$$

Hyperplanes divides a space into two parts, which are called halfspaces. The positive halfspace of a hyperplane is defined in Equation 3.7.

$$H^+ = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^T\boldsymbol{x} \geq b\} \tag{3.7}$$

Hyperplanes are called supporting hyperplanes to convex sets if they lie at the boundary of the convex sets without crossing the boundary. For non-convex sets, supporting hyperplanes may not exist.

3.1.5 Duality theory

Duality theory builds on a principle that optimization can be viewed from two different perspectives. These two perspectives are the primal problem and the dual problem.

For a minimization problem on $X \subseteq \mathbb{R}^n$ with an objective function $f : X \to \mathbb{R}$ and a set of inequality constraints $\boldsymbol{g} : X \to \mathbb{R}^m$, the primal problem (PRIMAL) is given below:

$$\begin{aligned} \min_{\boldsymbol{x}\in X} \quad & f(\boldsymbol{x}) \\ s.t \quad & \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0} \end{aligned} \tag{PRIMAL}$$

The dual problem is then created by using the Lagrangian function for the optimization problem, $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\mu})$ given in Equation 3.8. Through the Lagrangian, the constraints are added to the objective function by using the Lagrangian multipliers, $\boldsymbol{\mu}$.

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \mu_i g_i(\boldsymbol{x}) \tag{3.8}$$

The value f^* is defined to be a feasible value of the problem $\inf\{f(\boldsymbol{x}) : \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}, \boldsymbol{x} \in X\}$. According to duality theory, a Lagrangian multiplier, $\boldsymbol{\mu}^*$ is a duality multiplier if $\boldsymbol{\mu}^* \geq \boldsymbol{0}$ and $f^* = \inf\{\mathcal{L}(\boldsymbol{x}, \boldsymbol{\mu}^*) : \boldsymbol{x} \in X\}$.

The set $S \subset \mathbb{R}^{m+1}$ is called the image set of the primal problem. An image set is defined by the values that are taken on by the constraints and the objective function, as given in Equation 3.9. This concept can be referred to as mapping an image set, illustrated in Figure 3.4 with $m = 1$.

$$S = \{(\boldsymbol{g}(\boldsymbol{x}), f(\boldsymbol{x})) : \boldsymbol{x} \in X\} \tag{3.9}$$

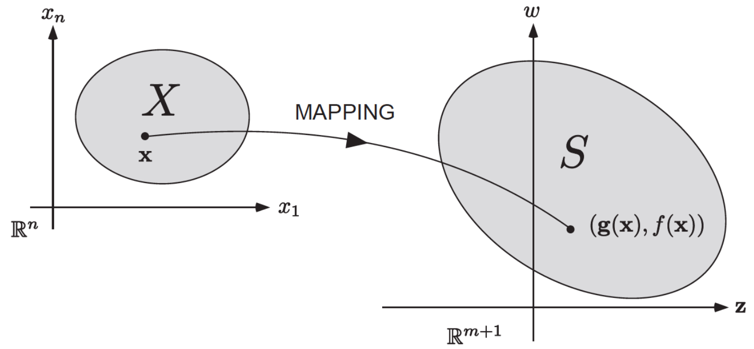


Figure 3.4: Mapping an image set, S, of the primal problem [40].

A hyperplane, H, is defined by the normal vector $(\boldsymbol{\mu}, 1) \in \mathbb{R}^{m+1}$ and a constant c in the space $(\boldsymbol{z}, w) \in \mathbb{R}^{m+1}$. If this hyperplane passes through S as previously described, it follows from Equation 3.10 that H intersects the vertical axis at $w = \mathcal{L}(\boldsymbol{x}, \boldsymbol{\mu})$.

$$\boldsymbol{\mu}^T \boldsymbol{z} = c = \boldsymbol{\mu}^T \boldsymbol{g}(\boldsymbol{x}) + f(\boldsymbol{x}) \tag{3.10}$$

The hyperplane, H, contains the image set, S, in its positive halfspace if $\inf\{\mathcal{L}(\boldsymbol{x}, \boldsymbol{\mu}) : \boldsymbol{x} \in X\} \geq c$. In Figure 3.5, the hyperplanes defined by $\boldsymbol{\mu}$ is illustrated. It is seen that $\boldsymbol{\mu}$ is the only duality multiplier and that f^* is the smallest value for w that is feasible.

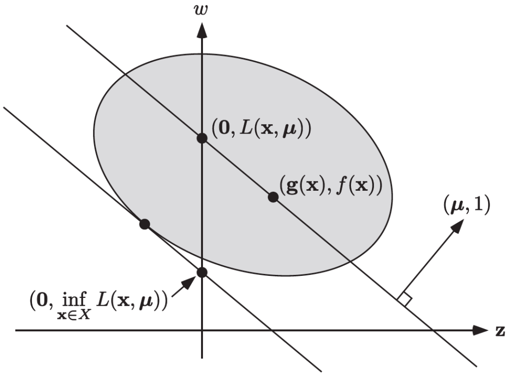

w
$(\mathbf{0}, L(\mathbf{x}, \boldsymbol{\mu}))$
$(\mathbf{g}(\mathbf{x}), f(\mathbf{x}))$
$(\boldsymbol{\mu}, 1)$
$(\mathbf{0}, \inf_{\mathbf{x} \in X} L(\mathbf{x}, \boldsymbol{\mu}))$
$\mathbf{z}$

Figure 3.5: Hyperplanes through the image set, S, and the duality multipliers, $\boldsymbol{\mu}^*$ [40].

The duality multipliers that defines the hyperplane which gives the smallest value of w while still holding the entire image set S in its positive halfspace will give the solution to the dual formulation in Problem (DUAL).

$$\begin{aligned} \sup_{\boldsymbol{\mu} \in M} \quad & q(\boldsymbol{\mu}) \\ s.t \quad & \boldsymbol{\mu} \geq \mathbf{0} \end{aligned} \tag{DUAL}$$

where the set $M = \{\mu \in \mathbb{R}^m : \inf_{\boldsymbol{x} \in X} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\mu}) \geq -\infty\}$ and $q : M \to \mathbb{R}$ is the dual function given by $q(\boldsymbol{\mu}) = \inf_{\boldsymbol{x} \in X} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\mu})$. The solution to the dual problem then gives the best lower bound to the primal problem.

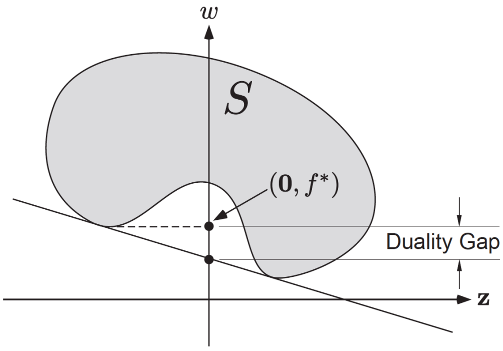


Figure 3.6: Duality gap following non-convexity [40].

However, duality requires convex primal problems in order to generate a hyperplane that holds the entire image set, S, in its positive half-space and intersects the vertical axis at the same point as S. If either the objective function or the feasible set has non-linear relations that cause non-convexity, the image set may create a duality gap for the dual problem, as illustrated in Figure 3.6.

3.2 Mixed-integer programming

In standard convex optimization, there is usually no restriction on which values the decision variables can take other than being within the specified range of values and satisfying constraints. However, optimization problems will in many cases involve constricting decision variables by defining discrete sets of values. In integer programming, a special branch of optimization problems, all decision variables are restricted to be integers. In addition to being beneficial for modelling quantities that can only be integers, integer programming can be very useful for representing yes/no-decisions.

These types of decisions can be modelled using the special case of integer programming, called binary integer programming. A vector of size n_z of binary decision variables $\boldsymbol{z} \in \{0,1\}^{n_z}$ will be restricted to two integer values, where the value 1 can indicate if an event should happen while the value 0 will indicate that an event should not happen.

An optimization problem where only some of the variables are constrained to be integers or binary integers, is called a mixed-integer programming (MIP) problem. The decision variables can be discrete or continuous, and MIP formulations can be very useful when modeling real-life problems. The formulation of a MIP problem with integer and continuous variables is given in Problem (MIP).

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{z}} \quad & f(\boldsymbol{x},\boldsymbol{z}) \\ \text{s.t.} \quad & \boldsymbol{g}(\boldsymbol{x},\boldsymbol{z}) \leq \boldsymbol{0}, \\ & \boldsymbol{h}(\boldsymbol{x},\boldsymbol{z}) = \boldsymbol{0} \\ \boldsymbol{x} \in X \subset \mathbb{R}^{n_x}&, \boldsymbol{z} \in Z \subset \mathbb{Z}^{n_z} \end{aligned} \tag{MIP}$$

where $f : \mathbb{Z}^{n_z} \times \mathbb{R}^{n_x} \to \mathbb{R}$, $\boldsymbol{g} : \mathbb{Z}^{n_z} \times \mathbb{R}^{n_x} \to \mathbb{R}^{m_I}$, $\boldsymbol{h} : \mathbb{Z}^{n_z} \times \mathbb{R}^{n_x} \to \mathbb{R}^{m_E}$ and m_I and m_E are the number of inequality and equality constraints. The decision variables $\boldsymbol{z}$ and $\boldsymbol{x}$ are integer and continuous respectively, where $\boldsymbol{z}$ can be both regular integer and binary. Additionally, two subclasses of MIPs are often used for describing the problems. If all the constraints and the objective function are linear, the problem is called a mixed-integer linear program (MILP). On the other hand, any non-linear constraints or objective function will make the problem a mixed-integer non-linear program (MINLP).

3.3 Stochastic Programming

Uncertainty is inherently a part of real-life problems. In the field of optimization, this uncertainty can prove to be decisive in reaching optimality. Modeling of uncertainty involves the inclusion of uncertain parameters in the optimization model where each parameter has a range of possible values. Not accounting for several realizations of an uncertain parameter may yield solutions very sensitive to variations or downright sub-optimal solutions. Stochastic programming is a framework for accounting for uncertain parameters in optimization models.

3.3.1 Two-stage stochastic programming

In the two-stage stochastic program framework, decisions have to be made in two separate stages. In the first-stage, a set of decisions has to be made irrespective of the actual realization of uncertainty. The decisions in the first-stage will then be common for all decisions made in the second-stage, which are then made with full information. The uncertain parameters are assumed to have a fixed number of realizations with a known probability, which in turn will be used to construct scenarios for the program.

The goal of stochastic programming is therefore to minimize the expected value of the objective function by accounting for all possible scenarios and their respective probabilities. This problem is a recourse problem as the recourse actions in the second stage can be viewed as corrective actions based on the revealed scenario. The objective value of the recourse problem (RP) is found by the minimizing the sum of the value of the first stage and the expected value of the second stage.

With $\boldsymbol{y} \in Y \subset \mathbb{R}^{n_y}$ and $\boldsymbol{x} \in X \subset \mathbb{R}^{n_x}$ as the first and second stage decision variables respectively, while uncertainty is represented by the vector $\boldsymbol{\xi}$, the formulation of a two-stage stochastic program can be written as in Problem (SP.1) and (SP.2). n_y and n_x represent the number of first and second stage decision variables respectively.

$$\begin{aligned} RP = \quad \min_{\boldsymbol{y}} \quad & f^{(1)}(\boldsymbol{y}) + \mathbb{E}_{\boldsymbol{\xi}}[\boldsymbol{R}(\boldsymbol{y}, \boldsymbol{\xi})], \\ \text{s.t.} \quad & \mathbf{g}^{(1)}(\boldsymbol{y}) \leq \mathbf{0} \\ & \boldsymbol{h}^{(1)}(\boldsymbol{y}) = \mathbf{0} \\ & \boldsymbol{y} \in Y \end{aligned} \tag{SP.1}$$

where $f^{(1)} : \mathbb{R}^{n_y} \to \mathbb{R}$ is the objective function contribution from solely first stage decisions, $\boldsymbol{g} : \mathbb{R}^{n_y} \to \mathbb{R}^{m_I^{(1)}}$ and $\boldsymbol{h} : \mathbb{R}^{n_y} \to \mathbb{R}^{m_E^{(1)}}$ are the first stage constraints, and $m_I^{(1)}$ and $m_E^{(1)}$ denote the number of first-stage constraints. $\mathbb{E}_{\boldsymbol{\xi}}$ denotes the expected value over the number of scenarios and $\boldsymbol{R}(\boldsymbol{y}, \boldsymbol{\xi})$ contains the recourse functions. $R_h(\boldsymbol{y}, \boldsymbol{\xi}_h)$ gives the optimal value of the second stage of the problem for a scenario h with given first stage decisions, $\boldsymbol{y}$, and uncertainty realizations for the scenario, $\boldsymbol{\xi}_h$:

$$\begin{aligned} R_h(\boldsymbol{y}, \boldsymbol{\xi}_h) := \quad \min_{\boldsymbol{x}} \quad & f_h^{(2)}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi}_h) \\ \text{s.t.} \quad & \mathbf{g}^{(2)}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi}_h) \leq \mathbf{0} \\ & \mathbf{h}^{(2)}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi}_h) = \mathbf{0} \\ & \boldsymbol{x} \in X(\boldsymbol{\xi}_h) \end{aligned} \tag{SP.2}$$

where $f_h^{(2)} : \mathbb{R}^{n_x \times n_y} \to \mathbb{R}$ is the objective function contribution from both first and second stage decisions in scenario h, $\boldsymbol{g} : \mathbb{R}^{n_x \times n_y} \to \mathbb{R}^{m_I^{(2)}}$ and $\boldsymbol{h} : \mathbb{R}^{n_x \times n_y} \to \mathbb{R}^{m_E^{(2)}}$ are the second stage constraints, and $m_I^{(2)}$ and $m_E^{(2)}$ denote the number of second-stage constraints. The recourse function can give a different value for all scenarios and recourse contribution can depend on the probability of the scenario. The expected value of the second-stage for all scenarios can be found by using their respective probability, as shown in Equation 3.11.

$$\mathbb{E}_{\boldsymbol{\xi}}[\boldsymbol{R}(\boldsymbol{y}, \boldsymbol{\xi})] = \sum_{h=1}^{N_S} \omega_h R_h(\boldsymbol{y}, \boldsymbol{\xi}_h). \tag{3.11}$$

Here ω_h is the probability of scenario h, $\boldsymbol{\xi}_h$ is the realized uncertainty in scenario h, and N_S is the finite number of discrete scenarios.

The structure of a probabilistic two-stage stochastic program is illustrated in Figure 3.7. The first stage decisions are independent of which scenario actually happens, while the second stage decisions then will depend on parameter realizations and decisions in the first stage.

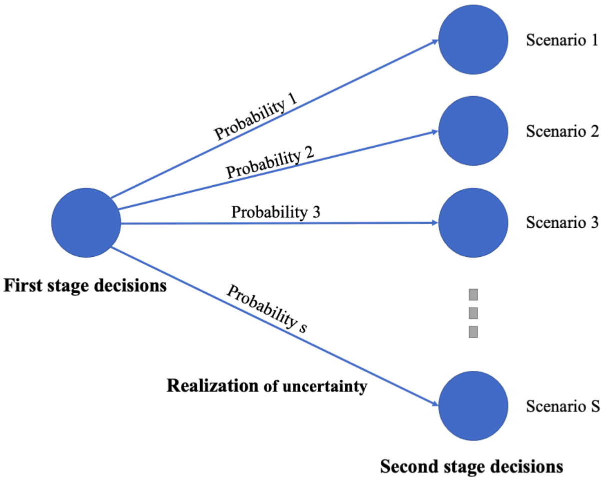


Figure 3.7: Scenario tree for a two-stage stochastic program.

With $\boldsymbol{x}_h$ now as the decision variables for scenario h, the single-level, deterministic equivalent formulation of the two-stage stochastic program, Problem (SP), can be obtained by combining the formulations in Problems (SP.1) and (SP.2), and Equation 3.11. Problems (SP.1) and (SP.2) can be viewed as the projections of Problem (SP).

$$\begin{aligned} \underset{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_{N_S},\boldsymbol{y}}{min} \quad & f^{(1)}(\boldsymbol{y}) + \sum_{h=1}^{S} \omega_h f_h^{(2)}(\boldsymbol{x}_h, \boldsymbol{y}, \boldsymbol{\xi}_h) \\ \text{s.t.} \quad & \mathbf{g}^{(1)}(\boldsymbol{y}) \leq \mathbf{0} \\ & \mathbf{h}^{(1)}(\boldsymbol{y}) = \mathbf{0} \\ & \boldsymbol{y} \in Y \\ & \mathbf{g}_h^{(2)}(\boldsymbol{x}_h, \boldsymbol{y}, \boldsymbol{\xi}_h) \leq \mathbf{0}, \qquad \forall h \in \{1, \cdots, N_S\} \\ & \mathbf{h}_h^{(2)}(\boldsymbol{x}_h, \boldsymbol{y}, \boldsymbol{\xi}_h) = \mathbf{0}, \qquad \forall h \in \{1, \cdots, N_S\} \\ & \boldsymbol{x}_h \in X(\boldsymbol{\xi}), \qquad \forall h \in \{1, \cdots, N_S\} \end{aligned} \tag{SP}$$

3.3.2 Value of the stochastic solution

Next, a few central concepts related to stochastic programming will be addressed to evaluate the benefit of optimizing under uncertainty.

- The expected value problem (EVP) is the deterministic approach to optimization without accounting for uncertainty, assuming only one realization of each uncertain parameter. The realizations are assumed to be the expected value of each parameter, also referred to as the mean value of the realizations. The problem can be viewed as a stochastic program with only one scenario where the first stage decisions are optimized for the expected value scenario.

- The evaluation of the expected value solution (EEV) is the value of expectation of the first-stage decisions found in the EVP when used in a stochastic environment.

- The recourse problem value (RP) is the optimal objective function value of the recourse problem obtained by accounting for uncertainty when optimizing the first stage decisions.

- The value of the stochastic solution (VSS) is the added value by accounting for uncertainty in the 1st stage decision process. The value is found by the difference between the EEV and RP:
$$VSS = EEV - RP \tag{3.12}$$

3.4 Decomposition algorithms

Decomposition approaches can be useful for solving two-stage stochastic programs as the number of scenarios can drive an exponential growth in computational complexity [13]. Divide-and-conquer strategies can be employed by creating and solving sub-problems to more efficiently obtain a solution for the original problem. Generalized Benders decomposition (GBD) is an algorithm that can be used for solving MIPs of the form in Problem (MICP).

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y}} \quad & f(\boldsymbol{x},\boldsymbol{y}) \\ \text{s.t.} \quad & \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) \leq \boldsymbol{0}, \\ & \boldsymbol{x} \in X \subset \mathbb{R}^{n_x}, \boldsymbol{y} \in Y \subset \mathbb{R}^{n_y} \end{aligned} \tag{MICP}$$

where $\boldsymbol{y}$ are the variables that complicates the optimization process.

GBD can be used for solving the above problem when fixing $\boldsymbol{y}$ reveals a special block structure, creating independent sub-problems with different sub-vectors of $\boldsymbol{x}$. Additionally, GBD requires a convex program when the complicating variables are held constant. For linear programs, i.e. when f and all functions in $\boldsymbol{g}$ are linear, the problem turns to an MILP, on which the classical Benders decomposition (BD) can be used [13]. It is noted that both BD and GBD also are applicable to programs with continuous complicating variables, provided a decomposable structure of independent sub-problems.

The decomposable block structure with submatrices that is needed for the Benders decomposition strategies is illustrated in Figure 3.8. In this figure: $\boldsymbol{x}$ are complicating variables and $\boldsymbol{y}_\omega$ are the independent variables for sub-problem ω; and $\boldsymbol{T}_\omega$ and $\boldsymbol{W}_\omega$ are the matrices for the variables for each sub-problem. The block structure is exploited by iteratively exploring and cutting off solutions by solving sub-problems. Two-stage stochastic programs where scenarios can be regarded as independent are examples of optimization problems in which the decomposable structure is present. Fixing the complicating variables, i.e. first stage variables, will turn the problem into different scenario-wise independent problems.

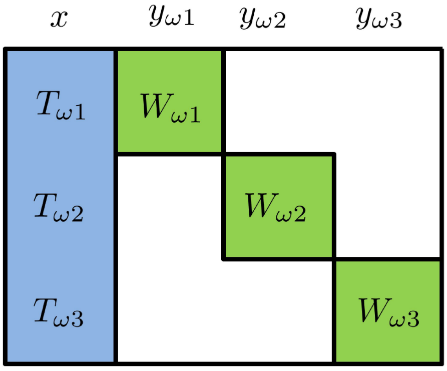


Figure 3.8: Idea behind Benders decomposition [13].

For problems of the form as in Problem (MICP), with non-convex constraints in the block independent variables, are referred to as non-convex mixed-integer non-linear problems (MINLPs). Such problems are known to be hard to guarantee a global solution.

The non-convex generalized Benders decomposition (NGBD) is a non-convex extension of GBD, which can guarantee global optimality under certain requirements. The core of the concept of the NGBD algorithm is to create improved upper and lower bounds on the original problem by iteratively solving bounding problems until convergence:

- The upper bound is found by a primal problem (PP), created by restriction of the original problem through fixing realizations of the complicating variables to particular values.
- A lower bounding problem (LBP) is created by performing convex relaxations of non-convex functions in the original problem.

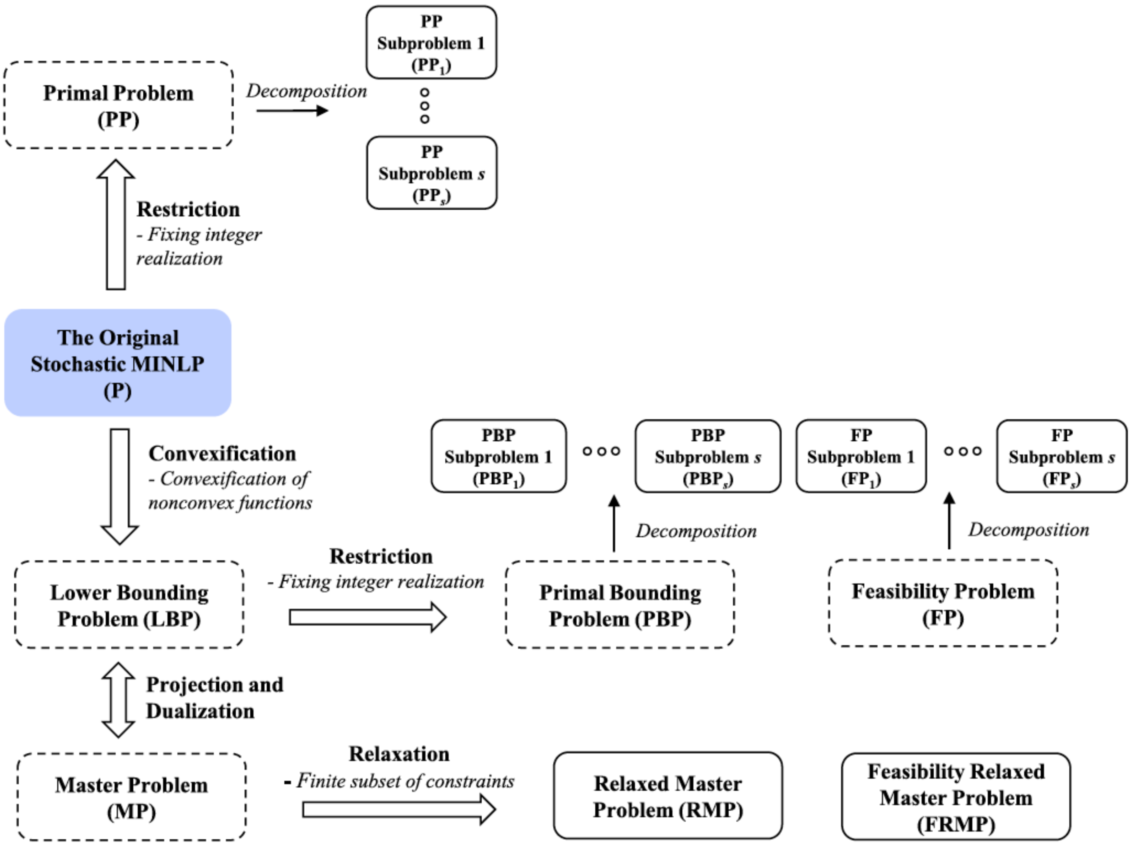

Primal Problem (PP)
Decomposition
PP Subproblem 1 (PP$_1$)
PP Subproblem s (PP$_s$)
Restriction
- Fixing integer realization
The Original Stochastic MINLP (P)
Convexification
- Convexification of nonconvex functions
PBP Subproblem 1 (PBP$_1$)
PBP Subproblem s (PBP$_s$)
FP Subproblem 1 (FP$_1$)
FP Subproblem s (FP$_s$)
Decomposition
Decomposition
Restriction
- Fixing integer realization
Lower Bounding Problem (LBP)
Primal Bounding Problem (PBP)
Feasibility Problem (FP)
Projection and Dualization
Relaxation
- Finite subset of constraints
Master Problem (MP)
Relaxed Master Problem (RMP)
Feasibility Relaxed Master Problem (FRMP)

Figure 3.9: Overview of the NGBD algorithm [42].

An overview of the NGBD algorithm is presented in Figure 3.9, and can be conceptually explained by:

1. The upper bounding problem (PP), obtained by restrictions, is solved by decomposition; i.e. solving independent non-linear sub-problems.
2. The lower bounding problem (LBP) is constructed by replacing all the non-convex participating functions of Problem (P) with their corresponding convex relaxations to give a two-stage stochastic MICP. The solution to the (LBP) is obtained by GBD:

    i A master problem (MP) is constructed by projection onto the space of complicating variables and dualization of the (LBP).

    ii The master problem is relaxed by including only a finite subset of constraints.

    iii Solution to the relaxed master problem (RMP) provides a lower bound to (LBP), improved by iteratively adding constraints to the (RMP).

    iv The upper bound of the (LBP) is found by fixing complicating variables in a primal bounding problem (PBP) to solutions of the (RMP) that is providing lower bounds.

To obtain convergence of the NGBD, previously visited solutions are cut off by adding affine constraints and improving the upper and lower bounds are sufficiently close.

3.5 GOSSIP software

GOSSIP is a software framework for modelling and solving non-convex two-stage stochastic mixed-integer non-linear programs [15]. GOSSIP is applicable to solving both two-stage stochastic MILPs and MICPs, i.e. convex mixed-integer problems. However, the GOSSIP software has an implementation of the NGBD algorithm. Thus, GOSSIP can be used to obtain a guaranteed global minimum of non-convex MINLPs under certain requirements, provided a feasible problem.

For non-convex two-stage stochastic MINLPs and solution by NGBD, GOSSIP requires a scenario-wise decomposable structure and the problem formulation given in Problem (MINLP)

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y}} \quad & f(\boldsymbol{x},\boldsymbol{y}) \\ \text{s.t.} \quad & \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) \leq \boldsymbol{0}, \\ \boldsymbol{x} \in X \subset \mathbb{R}^{n_x}&, \boldsymbol{y} \in Y \subset \mathbb{R}^{n_y} \end{aligned} \tag{MINLP}$$

where $\boldsymbol{y}$ now is a set of discrete first-stage variables and $\boldsymbol{x}$ is a vector of continuous second-stage variables.

The key strength of GOSSIP is the possibility of using decomposition approaches for efficiently solving large problems. However, there is also an option to solve the full-space problem through a linked version of ANTIGONE [43].

4 | Renewable Energy System Model

The flexible design problem of renewable energy systems can be modelled using the two-stage stochastic programming framework for efficient solutions in GOSSIP. In addition to a simple case study with model inputs, this chapter presents the background, the methodology and the complete formulation for a RES model consisting of solar power, wind power and battery energy storage.

4.1 Modelling of a renewable energy system

Modelling the optimal flexible design problem requires attention to the RES model choices and decisions. In this section, the modelling approaches for the physical system, uncertainty and components are presented.

4.1.1 Energy hub

The renewable energy system to be studied can be viewed using the energy hub concept [17]. The concept involves looking at the components of the system as an interface between energy carriers and demands to be covered by the system [44]. The components in the interface will then be used for transformations and conversions between energy carriers and storage of energy carriers to optimally cover a demand load.

Energy carriers are transported across the energy hub boundary, with both inputs and outputs of energy. Note that energy sources such as solar and wind energy are regarded as energy carriers when using the concept. Energy carriers such as electricity, solar radiation and wind can be regarded as inputs, and may be converted in the energy hub into other types of energy, for instance from solar radiation to electricity. The energy, often heat or electricity, can either be used to cover a load, stored for later use through energy storage technologies, or exported. The energy hub concept is illustrated in Figure 4.1 for a system consisting of solar PV, wind turbines and batteries with electricity as the main energy carrier. In addition to be expandable to other technologies and energy carriers, the concept allows for connections with other energy hubs, creating energy networks [44].

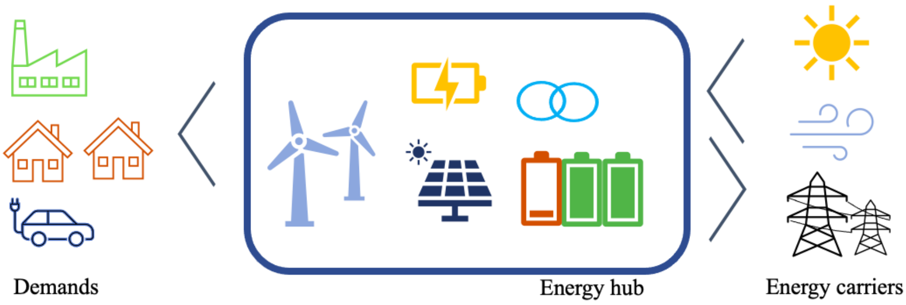


Figure 4.1: Concept of an energy hub.

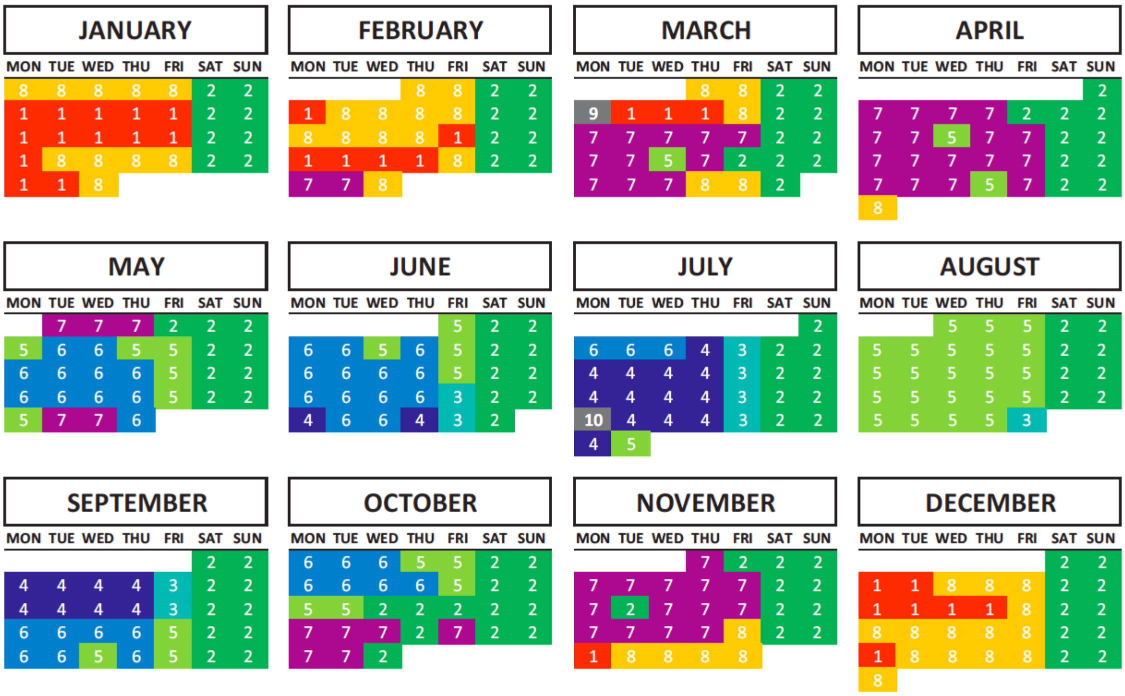
4.1.2 Uncertainty modelling and design day approach

In renewable energy systems, many of the parameters can be considered as uncertain. However, the largest portion of variation in optimization results due to uncertainty can often be characterized by a few parameters. In a global sensitivity analysis of parameters in a distributed energy system [45], three aspects were identified as particularly crucial:

- The variation in energy demand
- Energy carrier price variation
- Fluctuating weather patterns

Investment costs and most of the technical aspects of conversion technologies were found to be considerably less significant to the optimal solution, allowing them to be considered fixed. For solar panels and wind turbines, the weather patterns that have the largest influence on power generation are the irradiation and the wind speed respectively, as presented previously in Chapter 2.

The variation of the uncertain parameters are modeled to be discrete, i.e. it is assumed the uncertain parameters can take on finite numbers of realizations. Each combination of possible parameter realizations will then produce a scenario for the program, resulting in a finite set of scenarios. Each scenario is modelled using the design day approach, where sets of typical parameter fluctuations are constructed for a 24-hour period. The combination of these sets, or design days, will then represent the uncertainty and variability the system is subject to. For instance, a design day can be constructed by a single combination of one daily pattern for each load curve, solar radiation, wind speed and electricity price. The typical design days are created to represent uncertain and seasonal variation in a full year, as illustrated in Figure 4.2.



JANUARY

MON	TUE	WED	THU	FRI	SAT	SUN
8	8	8	8	8	2	2
1	1	1	1	1	2	2
1	1	1	1	1	2	2
1	8	8	8	8	2	2
1	1	8				

FEBRUARY

MON	TUE	WED	THU	FRI	SAT	SUN
			8	8	2	2
1	8	8	8	8	2	2
8	8	8	8	1	2	2
1	1	1	1	8	2	2
7	7	8				

MARCH

MON	TUE	WED	THU	FRI	SAT	SUN
			8	8	2	2
9	1	1	1	8	2	2
7	7	7	7	7	2	2
7	7	5	7	2	2	2
7	7	7	8	8	2	

APRIL

MON	TUE	WED	THU	FRI	SAT	SUN
						2
7	7	7	7	2	2	2
7	7	5	7	7	2	2
7	7	7	7	7	2	2
7	7	7	5	7	2	2
8						

MAY

MON	TUE	WED	THU	FRI	SAT	SUN
	7	7	7	2	2	2
5	6	6	5	5	2	2
6	6	6	6	5	2	2
6	6	6	6	5	2	2
5	7	7	6			

JUNE

MON	TUE	WED	THU	FRI	SAT	SUN
				5	2	2
6	6	5	6	5	2	2
6	6	6	6	5	2	2
6	6	6	6	3	2	2
4	6	6	4	3	2	

JULY

MON	TUE	WED	THU	FRI	SAT	SUN
						2
6	6	6	4	3	2	2
4	4	4	4	3	2	2
4	4	4	4	3	2	2
10	4	4	4	3	2	2
4	5					

AUGUST

MON	TUE	WED	THU	FRI	SAT	SUN
		5	5	5	2	2
5	5	5	5	5	2	2
5	5	5	5	5	2	2
5	5	5	5	5	2	2
5	5	5	5	3		

SEPTEMBER

MON	TUE	WED	THU	FRI	SAT	SUN
					2	2
4	4	4	4	3	2	2
4	4	4	4	3	2	2
6	6	6	6	5	2	2
6	6	5	6	5	2	2

OCTOBER

MON	TUE	WED	THU	FRI	SAT	SUN
6	6	6	5	5	2	2
6	6	6	6	5	2	2
5	5	2	2	2	2	2
7	7	7	2	7	2	2
7	7	2				

NOVEMBER

MON	TUE	WED	THU	FRI	SAT	SUN
			7	2	2	2
7	7	7	7	7	2	2
7	2	7	7	7	2	2
7	7	7	7	8	2	2
1	8	8	8	8		

DECEMBER

MON	TUE	WED	THU	FRI	SAT	SUN
					2	2
1	1	8	8	8	2	2
1	1	1	1	8	2	2
8	8	8	8	8	2	2
1	8	8	8	8	2	2
8						

Figure 4.2: Design days representing uncertain and seasonal variation [46].

4.1.3 Modelling of energy technologies

As previously discussed, renewable energy systems are generally complex and several simplifying assumptions have to made. Non-linearity in any optimization model is important to evaluate. Non-linear models lead to more complex problems as they have a tendency to lead to non-convexity.

Even though tools, such as NGBD, are available for finding solutions to non-convex problems, the computational complexity is significantly increased. A close look at the assumptions and approximations are imperative for an effective program.

In this work, the behavior of battery energy storage is approximated by a linear battery model, i.e. power flow in and out of the battery are linearly related to changes in energy stored in the battery. However, as discussed in Sections 2.3, the behaviour of battery energy storage is generally characterized by non-linearity. The most important effects for modelling purposes of the battery non-linearities can summarized by:

- Higher battery power flows causes a reduction in energy efficiency.
- Deep discharge, high states of charge and high battery power flows speed up cell degradation rates. This causes an increased power fading and capacity loss, and reduced future energy efficiency and cycle life.

Deep discharge and high power flows will be prevented to represent mitigation of the effects of degradation and restrict efficiency losses. Non-linear battery models are assumed to be more accurate for representation of these aspects, but produce non-linear relations between program decision-variables. Non-linear battery models are therefore expected to produce non-convex problems, which is undesirable when it comes to computational complexity. If precision of the solution is not significantly compromised, such non-linear models should be avoided.

Linear models from a program's perspective are also used for conversion of solar radiation and wind to electricity, i.e. only linear relations between decision-variables. However, the models may have non-linearities for fixed and uncertain parameters. While solar panel output may be approximated by conversion of radiation through a constant efficiency, the power output of wind farms has non-linear relation to wind speed with different production domains, as discussed in Section 2.2. On the other hand, linearity of the optimization model will be maintained.

Furthermore, it is noted that while the performances of PV systems and batteries are dependent on temperature, as discussed in Sections 2.1 and 2.3, the aspect of temperature is omitted in this work.

4.2 Formulating the flexible design problem using GOSSIP

The optimization software GOSSIP, presented in Section 3.5, provides a framework for two-stage stochastic programs. In this section, the formulation implications of the flexible design problem of energy systems by using the GOSSIP modelling framework are presented.

4.2.1 Decision-making structure

The GOSSIP modelling framework requires compatibility with a two-stage stochastic program. In Section 3.3, the decision-making structure of two-stage stochastic programs was presented, allowing two sets of decisions to be made in two stages separated by the realization of uncertainty. This decision-making structure can be applied to the flexible design problem of renewable energy systems by identifying decisions as either design or operational decisions:

- First-stage decisions refer to the design and planning phase of the RES. Before any uncertainty has been revealed, decisions related to choosing and sizing technologies have to be made.
- Second-stage variables represent the operational aspects of the RES after the uncertainties and variations have been revealed. After the design decisions have been set, the RES will operate differently for the different cases of revealed uncertainty.

4.2.2 Time dependency

Often in renewable energy systems, the program may be subject to variables or uncertainties that are dependent on what has happened previously. For instance, optimal operational decisions depend on a battery's state of charge, which in turn depends on previous operational decisions. For inclusion of some uncertainty in the operational aspect, the decision-making structure then requires all decisions for previous time steps to be made in advance before consideration of the decisions for the current time step. Consequently, the decision-making process of a RES gives a multi-stage stochastic program. The first stage would represent the design decisions, where the following stages represents a time step.

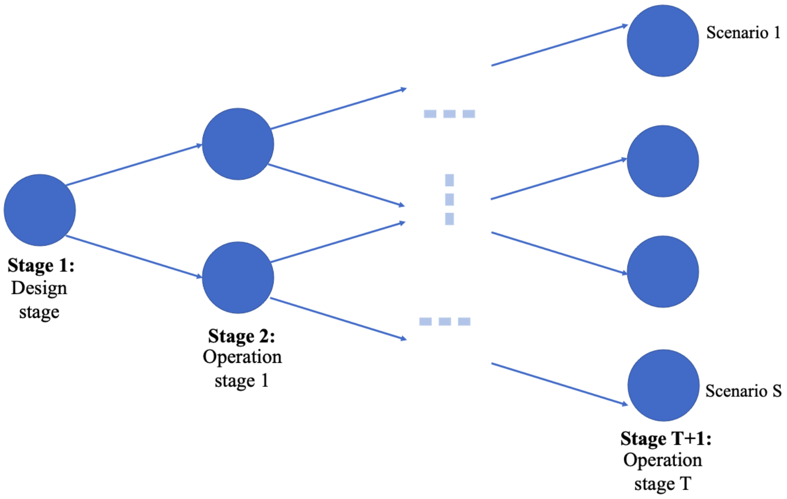


Figure 4.3: Scenario tree for a RES multi-stage stochastic program.

The scenario tree for a RES multi-stage stochastic programs is presented in Figure 4.3, where each stage requires a set of decisions to be made. The multi-stage nature and the rapid increase in scenarios and decision variables significantly complicate the optimization process. Moreover, multi-stage stochastic programming is widely acknowledged to be computationally intractable [15].

As discussed in the previous sections, the GOSSIP software framework requires the model formulation to be compatible with a two-stage stochastic program. This means that all decisions involving time steps, i.e. operational decisions, have to be made in the same stage. Thereby, it is assumed that uncertainty is only revealed once, i.e. the same information will be available when making decisions in the first time step as when the final time step-decisions are made.
Figure 4.4 illustrates the concept of multiperiods in the scenario tree of a two-stage stochastic program.

When using a two-stage multiperiod formualtion instead of a multi-stage formulation, information about the realization of the uncertain parameters for all time steps is available at once. Thus, second-stage decisions are optimized with perfect information of future time steps. Although this assumption is not realistic, it is necessary for computational tractability.

4.2.3 Decoupling of design days using periodicity for state variables

Using the design day approach with hourly temporal resolution, the scenario multiperiods are represented by design days with 24 timesteps. For the flexible design problem, the evaluation of optimal design will then be based on the expected values for 24 hours of operation in the design days. In reality, the design days are coupled by their boundaries, i.e. initial and final timestep of different design days will be linked. The sequence of design days is modelled using a two-stage stochastic program formulation, shown in Figure 4.5. However, the actual sequence of days is

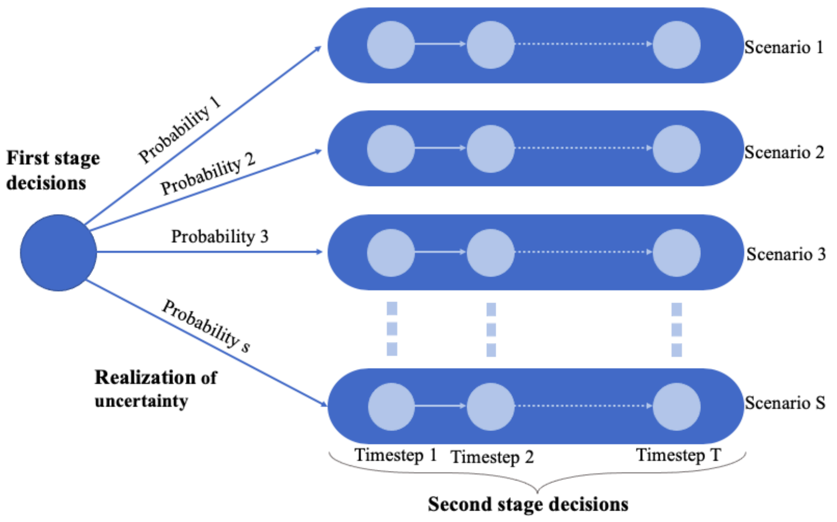


Figure 4.4: Scenario tree for a two-stage stochastic multiperiod program.

uncertain, making it difficult to model the coupling in a two-stage structure.

With the possible inclusion of energy storage in the design, the dynamic aspects and state variables of the model become important. In this work, periodic boundary conditions are applied for the design days, i.e. initial states and final states of all design days are set to the same fixed value. This essentially performs a decoupling, producing independent design days. Thereby, contrary to a real energy system, energy cannot be stored for use in other design days.

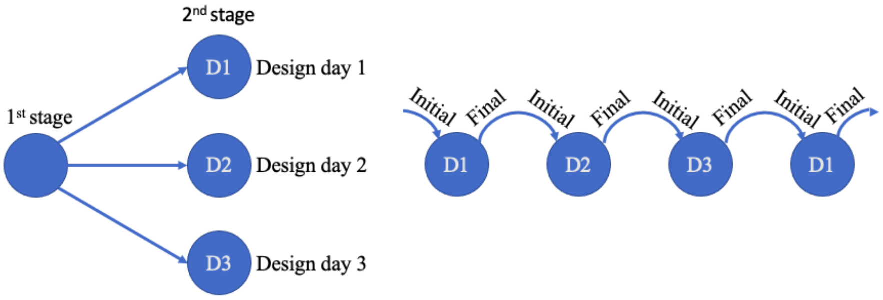


Figure 4.5: Day-sequence modelling by coupling design days through initial and final states.

4.2.4 Scenario-wise decomposable constraints

As explained in Section 3.4, using any version of Benders decomposition requires problems to have a special block structure that can be exploited. For the flexible design problem, this entails that only the coupling design decisions, i.e. the first-stage variables, can be present in constraints in several scenarios. On the other hand, second-stage variables, i.e. operational decisions, have to be restricted to existence in only one scenario.

All design days have to be possible to solve independently as subproblems when the design decisions are made. For instance, it is not possible to create constraints on expected values of second-stage variables as the constraints would involve variables from multiple scenarios. While GOSSIP includes other algorithms for solving problems, full utilization of GOSSIP calls for the need for compatibility with the decomposition approaches.

4.2.5 Discrete first-stage variables

GOSSIP provides a framework for studying the effects of non-linearities in the flexible RES design problem through the NGBD algorithm. For guaranteed convergence of the NGBD algorithm, the first-stage variables have to be discrete. A continuous design variable, y, can be discretized by:

1. Assuming there is a finite number of discrete values, N_D, that the first-stage variable can hold.
2. Constructing a set of discrete values, $\boldsymbol{z}^d$, with N_D entries containing the possible discrete values that the first-stage variable can take on.
3. Constructing a set of binary integer variables, $\boldsymbol{z}^b$, with N_D entries to represent the selection of value in the discrete set for the first-stage variable.
4. Specifying that only one of the binary variables in the set can have 1 as their value.
5. Letting the continuous variable be defined as the sum of products of the entries in the set for discrete values and their respective binary variables, as given in Equation 4.1.

$$y = \sum_{j=1}^{N_D} z_j^b \cdot z_j^d \tag{4.1}$$

While the decomposable block structure discussed in the previous section is the core concept of Benders decomposition, i.e. needed for MILPs, MICPs and MINLPS, the discrete set of values for first-stage variables are a requirement for the NGBD only. In other words, discrete first-stage variables are only a requirement for non-convex mixed-integer problems in GOSSIP. For consistency, discrete first-stage variables are also applied for the development of the MILP model for the flexible design problem.

4.3 Model formulation

In this section, the complete MILP model formulation for the optimal design under uncertainty problem of a flexible energy system hub with solar power, wind power, end user, grid and battery energy storage is presented. The model is built on the basis of the deterministic equivalent formulation of two-stage stochastic programs in Problem (SP) and the assumptions presented in the previous sections. The complete $C++$ model file used in GOSSIP can be found in Appendix B.

4.3.1 Model variables and uncertain parameters

Defining the uncertain parameters and decision variables of the system is key for the two-stage stochastic model formulation. This section defines and presents the relationship between these entities.

First-stage decision variables

The first-stage decisions involve choosing the optimal capacity of each technology. The design decisions for the stochastic program are the optimal values for area of solar panels installed, the number of wind turbines built and battery pack capacity, denoted by Z_{PV}, Z_{WT} and Z_{BAT} respectively in Table 4.1. Upper and lower bounds are denoted by UB and LB respectively.

Table 4.1: First-stage decision variables.

Variable	Unit	Description	UB	LB
Z_{PV}	m^2	Installed area of solar panels	Z_{PV}^{LB}	Z_{PV}^{UB}
Z_{WT}	-	Installed number of wind turbines	Z_{WT}^{LB}	Z_{WT}^{UB}
Z_{BAT}	MWh	Installed battery capacity	Z_{BAT}^{LB}	Z_{BAT}^{UB}

Following the requirement of discrete first-stage variables in the NGBD algorithm, each component is assumed to only take on one size out of a discrete set of sizes, as explained in Section 4.2.5.

$$Z_i = \sum_{j=1}^{N_{D,i}} z^b_{i,j} \cdot z^d_{i,j}, \forall i \in I \tag{4.2}$$

where $I = \{PV, WT, BAT\}$ is the set of possible technologies, $\boldsymbol{z^b}$ is the vector of binary variables, and $\boldsymbol{z^d}$ represents the discrete set of sizes. The set $\boldsymbol{N_D}$ contains the number of entries in the sets for each technology. The possible sizes in the discrete set are defined by Equation 4.3:

$$z^d_{i,j} = Z^{LB}_i + \frac{j-1}{N_{D,i}-1}(Z^{UB}_i - Z^{LB}_i), \forall i \in I, \forall j \in \{1, 2, \cdots, N_{D,i}\} \tag{4.3}$$

where Z^{UB}_i, Z^{LB}_i and $N_{D,i}$ are the upper bound, lower bound and the number of discrete values for technology i respectively.

Second-stage decision variables

After the uncertainty is revealed, a second set of decisions is made. The second-stage decision variables of a RES can be divided into two different classes of state and flow variables. State variables are necessary for describing the state of the dynamic components, while flow variables describe the interaction between components. In this model, the state variables are the available capacity in the battery, denoted by $\boldsymbol{E_{BAT}}$, while $\boldsymbol{f}$ denotes all flow variables that are the electric power flows between components. Table 4.2 presents the second-stage decision variable definitions.

Table 4.2: Second-stage decision variables.

Variable	Units	Description	LB	UB
$\boldsymbol{E_{BAT}}$	MWh	Available battery capacity	0	Z^{UB}_{BAT}
$\boldsymbol{f_{PV}}$	MW	Power flow from solar farm	0	f^{UB}_{PV}
$\boldsymbol{f_{WT}}$	MW	Power flow from wind farm	0	f^{UB}_{WT}
$\boldsymbol{f_{REN}}$	MW	Hub generated power flow to end user	0	$f^{UB}_{PV}+f^{UB}_{WT}$
$\boldsymbol{f_{DEM}}$	MW	Total power flow to end user	0	f^{UB}_{DEM}
$\boldsymbol{f^{ch}_{BAT}}$	MW	Power flow into battery	0	f^{UB}_{BAT}
$\boldsymbol{f^{dis}_{BAT}}$	MW	Power flow out of battery	0	f^{UB}_{BAT}
$\boldsymbol{f^{exp}_{GRID}}$	MW	Power flow exported to grid	0	$f^{UB}_{PV}+f^{UB}_{WT}$
$\boldsymbol{f^{imp}_{GRID}}$	MW	Power flow imported from grid	0	f^{UB}_{DEM}

Upper bounds, with a margin of 5%, on system flows are given by calculation of maximum output from the PV and wind farms, maximum flow for the battery, and maximum flow to the end user, as shown in Equations 4.4, 4.5, 4.6, and 4.7.

$$f^{UB}_{PV} = max(\boldsymbol{IR}_{PV}) \cdot Z^{UB}_{PV} \cdot \eta_{PV} \cdot 1.05 \tag{4.4}$$

$$f^{UB}_{WT} = Z^{UB}_{WT} \cdot q_d \cdot 1.05 \tag{4.5}$$

$$f^{UB}_{BAT} = Z^{UB}_{BAT} \cdot C^{max}_{rate} \cdot 1.05 \tag{4.6}$$

$$f^{UB}_{DEM} = max(\boldsymbol{L}_{DEM}) \cdot 1.05 \tag{4.7}$$

In Equation 4.4, $max(\boldsymbol{IR}_{PV})$ is the maximum possible radiation incident on the solar panel and η_{PV} is the conversion efficiency to electricity. Further, q_d in Equation 4.5, C^{max}_{rate} in Equation

4.6, and $max(\boldsymbol{L}_{DEM})$ in Equation 4.7 are the rated wind turbine power, maximum C-rate, and maximum power demand respectively. It is noted that $max(\boldsymbol{IR}_{PV})$, q_d, C^{max}_{rate}, and $max(\boldsymbol{L}_{DEM})$ all are given in megawatts.

Uncertain parameters

This model assumes four uncertain parameters and a finite number of scenarios, N_S, to be representative for the uncertainty. The set of scenarios, $S = \{h : h = \{1, \cdots, N_S\}\}$ are combinations of different realizations of the uncertain parameters, and all scenarios are assumed to have an associated probability defined by a vector $\boldsymbol{\omega}$ of length N_S. The uncertain parameters in the model are presented in Table 4.3.

Table 4.3: Uncertain parameters.

Parameter	Units	Description
$\boldsymbol{IR_{PV}}$	MW/m^2	Solar irradiance
$\boldsymbol{W_{WT}}$	m/s	Wind speed
$\boldsymbol{L_{DEM}}$	MW	Demand load
$\boldsymbol{OC_{GRID}}$	$\$/MWh$	Occurred cost when importing from grid

Program structure

Model topology including uncertain parameters and decision variables is summarized in Figure 4.6.

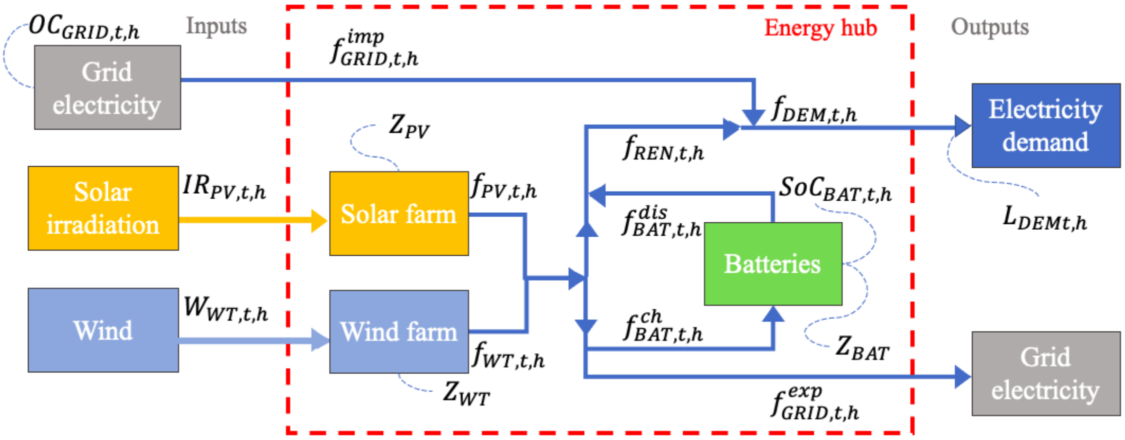


Figure 4.6: Model topology.

4.3.2 Objective function

The objective in the flexible design problem is to minimize the total expected costs of the system. For two-stage stochastic programs, the total cost contributions can be split into two groups:

- First-stage costs, CAP: Capital expenses and maintenance costs assumed to be governed solely by the design decisions.
- Second-stage costs, OP_h: Related to purchasing and selling electricity for a scenario h. These costs are entirely dependent on the second-stage variables and are therefore not constant for all scenarios. Importing electricity from the grid will lead to a cost, while exporting will generate revenue. Note that revenue will be regarded as a cost with a negative sign in the model.

The total cost of a scenario, h, is the sum of CAP and OP_h, while the total expected cost of the system is given by Equation 4.8. The expected value is denoted by the symbol $\mathbb{E}$.

$$\textit{Total Annual Expected Cost} = CAP + \mathbb{E}_{h\in S}(OP_h) \tag{4.8}$$

First-stage costs

The first-stage costs include the up-front investment costs of building the system and the corresponding yearly maintenance cost, M^{yearly}. In the objective function, the costs have to be accounted for with the same time horizon. The investment cost of a technology, i, is annualized by a capital recovery factor, CRF_i, given by Equation 4.9.

$$CRF_i = \frac{r(r+1)^{l_i}}{(r+1)^{l_i} - 1} \tag{4.9}$$

where r is the interest rate and l_i is the lifetime of the technology. The yearly investment cost is then given by:

$$Invest^{yearly} = \sum_{i\in I} CRF_i \cdot C_i \cdot Z_i \tag{4.10}$$

where C_i is the linear cost per unit size of each technology, $\$/m^2$ for PV, $\$/turbine$ for wind and $\$/MWh$ for batteries, The yearly maintenance cost for technology i is given as a fraction, ξ_i, of the yearly investment cost in Equation 4.11.

$$M^{yearly} = \sum_{i\in I} Invest_i^{yearly} \cdot \xi_i \tag{4.11}$$

The total annual first-stage costs can then be calculated by Equation 4.12.

$$CAP = Invest^{yearly} + M^{yearly} \tag{4.12}$$

Second-stage costs

Expenses and revenue associated with operating decisions represent the second-stage costs in the objective function. These operating decisions involve the the amount of energy imported grid, which are penalized with an occurred cost per megawatt-hour, OC_{GRID}. Additionally, energy exported to the grid and used for covering the demand load will create revenue. The operating cost for a scenario h is given in Equation 4.13.

$$OP_h = \frac{T_{year}}{T}\Delta t \left(\sum_{t=1}^{T} (f_{GRID,t,h}^{imp} \cdot OC_{GRID,t,h} - f_{GRID,t,h}^{exp} \cdot FiT_{extra} - (f_{DEM,t,h} - f_{GRID,t,h}^{imp}) \cdot FiT) \right) \tag{4.13}$$

where Δt is the timestep in hours, T is the number of timesteps in a scenario, T_{year} is the the number of timesteps in a full year, and FiT is the constant feed-in-tariff of renewable power for covering demand.

Furthermore, excess power which is sold to the grid is assumed to create a revenue of a value of FiT_{extra}. This represents a decreased value with increasing shares of renewables in the system's energy mix. Additionally, the imported power from the grid is assumed to not create revenue when used for covering demand. In Equation 4.14, the expected second-stage costs are calculated from the probability of each scenario and the associated operating costs.

$$\mathbb{E}_{h\in S}(OP_h) = \sum_{h\in S} \omega_h \cdot OP_h \tag{4.14}$$

4.3.3 First-stage constraints

The use of binary variables for creating the discrete sets for the first-stage decision variables requires sets of constraints that make sure only one of the binary variables in each set is 1 ("True"), as given by Equation 4.15.

$$\sum_{j=1}^{N_{D,i}} z^b_{i,j} - 1 = 0, \forall i \in I \tag{4.15}$$

For the sizing of each technology, this model assumes a constraint on the total initial investment cost of the system given in Equation 4.16. The purpose of the constraint is to include the possibility of a user-specified budget. This is important to include renewable energy projects that has limited funds through loans and investors.

$$\sum_{i \in I} C_i \cdot Z_i \leq Budget \tag{4.16}$$

In the problem of evaluating the expected value solution in a stochastic environment, referred to as the EEVP in this work, the first-stage variables will be set equal to the solution of the EVP by Equation 4.17. We note that the EEVP is the optimization problem of evaluating the nominal design in the stochastic environment, while EEV is the minimum value of the objective function to the EEVP.

$$z^b_{i,j} - 1 = 0, \forall (i,j) \in \{(PV, n_{PV}), (WT, n_{WT}), (BAT, n_{BAT})\} \tag{4.17}$$

where n_{PV}, n_{WT} and n_{BAT} is the discrete variable numbers representing the EVP design decisions.

4.3.4 Second-stage constraints

The second-stage constraints include decision variables to be undertaken after uncertainty has been revealed. In other words, these constraints specify the rules for behavior of system operation.

Energy balances for the energy hub

The energy balances for the energy hub are key to governing the behaviour of the system and act as links between the inputs and outputs of the system. The energy balances make sure the net power flow in or out of the junctions in Figure 4.6 are zero for each time step in all scenarios. The first energy balance is given in Equation 4.18, and requires that the sources of electricity, grid and renewable power, have to be able to cover the demand load at all time steps.

$$f_{DEM,t,h} = f_{REN,t,h} + f^{imp}_{GRID,t,h} \forall t \in \{1, 2, \cdots, T\}, \forall h \in S \tag{4.18}$$

The second energy balance, given in Equation 4.19, is necessary to determine when, where and the amount of renewable power in the system flows. The renewable power can be used to cover the end user load, exported to the grid, used for charging the battery, or extra renewable power can be extracted from the battery if available capacity.

$$f_{PV,t,h} + f_{WT,t,h} + f^{dis}_{BAT,t,h} = f_{REN,t,h} + f^{exp}_{GRID,t,h} + f^{ch}_{BAT,t,h}, \forall t \in \{1, 2, \cdots, T\}, \forall h \in S \tag{4.19}$$

Renewable power generation

For renewable power generation, a relation has to be established between the uncertain parameters, the design decisions on the technology and the renewable power flows generated from the solar and wind farms.

In Equation 4.20, the generated power flow from the PV farm is given by the area of solar panels installed and the irradiance from the sun. The relation is explicitly defined and thereby only one legal operational decision can be made for this variable per timestep in each scenario.

$$f_{PV,t,h} = \eta_{PV} \cdot IR_{PV,t,h} \cdot Z_{PV}, \forall t \in \{1,2,\cdots,T\}, \forall h \in S \tag{4.20}$$

where η_{PV} is the efficiency of the solar panels.

Estimation of a wind turbine power output, $\boldsymbol{q}$, is obtained through a wind velocity determined approximation of the power curve of pitch controlled wind turbines described in [31]. The power response to wind speed is given in four different operating regimes by Equation 4.21.

$$q_{t,h} = \begin{cases} 0, & \text{if } W_{t,h} < W_{min} \\ q_d \cdot \frac{W_{t,h}^3 - W_{min}^3}{W_d^3 - W_{min}^3}, & \text{if } W_{min} \leq W_{t,h} < W_d \\ q_d, & \text{if } W_d \leq W_{t,h} \leq W_{max} \\ 0, & \text{if } W_{t,h} > W_{max} \end{cases} \forall t \in \{1,2,\cdots,T\}, \forall h \in S \tag{4.21}$$

where q_d is the rated power, W_d is the rated wind speed, W_{min} is the cut-in wind speed, and W_{max} is the cut-off wind speed for the wind turbine. The total wind power generated, i.e. the total power flow from the wind farm, is then calculated by Equation 4.22.

$$f_{WT,t,h} = q_{t,h} \cdot Z_{WT}, \forall t \in \{1,2,\cdots,T\}, \forall h \in S \tag{4.22}$$

where $q_{t,h}$ is the power output for a single wind turbine at time step t for scenario h and Z_{WT} is the total number of wind turbines.

End user

The model assumes that for all scenarios, the entire uncertain demand load has to be covered by power flows from the grid, PV farm or battery. Additionally, the demand is independent of system states and flows in the system. This entails that the power flow to the end user has to be the same value as the uncertain variable, as given by Equation 4.23[1].

$$f_{DEM,t,h} = L_{t,h}, \forall t \in \{1,2,\cdots,T\}, \forall h \in S \tag{4.23}$$

Battery energy storage

The battery provides operational decisions with degrees of freedom to the model. The available energy stored in the battery is kept track of by Equations 4.24 and 4.25. The energy balance for the battery makes sure that the energy flow into and out of the battery equals the change in energy stored in the battery:

$$\begin{aligned} E_{BAT,t,h} = E_{BAT,t-1,h} \cdot \eta_{SD} + \Delta t \cdot (f_{BAT,t,h}^{ch} \cdot \eta_{BAT}^{ch} - \frac{f_{BAT,t,h}^{dis}}{\eta_{BAT}^{dis}}), \\ \forall h \in S, \forall t \in [2,3,\cdots,T] \end{aligned} \tag{4.24}$$

$$E_{BAT,t=1,h} = (SoC_{BAT}^{initial} - SoC_{BAT}^{min}) \cdot Z_{BAT} + \Delta t \cdot (f_{BAT,t,h}^{ch} \cdot \eta_{BAT}^{ch} - \frac{f_{BAT,t,h}^{dis}}{\eta_{BAT}^{dis}}), \forall h \in S \tag{4.25}$$

where η_{BAT}^{ch} and η_{BAT}^{dis} are the efficiencies for charging and discharging the battery respectively. η_{SD} represents the storage efficiency and the losses that arise from battery self-discharge over the course of one timestep. Equation 4.24 provides the relation for previous states in the system, whereas Equation 4.25 provides the relation for the first time step using an assumed initial state of charge, denoted by $SoC_{BAT}^{initial}$.

To prevent deep discharges, the energy available in the battery is restricted by a minimum state of charge of the battery, SoC_{BAT}^{min}, and high states of charge are prevented by a maximum state

[1] These equations are not implemented as constraints in the model file. Rather, the uncertain parameter $L_{t,h}$ has been used directly in equations where $f_{DEM,t,h}$ is found.

of charge, SoC_{BAT}^{max}. Equation 4.26 defines the available capacity of the battery. Additionally, the available energy in the battery cannot be negative, defined by Equation 4.27.

$$E_{BAT,t,h} \leq (SoC_{BAT}^{max} - SoC_{BAT}^{min}) \cdot Z_{BAT}, \forall h \in S, \forall t \in [1, 2, \cdots, T] \tag{4.26}$$

$$E_{BAT,t,h} \geq 0, \forall h \in S, \forall t \in [1, 2, \cdots, T] \tag{4.27}$$

As discussed previously in Section 2.3, how fast the battery can charge and discharge will affect the battery performance and lifetime. The maximum average power flow out of and in to the battery during one timestep is controlled through the user-defined maximum C-rate, C_{rate}^{max}, as shown in Equations 4.28 and 4.29.

$$f_{BAT,t,h}^{ch} \leq \Delta t \cdot C_{rate}^{max} \cdot Z_{BAT}, \forall h \in S, \forall t \in [1, 2, \cdots, T] \tag{4.28}$$

$$f_{BAT,t,h}^{dis} \leq \Delta t \cdot C_{rate}^{max} \cdot Z_{BAT}, \forall h \in S, \forall t \in [1, 2, \cdots, T] \tag{4.29}$$

Additionally, periodicity is maintained by Equation 4.30 through the assumption that the final available energy stored in the battery has to be equal to the initial state:

$$E_{BAT,t=T,h} = (SoC_{BAT,h}^{initial} - SoC_{BAT}^{min}) \cdot Z_{BAT}, \forall h \in S \tag{4.30}$$

4.4 Input data

Evaluation of the model in the previous section is performed by a simple case study consisting of certain inputs. In this section, the constant model parameters and realizations of uncertain model parameters are presented.

4.4.1 Constant input data

The model requires certain inputs that are assumed to be independent of both scenarios and design-decisions, i.e. they are fixed. Table 4.4 presents the fixed design and cost parameters for the different technologies, while Table 4.5 gives the assumed fixed operational parameters related to conversion by the technologies. Other constant parameters are presented in Table 4.6.

Table 4.4: Constant design and cost parameters for technologies.

		PV		**WT**		**BAT**	
Param	Description	Value	Unit	Value	Unit	Value	Unit
Z^{UB}	Design upper bound	72,000	m^2	18	-	13.5	MWh
Z^{LB}	Design lower bound	0	m^2	0	-	0	MWh
N_D	Discrete values	10	-	10	-	10	-
l	Lifetime [27][33][39]	30	years	20	years	15	years
r	Discount rate	6	%	6	%	6	%
C	Cost factor [47]	130	$\$/m^2$	1.2	$M\$/no.$	0.18	$M\$/MWh$
ξ	Maintenance factor [45]	5	%	5	%	2	%

Table 4.5: Constant operational parameters for technologies.

PV			**WT** [48]			**BAT**		
Param	Value	Unit	Param	Value	Unit	Param	Value	Unit
η [25]	20	%	q_d	1	MW	η_{ch}[47]	95	%
			W_d	14	m/s	η_{dis} [47]	95	%
			W_{min}	4	m/s	η_{SD} [49]	99.8	%
			W_{max}	25	m/s	C_{rate}^{max}	0.25	-
						SoC_{BAT}^{max}	90	%
						SoC_{BAT}^{min}	11	%
						$SoC_{BAT}^{initial}$	50	%

Table 4.6: Other constant parameters.

Parameter	Description	Value	Unit
Δt	Length of timestep	1	h
T	Timesteps in each scenario	24	$-$
T_{year}	Timesteps in a year	8760	$-$
FiT	Assumed FiT for covering demand	70	$\$/MWh$
FiT_{extra}	Assumed FiT for exports beyond demand	35	$\$/MWh$
$Budget$	Max investment cost	20	$M\$$

4.4.2 Scenarios

In this case study, there is assumed to be four seasons to describe seasonal variation: Winter, spring, summer and fall. Additionally, there is assumed to be three scenarios each season to model the fluctuations in power generation to a certain degree. The three scenarios per season can be described as:

- Average case: All uncertain parameters take on their assumed average values. The assumed probabilities for the average cases are assumed to be 50%.
- High radiation, low wind speed: The solar radiation incident on the panel is scaled by a factor of 1.35 and the wind speed is scaled by a factor of 0.65. The assumed probability for each season is 25%.
- Low radiation, high wind speed: The solar radiation incident on the panel is scaled by a factor of 0.65 and the wind speed is scaled by a factor of 1.35. The assumed probability for each season is 25%.

Uncertain data and all 12 scenarios can be found in Appendix C.

Irradiance

As discussed in Section 2.1, the solar irradiance incident on an inclined panel depends on the position of the sun and atmospheric effects. The position of the sun will depend on the time of day and the season, while the atmospheric effects depend on the position of the sun, the ground albedo, atmospheric diffusion, and atmospheric absorption and reflection. The average seasonal values and the yearly average values for EVP, illustrated in Figure 4.7, for a location at 52°north, 13°east and an altitude at 35 meters are generated by [21]. The solar panels are assumed to have an inclination of 30°and an albedo of 0.2 is assumed for the ground.

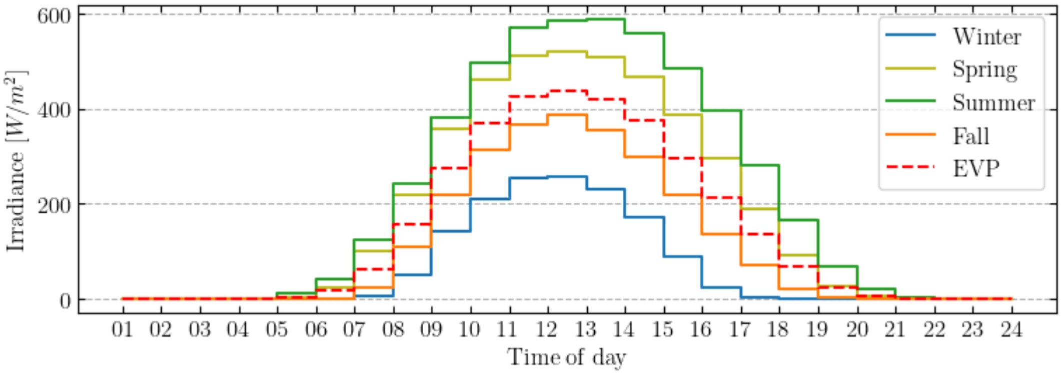


Figure 4.7: Daily radiation patterns.

Wind speeds

Seasonal average wind speeds at wind turbine height are estimated from the average speed near the ground by the boundary layer equation, Equation 2.6, in Section 2.2. The average values of wind are gathered from [50], and the typical daily fluctuations are based on [51]. The resulting average seasonal wind speed variations are illustrated in Figure 4.8 together with the yearly averages for the EVP.

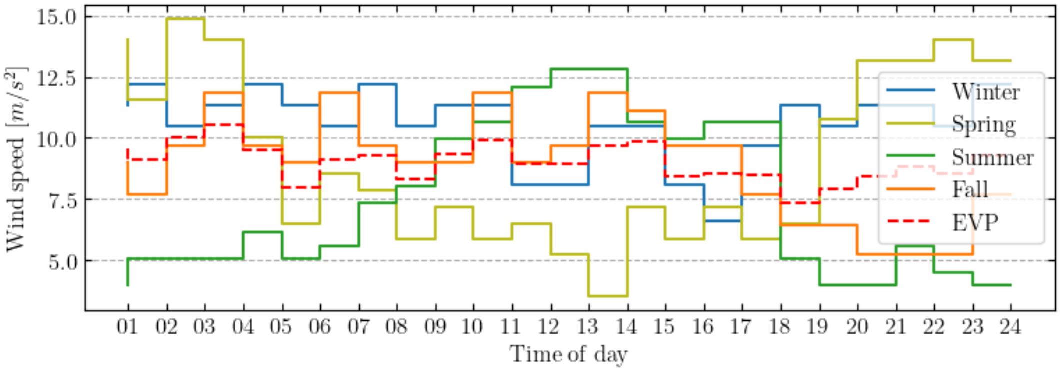


Figure 4.8: Daily wind patterns.

Grid electricity price

The penalty cost of not being able to cover for demand with self-generated renewable energy is assumed to follow typical daily price variations in electricity. The average variations for each season and data for EVP are shown in Figure 4.9. The hourly fluctuating patterns are based on [52], and the differences in seasonal variations are based on [53]. The average value of the penalty when importing from the grid is assumed to be 79 $\$/MWh$

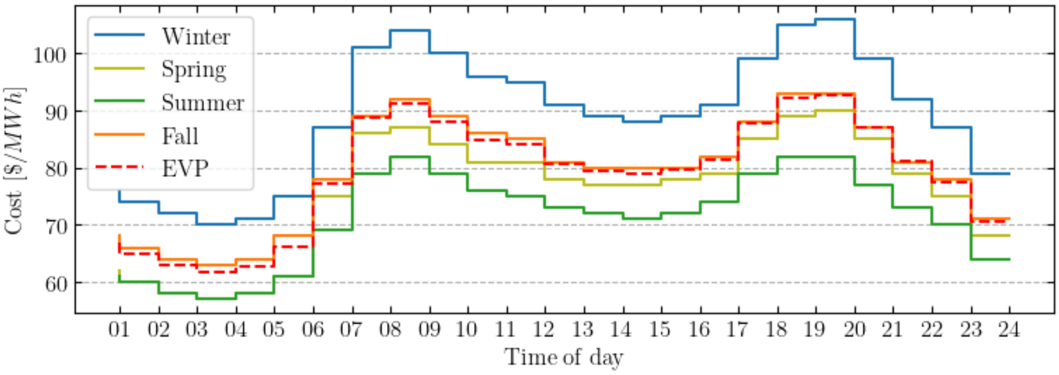


Figure 4.9: Hourly variation of penalty when importing from grid.

Demand load

Hourly demand load data for a full year is gathered from the base year for the model used to estimate future load patterns in [54]. The seasonal average values are used as scenarios in this case study and the yearly average is used for the EVP. The load variations are presented in Figure 4.10.

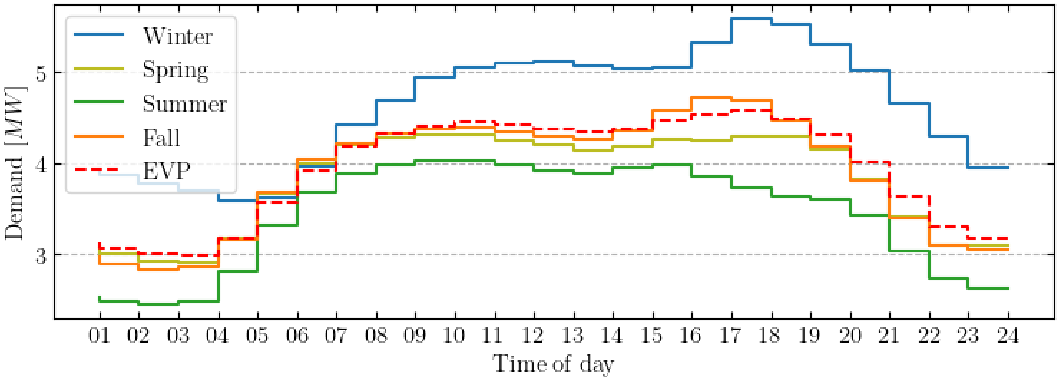


Figure 4.10: Hourly demand load variations.

5 | Results and interpretations

5.1 Optimization strategies

Flexible design and expected values

The optimal design decisions and resulting costs, expected operation, and expected yearly profit are shown in Table 5.1. The three problems in the results can be described as:

- The expected value problem (EVP) is set in a deterministic environment, i.e. one typical day with expected values for the uncertain parameters. The design decisions found in the EVP is referred to as the nominal design.
- For the EEVP, i.e. the problem of finding the EEV, the design decisions are set to the nominal design while subject to the stochastic environment.
- The stochastic problem (SP) is set in the stochastic environment and a flexible design strategy is utilized. The design decisions obtained by using stochastic programming are referred to as the flexible design.

Expected operational decisions are the expected values of system flows, $\mathbb{E}_f$, and battery state of charge, $\mathbb{E}_{SoC}$. The daily expected value for a system flow, f, is calculated by Equation 5.1 and the expected value for the state of charge, SoC, is calculated from the design decision, Z_{BAT}, and energy levels in the battery, $\boldsymbol{E_{BAT}}$, by Equation 5.2.

$$\mathbb{E}_f = \sum_{h=1}^{N_S} \left(\omega_h \cdot \sum_{t=1}^{T} f_{t,h} \right) \tag{5.1}$$

$$\mathbb{E}_{SoC} = \sum_{h=1}^{N_S} \left(\frac{\omega_h}{T} \cdot \sum_{t=1}^{T} \frac{E_{BAT,t,h}}{Z_{BAT}} \right) \tag{5.2}$$

The nominal design obtained in the EVP was found to be 16,000 m^2 of solar panels, 14 wind turbines and a 6 MWh battery pack. The nominal design seems to heavily favor wind over solar energy as the primary energy source, contributing to almost 90% of the total energy production. Further, the nominal design in the deterministic environment is expected to use the battery for more than one full cycle on average per day. This way, imported and exported power flows from the grid are kept to a minimum while keeping budget specifications. The program expects an annualized profit of 323 k$ for the project.

Using a nominal design in stochastic environments, however, entails a risk of sub-optimal decisions. In this case, the nominal design is found to be unprofitable, specifically 380 k$ less than what was estimated in the EVP. The expected utilization of the battery is also notably lower, with a decrease in both expected discharge flow and state of charge for the EEVP. Meanwhile, grid electricity import and export increase by multiple folds. Thereby, an increasing amount of energy has to be acquired at a high cost and more of the generated power has to be sold at an unfavourable price. This is despite the fact that the expected energy generation for the nominal

Table 5.1: Nominal and flexible design results.

Problem	**EVP**	**EEVP**	**SP**
Number of scenarios	1	12	12
Number of timesteps	24	24	24
Design	Nominal	Nominal	Flexible
Design Decisions			
Area of solar PV (m^2)	16000	16000	48000
Number of wind turbines (-)	14	14	10
Battery size (MWh)	6.0	6.0	9.0
Costs			
Construction (M$)	19.96	19.96	19.86
Annualized investment (M$)	1.73	1.73	1.67
Yearly maintenance ($)	83014	83014	78313
Expected demand (MWh/day)	94.14	94.14	94.14
Expected energy generation (MWh/day)			
Solar PV	10.51	10.51	31.54
Wind power	85.39	114.21	81.58
Expected energy flows (MWh/day)			
Grid (import)	6.56	25.1	18.55
Grid (export)	7.54	55.07	36.54
From battery	6.3	4.65	7.75
To battery	7.09	5.26	8.74
Expected state of charge (%)	36.31	23.23	44.77
Expected annual loss ($)	-323,200	57,518	-139,540
Value of stochastic solution ($/year)	-	-	197,058

design are greater in the stochastic environment than in the deterministic environment.

Using the flexible design strategy results in an annualized profit of around $140K, while a nominal design ends up with losses. Four wind turbines are cut from the design, while the capacity of solar PV increases threefold and the energy storage capacity increases by 50%. Notably, this is found to decrease the amount of expected energy generated by about 10%. Further, a higher utilization of the battery and generated power is found: For a 50% increase in battery capacity, expected discharge flow from the battery increased by 67%. Thus, the amount of power exported and imported power decrease significantly, yielding a value of the stochastic solution of 198 k$.

Levelized cost of energy

The results in Table 5.1 show that each wind turbine could produce more energy in the stochastic environment than in the deterministic environment used in the EVP. In the stochastic environment, the amount of wind power produced increases by 34% despite construction and maintenance costs remaining constant. On the other hand, solar power generation in the nominal design is not affected by the environment.

The levelized cost of energy (LCOE) is the expected net present cost of energy over the course of the lifetime of the energy generating plant and is dependent on the environment the system is placed in. The LCOEs are calculated by the ratio between the sum of cost over the lifetime to the sum of electricity produced over the lifetime for a given energy source, as shown in Equation 5.3.

$$LCOE_i = \frac{\sum_{y=1}^{l_i} \frac{Inv_{i,y} + M_{i,y}}{(1+r)^y}}{\sum_{y=1}^{l_i} \frac{E_{f_i} \cdot 365}{(1+r)^y}}, \forall i \in PV, WT \tag{5.3}$$

where $Inv_{i,y}$ is the investment cost, $M_{i,y}$ is the maintenance cost and $(E_{f_i} \cdot 365)$ is the expected energy produced in the year y and for technology i. The lifetime of technology i is l_i and r is the discount rate.

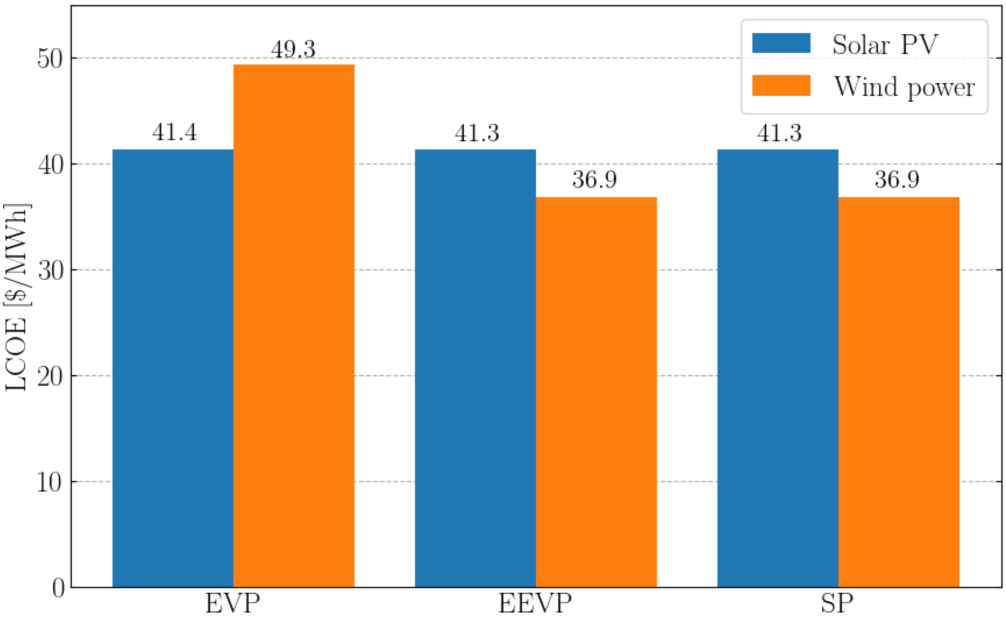


Figure 5.1: Levelized cost of renewable energy.

The LCOEs of wind and solar power in a deterministic and stochastic environment are given in Figure 5.1. Nominal designs are used in both the EVP and EEVP, but they are placed in different environments. It is illustrated that the cost of wind decreases by 25% in the uncertain environment, with a corresponding increase of 34% in wind energy output. This is to be expected as calculation of levelized cost of energy uses values of expected power produced.

Variations in LCOE of wind in the environments originate from the wind power model and how the environment in the EVP is calculated. The wind speeds naturally have larger variations in the stochastic environment than what is used in the EVP. The wind power model, as previously discussed in Section 2.2, introduces a non-linear relationship between power and wind. The expected power output will therefore depend on both wind speed variance and mean, and this shows the importance of modelling uncertainty for power production from wind. Additionally, the wind turbine performance in a certain environment depends on important characteristics such as cut-in and cut-out wind speed, and rated nominal power and wind speed. Different types of wind turbines could therefore beneficially be taken into account when designing the system.

On the other hand, a linear solar PV power model is assumed. The LCOE for solar PV is therefore expected to be identical in the two environments. The small difference in the LCOE is expected to have arisen from rounding errors in the creation of scenarios. Furthermore, the LCOEs for PV and wind with nominal and flexible designs in the stochastic environment are shown to be identical. This is to be expected as the flexible design improves the ability to cover a certain demand, while the actual costs of the systems remain the same with the cost model used. One might expect to see LCOEs that are dependent on designs if other cost models were employed, such as economies of scale.

System value of renewable energy

The flexibility of the renewable energy system can be measured by its ability to cover the required load without importing power from external sources. The estimations for nominal design's expected imported and exported power in the EVP were off by 280% and 630% respectively, when compared to the EEVP. Opting for the flexible design strategy, both the expected amount of power exported at a unfavourable price and the imported power were significantly reduced. Thus, the RES is more likely to ensure the demand load with renewable energy supply with the flexible design than with the nominal design.

Security of energy supply (SOES) is a term used about the availability of energy and the energy supply's ability to cover load with an affordable cost. In this case, the security of energy supply will indicate the amount of energy that is supplied to meet the demand by using energy that is generated in the energy hub, i.e. renewable power, and is calculated by Equation 5.4. The remaining percentages will point to the expected portion of the demand load that is covered by imported electricity.

$$SOES = \sum_{h=1}^{N_S} \left(\frac{\omega_h}{T} \cdot \sum_{t=1}^{T} \frac{f_{REN,t,h}}{f_{DEM,t,h}} \right) \tag{5.4}$$

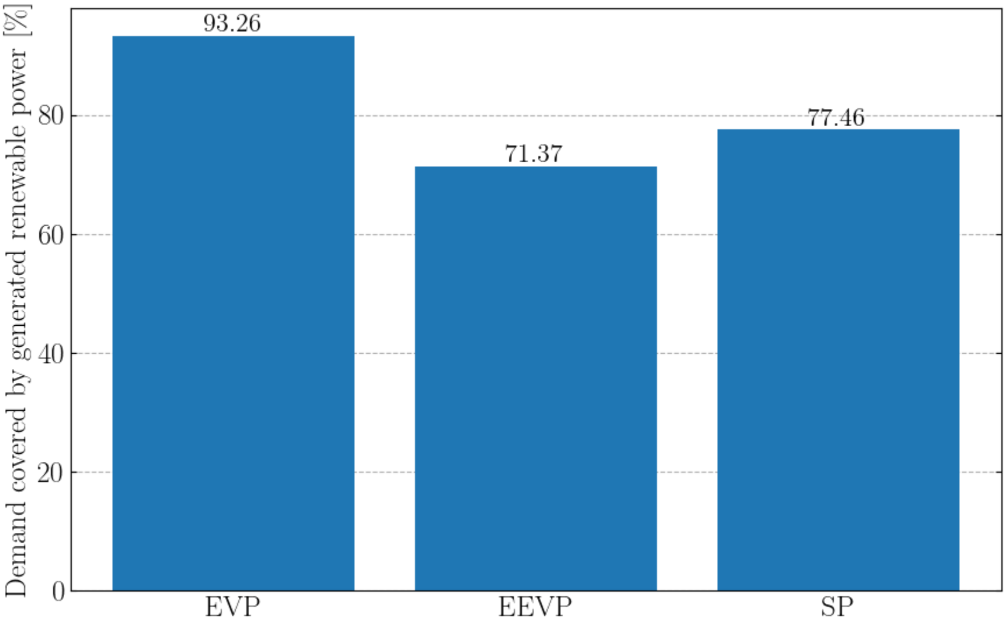


Figure 5.2: Security of renewable energy supply.

Figure 5.2 shows that the expected amount of energy sold at the preferable price decreases with over 20 percentage points for the EEVP. It may be logical to assume that the value of wind to the system should increase when the LCOE of wind was reduced, but as the nominal and flexible design in this work show, this is not necessarily the case. It is seen that the EVP risks erroneous evaluation of production patterns of the energy sources. The amount of wind energy produced in the EEVP seems to produce too much energy at certain unfavourable hours, increasing the amounts of exported power sold at an unfavourable price.

On the other hand, the flexible design identifies larger investments in solar and storage capacity as means to increase the security of energy supply and reducing the expected amount of imported energy by 26%. The total amount of generated power is reduced compared to the nominal design, seemingly counter-intuitive as increased capacities have been posed as a way to improve system flexibility. However, the program in this case found the two other measures for increased flexibility to be more valuable: Energy storage to offset fluctuations; and other production patterns more suitable for the demand curve.

Thus, it is illustrated how the cost of energy is not the only factor determining the system value of energy. Possible synergies between the respective environments of the technologies, i.e. how the technologies can work together to cover the demand, are crucial to investigate for optimal design of renewable energy systems. While LCOEs were independent of design strategies, both environment and design strategy will affect the security of energy supply. A flexible design strategy is therefore crucial for maximising the system value.

System operation

The battery are shown to be able to mitigate the power fluctuations of the renewable energy system. The battery behaviour during operation for each scenario is illustrated by two example scenarios:

- One scenario with high wind speeds and power surplus is presented in Figure 5.3.
- One scenario with a general power deficit and a solar PV dominated production curve in Figure 5.4.

Figure 5.3 shows how the system operates on a particularly windy day, with rated power for the wind turbines reached for larger parts of the day. Consequently, there is a large amount of surplus energy as demand for the most part remains below 50% of the power output. In other words, full utilization of wind leads to high amounts of exported power. This is expected to be one of the reasons to why the amount of wind turbines was reduced in the flexible design.

The bottom two plots of Figure 5.3 show how the battery operates for hourly energy storage in this type of scenario. A low utilization at well under 50% is noted, mostly due to the large amounts of surplus power. While there is only one timestep with a power deficit, the C-rate will prevent the battery from balancing out the difference. Thus, requiring importation of power even for a scenario with a large amount of power surplus. Although not visible in the bottom plot, there is a gradual slope for the state of charge when the battery is idle due to the self-discharge losses of the battery. Therefore, the system charges the battery to a required SoC of 50% at the end of the day, contrary to using the power surplus earlier in the day for charging power. This way, the average SoC, and thereby the self-discharge losses, are minimized for the scenario.

A scenario with low wind speeds is illustrated in Figure 5.4, which has a power production curve that resembles a PV dominated system. Low-speed winds around cut-in velocity generate low amounts of power throughout the night, but the wind speed drops below cut-in velocity in the evening. Consequently, large amounts of power have to be imported at times of day that are not peak solar hours. In fact, the entire demand load is covered by grid electricity before 6AM and after 8PM. The value added through increased energy storage capacity is clear in this scenario. The battery cycles at least one full time, offsets power surplus by the solar farm around noon and mitigates power deficits at other times of the day. In fact, the battery is able to store all surplus power generated by the system and utilize it later.

For operation of a RES, it is logical to think that the battery follows two main rules of operation:

1. Store energy when there is a system power surplus and available storage capacity.
2. Discharge energy when there is a system power deficit and available energy in the battery.

However, it is seen that this is not necessarily the case: The RES charges the battery with surplus power around noon, but is not discharged at the first timestep (3PM) with a power deficit.

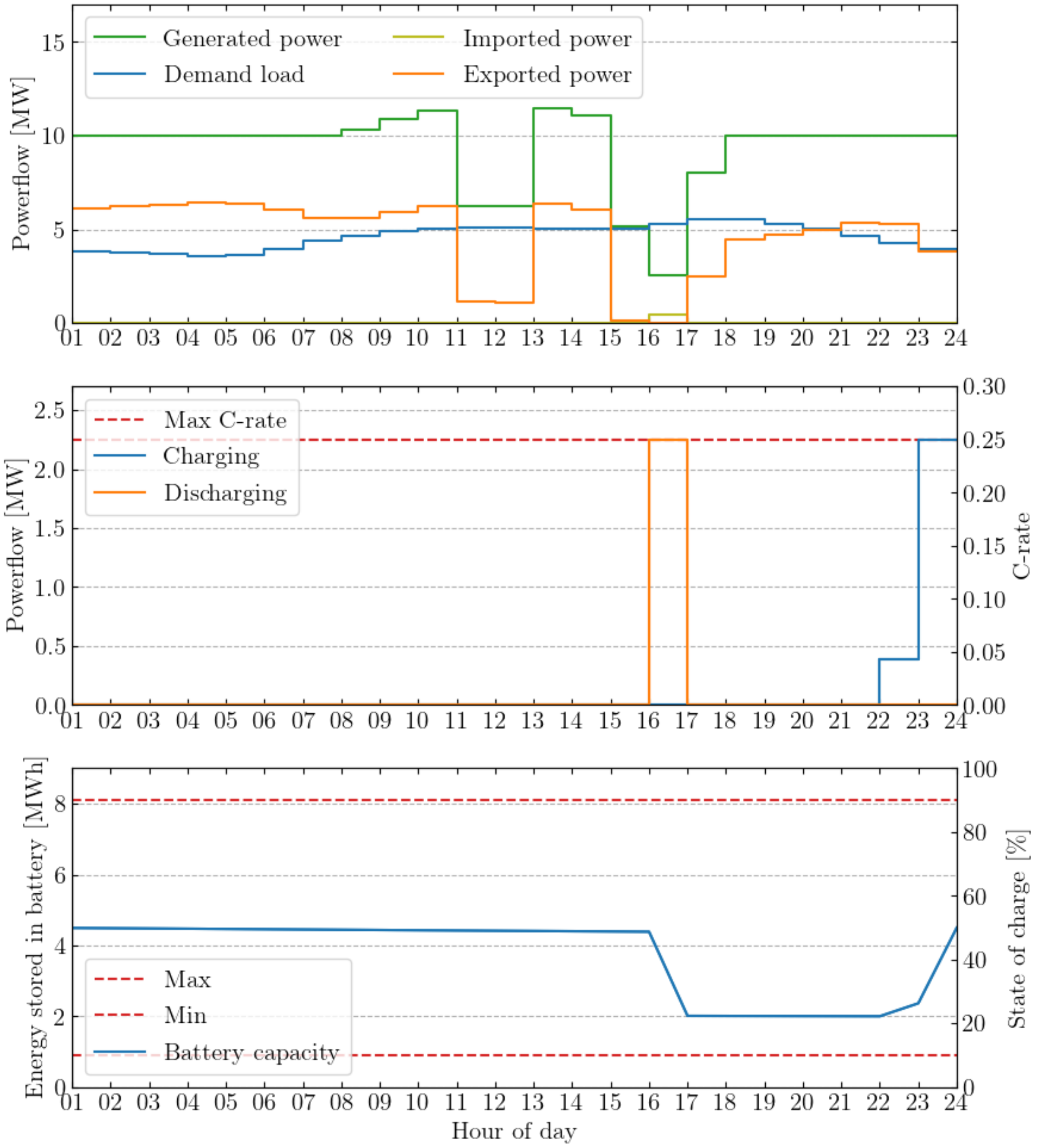


Figure 5.3: RES operation for an example scenario with high wind speeds.

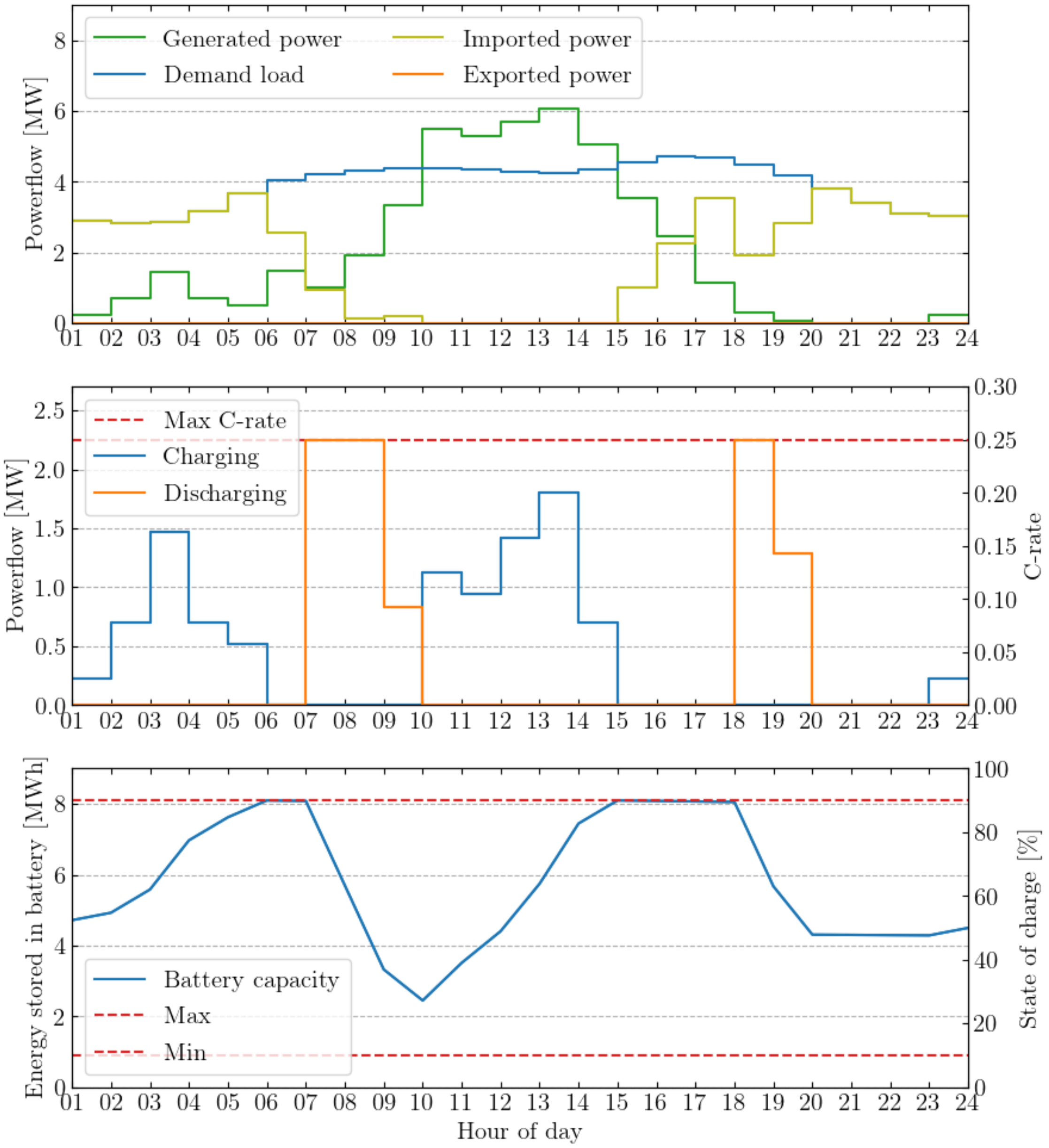


Figure 5.4: RES operation for an example scenario with low wind speeds.

Rather, the system has found it to be optimal to wait a few hours before discharging. While the battery increases its self-discharge losses during these timesteps, the fluctuating cost of importing from grid makes it beneficial to discharge the battery when there is a power deficit and a peak in cost of electricity import.

As previously shown by the grid electricity price patterns in Figure 4.9, two peaks are present during the day: One in the morning and one in the evening. Logically, it is preferable to import from the grid at more advantageous prices, enabled by the increased flexibility through battery storage. Hence, the battery is charged at the beginning of the day. In Figure 5.4 even though there is a substantial power deficit. The system imports the entire demand load, and all the wind power generated is used to charge the battery. This way, the amount of power imported at peak grid penalty cost is reduced, increasing the value of the battery.

Effect of battery and budget

The results in Table 5.1 show that the total production capacity was reduced to the benefit of two other measures for increased flexibility: battery energy storage and other production patterns. However, the budget seemed to be an active constraint in both nominal and flexible design.

To show the effect of increasing production capacity and the use of energy storage, the program is tested for two cases without a budget: One case with and one without a battery. The budget is removed by not including the constraint in Equation 4.16, while the battery is excluded in the program by removing the constraints in Equations 4.24-4.30. Additionally, the set of technologies will be reduced to $I = \{PV, WT\}$ in the case of no battery. The results for the two cases are shown in Table 5.2. It is noted that the upper bounds and number of discrete values defined in Table 4.4.1 have been increased to facilitate for larger designs.

The case of no budget or battery illustrates how a stochastic programming approach is used to achieve a flexible design by increasing expected production capacity. An increased production capacity leads to increased construction costs, but reduces the amount of imported electricity, giving a value of the stochastic solution of 273 k\$. Both area for solar PV and number of wind turbines increase in the flexible design, causing over a 50% reduction in expected imported power compared to the nominal design. However, for wind energy alone, it is seen that the expected amount of power produced is over 4 times as much as expected demand.

Energy storage is another measure for obtaining a flexible design. A battery in the nominal design leads to a significant decrease in expected amount of imported power in both the EVP and the EEVP compared to the nominal design without a battery. In the flexible design, there is an increase in the number of wind turbines instead of battery capacity, compared to the nominal design. This suggests that an increased production capacity is a more valuable measure for flexibility without a constraint on construction costs.

However, the battery is still able to provide a more flexible design. By comparison of the two cases, the battery can be seen to lower construction costs and production capacity. At the same time, imports from the grid are reduced and the expected yearly value of the project increases. Thus, energy storage reduced the need for flexibility through production capacity.

In this work, only two sources of energy were available. Wind and solar power are assumed to have provided some level of synergy for covering demand loads. For instance, the area installed for solar PV were reduced in the flexible design without a budget. However, it is expected that optimizing the combination of energy sources will be more valuable in larger problems with more different sources of renewable energy.

Table 5.2: Results from cases without a budget.

	Without battery			**With battery**		
Problem	**EVP**	**EEVP**	**SP**	**EVP**	**EEVP**	**SP**
Number of scenarios	1	12	12	1	12	12
Number of timesteps	24	24	24	24	24	24
Design	Nominal	Nominal	Flexible	Nominal	Nominal	Flexible
Design Decisions						
Area of solar PV (m^2)	32,000	32,000	56,000	48,000	48,000	40,000
Number of wind turbines (-)	18	18	48	16	16	46
Battery size (MWh)	-	-	-	4	4	4
Costs						
Construction (M$)	25.76	25.76	64.88	26.25	26.25	61.21
Annualized investment (M$)	2.19	2.19	5.55	2.21	2.21	5.27
Yearly maintenance ($)	109,270	109,270	277,535	108,031	108,031	261,186
Expected energy generation (MWh/day)						
Solar PV	21.03	21.03	36.8	31.54	31.54	26.28
Wind power	109.79	146.84	391.57	97.59	130.52	375.25
Expected demand (MWh/day)	94.14	94.14	94.14	94.14	94.14	94.14
Expected energy flows (MWh/day)						
Grid (import)	3.99	20.46	9.24	1.16	16.4	7.72
Grid (export)	40.67	94.18	343.47	35.67	83.85	314.76
From battery	-	-	-	3.77	3.69	2.63
To battery	-	-	-	4.26	4.16	2.99
Expected state of charge (%)	-	-	-	32.93	31.91	42.92
Expected annual loss ($)	-399,625	-202,692	-475,231	-473,588	-280,207	-487,034
Value of stochastic solution ($/year)	-	-	272,539	-	-	206,827

5.2 Investigation of battery operation

The battery is shown to affect the results and optimal solution by distributing energy more evenly throughout the day and enabling energy imports at more advantageous hours. However, several simplifying assumptions have been made in modelling battery operation:

- Initial state of charge, being an operational decision, is assumed to have a value of 50%.
- Maximum allowable C-rate is assumed to be 0.25 in order to prevent high losses and fast performance fading.
- Available capacity by a minimum and maximum state of charge of 10% and 90% respectively to prevent increased performance fading.
- Energy efficiency is assumed to be constant and represented by a contribution of a charging and discharging energy efficiency both of 95%.

Potential dangers with the assumptions are investigated by sensitivity analyses of the above points, shown by objective value variations in Figures 5.5-5.8, and design decisions in Tables 5.3-5.6. The row in the tables with red text indicate the flexible design obtained previously, and the red dots in the Figures show their respective objective function values.

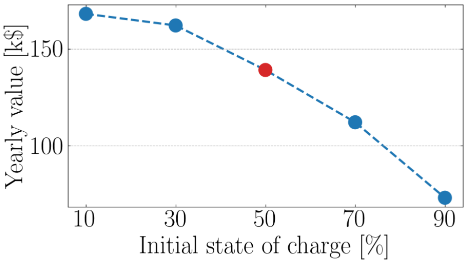


Figure 5.5: Objective value sensitivity to initial state of charge.

	Design		
Value [%]	PV $[m^2]$	WT $[no.]$	BAT $[MWh]$
10	48000	10	9
30	48000	10	9
50	48000	10	9
70	56000	10	3
90	40000	12	0

Table 5.3: Design sensitivity to initial state of charge.

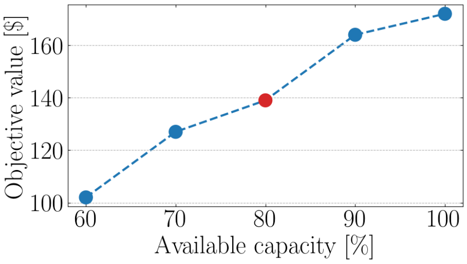


Figure 5.6: Objective value sensitivity to available capacity.

	Design		
Value [%]	PV $[m^2]$	WT $[no.]$	BAT $[MWh]$
60	56000	10	3
70	48000	10	9
80	48000	10	9
90	48000	10	9
100	48000	10	9

Table 5.4: Design sensitivity to available capacity.

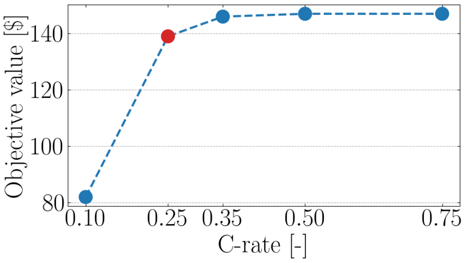


Figure 5.7: Objective value sensitivity to maximum C-rate.

	Design		
Value [%]	PV $[m^2]$	WT $[no.]$	BAT $[MWh]$
0.1	56000.0	10.0	3.0
0.25	48000.0	10.0	9.0
0.35	48000.0	10.0	7.5
0.5	48000.0	10.0	7.5
0.75	48000.0	10.0	7.5

Table 5.5: Design sensitivity to maximum C-rate.

Findings and interpretations from the analyses are summarized below:

- In this model, increasing initial state of charge decreased the value of the project. The cost of charging the battery at the end of each day seemed to be higher than the value obtained by having more power to spend at the beginning. The initial state of charge also had an impact on the optimal design when it was set too high. This indicate that the RES could have issues with charging the battery to the required level at the end of each day due to the periodic boundary condition.

- The available capacity increased the objective value of the project in a somewhat linear relationship. This is expected as, in principle, the available capacity changes the cost of storage capacity. For low available capacities, the program found the cost of energy storage too costly for covering certain differences between the demand load and the produced power.

- Maximum allowed C-rate seemed to potentially have a significant effect on the system. In the differences between the potential production patterns and the demand load pattern, there seems to be ranges of deficits and surpluses in power that is desirable to cover. For too low rates, the battery loses some of its value, not being able to offset certain power variations

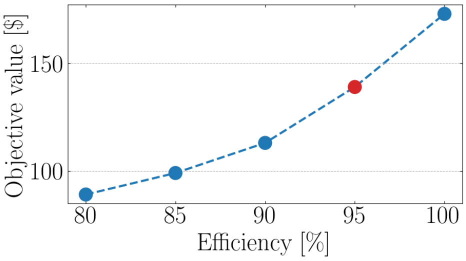


Figure 5.8: Objective value sensitivity to efficiency.

	Design		
Value [%]	PV $[m^2]$	WT $[no.]$	BAT $[MWh]$
80	40000.0	12.0	1.5
85	56000.0	10.0	3.0
90	48000.0	10.0	9.0
95	48000.0	10.0	9.0
100	48000.0	10.0	9.0

Table 5.6: Design sensitivity to efficiency.

at reasonable costs. For a maximum C-rate of 0.25, the program found that it would be beneficial to be able to cover deficits that require a certain amount of power. For higher C-rates than this, the program found no additional value of higher power, even reducing the size of the battery as the same amount of power could be achieved for a smaller battery. The rate of charging/discharging therefore is an important factor when designing the system.

- The energy efficiency on discharging and charging of the battery also seemed to affect the design decisions. Increased assumed efficiency logically increased the objective value, but the optimal design remained unaffected. Below 90%, the efficiency affected the design. The value of the battery seemingly was reduced to a significant extent. Thus, a flexible design was rather obtained by increasing production capacities and a smaller battery.

6 | Model discussion

6.1 Assumptions for modelling energy systems

Modelling uncertain parameters for stochastic environment

The results show how the type of environment, i.e. stochastic or deterministic, can affect the optimal design and how this can be taken into account for increased flexibility. In this work, a simple uncertainty modelling approach has been used for finding representative fluctuating patterns. Patterns were modelled to represent typical end users, costs of importing power, and characteristics of renewable energy sources. However, using only 4 typical days to represent seasonal variation and 3 scenarios of each season to represent fluctuations, the 12 scenarios have mainly illustrative purposes. Modelling uncertainty with higher accuracy would increase the variance of the uncertain parameters. Thus, a higher need for RES flexibility and a higher value of stochastic solution is expected for better representations of the stochastic environment. For results that can provide valuable insight in specific design processes, higher level of care should be taken when generating scenarios.

It is however shown that the LCOEs in the stochastic environment are found to be comparable to actual LCOEs in estimations provided by Lazard [55]. While solar PV are found to be in the top range, wind is in the found to be in the middle range of the estimations. The scenarios and data input are important drivers for the LCOE. As previously discussed, the value of renewable energy sources are dependent on their generating patterns, adding to the need for rigorous uncertainty modelling. The value of wind is deemed to be particularly sensitive to uncertainty modelling as the typical daily patterns are varying to a greater extent, both in form and relative magnitude. In turn, representative wind speeds may be harder to model. Solar radiation are determined by the sun and clearness of the atmosphere, as explained in Section 2.1. Therefore, the RES value of PV systems can be easier to estimate than the value of wind by using stochastic programs.

Modelling energy conversion

In the flexible design, the total expected power production of energy conversion technologies is reduced. Rather, in addition to increased energy storage capacity, the capacities of PV and wind are adjusted to provide a better match between supply and demand. In this work, only two energy conversion technologies were considered. However, the potential to obtain flexible designs by combination of different technologies is expected to increase with the number of energy sources available. For instance, hydroelectric energy and tidal energy can provide additional design flexibility.

The power models for solar and wind have simplifying assumptions that may be sources of uncertainty for the results, e.g. no degradation, solar power output to be linear to irradiance and an idealized power curve for the wind turbines. Even though the wind model is non-linear to wind speed, both models are linear from an optimization perspective as the wind speed is a parameter and not a decision variable. Non-linearity in an optimzation program can only be caused by decision-variables. Therefore, improvements in the models to reduce the effect of the assumptions do not necessarily lead to a non-linear program. Rather, the relationships between the decision variables and uncertain parameters may become increasingly non-linear. For instance, the efficiency of PV systems depend on uncertain parameteres such as irradiance and temperature.

Cost and revenue model

In this work, a simple cost modelling was applied with a linear construction cost and a factor of annualized cost representing maintenance. As a result, the cost of the smallest capacities may be underestimated as economies of scale usually can be expected for small plants under 5 MW [56]. However, the actual effect for the flexible design in this case is likely to be insignificant. Both capacities for the wind and solar farms were chosen to be around 10 MW in the stochastic program. As the largest effect of economies of scale occurs when changing type of application, e.g. residential, industrial or utility. Therefore, to use a constant cost for selection of a type of technology may be an adequate representation for economies of scale for a certain application.

The revenue generation model is expected to have a significant impact of the value of renewable systems, summarized as:

- Power used for covering the demand load generates revenue with a fixed price per unit of energy.
- All extra power exported to the grid is sold at a fixed price per unit of energy. That is 50% lower than the sum received when covering demand, which is for representing a drop of value of renewable energy during overproduction of electricity.

Without an upper limit on construction cost, the program seemed to significantly over-design the system. This indicates that the model may overestimate the value of renewables with increasing shares, reflecting the revenue-structure of the model. In particular, wind energy seemed to be over-designed. A possible explanation can be the cost to revenue ratio of wind energy. The LCOE of wind was only about 5% higher than the price received when exporting to the grid, meaning system experienced a minor penalty cost for producing too much wind energy. In reality, producing over four times as much as needed would potentially cause a crash in the electricity price. Therefore, a more accurate representation of the effect of renewables in energy markets could be required to not overestimate the renewable value, leading to an over-designed system.

Assumptions for energy storage

Assumptions for battery operation and battery parameters have been shown to potentially have a significant impact on the optimal flexible design in Section 5.2. While it may be logical to assume that available capacity and costs per MWh are the most determining factors, it is shown that it is not necessarily the case. The assumed rules for operation and for how the energy stored can be utilized, e.g. initial state of charge, available power and cost per MW could affect both expected profits and designs. If the cost per MW became too high, i.e. when the maximum allowed C-rate was low, offsetting a certain range of peaks and valleys in power generation were no longer profitable. It is argued that it is harder to assume appropriate maximum allowed C-rates for a model due to power surplus and deficit often coming in bulks of particular magnitudes.

An increased relative utilization of the battery in the stochastic program led to a higher system value. However, no costs associated with a higher utilization and more frequent cycling have been included in this model. As previously considered in Section 2.3, a battery's lifetime is often calculated by how many cycles that it lasts. In turn, the added value of a higher utilized battery is expected to have been potentially overestimated as it may induce a shorter battery lifetime. A high battery utlization is logically expected to be more valuable than a low utlization, but to a lesser extent than what is estimated in the program.

Battery degradation rate is also sped up by allowing an increased available capacity and C-rate, reducing future battery performance. Additionally, the instantaneous energy efficiency is affected by how the battery is operated. However, modelling degradation and battery behaviour with a higher accuracy inherently leads to non-linear problems. Battery flows and state of charge are decision variables and cannot be represented by model parameters. A non-linear battery model or modelling battery degradation will therefore produce a complex non-convex program and significantly complicate the optimization procedure.

State of charge was assumed to be fixed for the initial and final time steps in order to decouple the design days. However, requiring a certain level of energy stored at the end of each day could lead to a reduced value of the battery. The model assumes that a 50% state of charge will be reached at the end of each day. In this case, using a battery seem to be the most valuable when it is assumed to be empty at the start and end of each day. The periodic boundary could potentially be modelled as a supplementary battery design decision to reduce the risk of under-evaluating battery energy storage.

Furthermore, which hour periodicity for design days is assumed may also be a source of uncertainty for the results. Periodicity in this model is assumed at midnight, but it may be more fitting to aim for the coupling of design days at other hours. For instance, the best hours to aim for a certain battery level may be a couple of hours later or prior to midnight. It may even be possible to assume a periodic boundary condition with higher state of charges around noon, which is an hour with expected power surpluses in PV-dominated systems. The design day coupling and the energy generation patterns are therefore argued to be important for evaluation of battery energy storage, with a different set of general battery operational rules expected to be beneficial for systems dominated by different energy sources.

6.2 Two-stage stochastic programming framework

The inherent fluctuations and uncertainty present in renewable energy systems are accounted for by the method of stochastic programming, more specifically two-stage stochastic programming using the design day approach. The battery was not allowed to add flexibility through day-to-day energy storage due to the uncoupled design days used for problem formulation. It is expected to be a potential for day-to-day energy storage due to relatively high amounts of both imports and exports to grid. However, coupling of design days can prove to be hard to model using the two-stage SP framework. The realization of one design day affect the probability of the typical days that follow, but in a two-stage stochastic program non-anticipativity for decisions is only present once. Using scenarios with multiple days, such as a 'design week', will therefore operate with perfect information. This will entail a risk of over-evaluating the battery as the program will know with certainty how to operate the system for maximum utilization of the battery.

Additionally, perfect information is present when using design days in a two-stage SP. This is clearly illustrated in the RES operational profiles: The program is seen to prefer to charge the battery despite having a power deficit and reduce the amount of power imported at unfavourable prices. Even though the battery is not allowed to be charged by power directly from the grid, the optimizer uses the perfect information to foresee when importing to end user is beneficial. While some level of uncertainty will be present in all operational decisions, forecasting within 24 hours is expected to be fairly accurate for estimating power generation, grid prices and demand loads. Perfect information on operation within design days is therefore argued to not necessarily entail substantial risks of sub-optimal decisions due to over-evaluated energy storage.

6.3 Decomposition approach

The need for compatibility with the decomposition algorithm NGBD in the GOSSIP framework provided additional considerations to modelling choices and approach. In particular, two requirements were needed for NGBD to guarantee convergence to a global optimum:

1. Discrete set of first-stage variables

2. No constraints involving decision variables from multiple scenarios

The model produced was a MILP model, meaning NGBD essentially reduces to BD. Nevertheless, the model was verified for BD by comparison with the extension ANTIGONE's Full Space algorithm in GOSSIP (see Appendix D).

However, discretization of the first-stage variables may be a source of uncertainty. Optimal combination of design decisions may depend on the bounds and number of discrete values. For instance, building more solar panels has to be performed at increments of 8,000 m^2 when using 11 discrete values, despite the possibility of optimality in the interval between these values. The nominal design decision for solar PV is chosen to be the third lowest possible value, increasing the probability of a large relative gap to the actual optimal design.

Another possibly limiting factor of the NGBD algorithm is the need for the program to be scenario-wise decomposable. In this work, the restriction of independent scenarios may cause modelling limitations for some RES aspects. For instance, modelling expected yearly degradation of technologies or expected power flows as a method for coupling design days. For NGBD compatibility, only the objective function can include variables from several scenarios, meaning the inclusion as soft constraints is the only possibility.

Furthermore, the decomposition algorithm in GOSSIP is still under development. Consequently, it is found to be a higher need for a well defined model. In particular, issues with unnecessary constraints with only one decision variable and one parameter were identified. However, this is omitted in the final model. Additionally, the model should be provided with appropriate bounds when defining decision variables.

On the other hand, GOSSIP provides a framework for modelling and a way of efficiently obtaining solutions. There are possibilities for more accurate modelling of both technologies and uncertainty, including non-linear behaviour such as battery degradation. The GOSSIP software and modelling framework is therefore expected to be highly valuable for further development of this RES model for the flexible design problem.

7 | Final Remarks

7.1 Conclusions

Flexible designs for renewable energy systems are needed for mitigation of uncertain variations and fluctuations in the system's environment. Energy storage technologies, increased production capacities, and different combinations of energy sources are measures for improving flexibility, but these have to be considered at the design stage. The nature of renewables and energy systems lead to large and complex optimization problems subject to uncertainty. Thus, this work has investigated the modelling of the flexible design problem for renewable energy systems using the framework provided by the novel optimization software GOSSIP for its efficient solution:

1. A flexible design was obtained by a two-stage stochastic multi-period MILP model with hourly temporal resolution, created to be compatible with the non-convex generalized Benders decomposition algorithm in GOSSIP.
2. 12 design days based on typical fluctuating patterns were constructed for a discrete scenario representation of uncertainty in renewable energy systems.
3. The assumptions for including a linear battery model for hourly energy storage were investigated by a simple sensitivity analysis of the resulting parameters.

The results show that using a flexible design strategy in an uncertain environment can have significant value for a renewable energy system project. The expected amount of imported power is reduced by 26%, i.e. the energy security of the system is significantly improved. When constrained by a budget, increased energy storage and solar energy turned losses to profits compared to a design based on expected values. At the same time, the number of wind turbines and total expected power production were reduced. Test runs of the model without a budget show how uncertainty can be accounted for by increasing production capacities as well. Hence, the model is able to evaluate, compare and implement different measures for flexible designs.

Further, a notable difference in the cost of renewable energy sources compared to the value provided to the system is found. The scenarios were, on the other hand, significant for both cost and value of technologies. The risk of suboptimal decisions is increased when neglecting the effect of typical patterns of energy sources and their respective contributions to covering a demand load. In the assumed uncertain environment, the amount of installed wind power was reduced despite the LCOE of wind power decreased by 25%. The added flexibility by increasing the use of solar irradiation patterns and energy storage was found to outweigh the reduced LCOE for wind. Hence, a rigorous representation of the uncertainty and fluctuations in the system's environment is imperative for obtaining a flexible design.

Battery energy storage is shown to be able to mitigate fluctuations and counter-act the need for increased production capacities. The system is enabled to import power at favourable hours, causing a reduction in the associated penalty cost of power deficits. Furthermore, the battery was found to generate value for the system by storing surplus power for future deficits. Thus, energy storage could counter-act the need for increased production capacities and construction costs, but at the same time, improve security of energy supply.

The assumptions for battery operation in the model were found to potentially have a significant effect on optimal design and the expected values of program variables. This indicates that the estimated value of the battery has a risk of being inaccurate: Initial state of charge, assumed efficiency, available capacity and maximum allowable C-rate all showed potential to affect the value of the battery. In particular, the C-rate was shown to produce a sharp relative decline for low values. Furthermore, the battery increased its value when reducing state of charge at the boundaries of the design days.

The developed model can be used as a foundation for further study of the RES flexible design problem. The validity of simplifying assumptions can be investigated by including degradation and other non-linear aspects of the technologies in the model. Furthermore, in order to obtain a useful design for real-life applications, extensive modelling of the stochastic environment is important. GOSSIP provides a framework for modelling and solving these large and non-convex two-stage stochastic MINLPs. Thus, GOSSIP are found to be valuable for the further study of the flexible design problem for renewable energy systems.

7.2 Future work

The scope of this thesis was limited to the creation of an MILP model for the flexible renewable energy system design problem, and investigation of the modelling assumptions and framework. Modelling uncertainty and the resulting 12 scenarios were used for illustrative purposes. Thus, future work on the topic should include:

- A flexible design case study with rigorous uncertainty modelling.
- Further investigation and development of the RES model.

A case study with a more accurate representation of the fluctuating patterns in the system's uncertain environment should include:

- Adding temperature in the design days.
- Developing a method for constructing numerous scenarios for representation of uncertainty.

Further development of the model should also be a future priority for improving its validity, including:

- Investigating and implementing more energy carriers, energy sources, energy conversion technologies, and energy storage technologies.
- Implementing net present value as the objective function.
- Investigating and implementing methods for including degradation in the model.
- Investigating and implementing of non-linear models for battery storage.
- Implementing of different cost and revenue generation models, particularly to represent a penalty on overproduction.
- Developing of the solar PV model further, e.g. include efficiency dependence on irradiance and temperature.
- Investigating the effect of discretization of design decisions.
- Investigating of possible improvements to the periodic boundary conditions and design day coupling.

Finally, an aim should be the development of an easily extendable object-oriented framework with a user-friendly interface for modelling of renewable energy systems in GOSSIP. This would make the software more accessible to users with less hands-on knowledge of GOSSIP, and allowing for simpler constructions, extensions, and modifications of renewable energy system models [57].

Bibliography

[1] Rina Zeller, Agustin Delgado, Gerard Reid, and Kirsten Panerali. “Wind and solar PV will keep taking the lead”. *Global Future Council on Energy Technologies* August 202 (2020), p. 5.

[2] IRENA. *Renewable Power Generation Costs in 2019*. Tech. rep. International Renewable Energy Agency, 2020.

[3] IEA. *Projected costs of generating electricity 2020*. Tech. rep. International Energy Agency, Paris, 2020.

[4] IRENA. *Utility-scale batteries*. Tech. rep. International Renewable Energy Agency, 2019.

[5] Omar J. Guerra, Jiazi Zhang, Joshua Eichman, Paul Denholm, Jennifer Kurtz, and Bri-Mathias Hodge. “The value of seasonal energy storage technologies for the integration of wind and solar power”. *Energy and Environmental Science* 13 (7 2020), pp. 1909–1922. DOI: `10.1039/D0EE00771D`.

[6] John R. Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[7] Paolo Gabrielli, Florian Fürer, Georgios Mavromatidis, and Marco Mazzotti. “Robust and optimal design of multi-energy systems with seasonal storage through uncertainty analysis”. *Applied Energy* 238 (Mar. 2019), pp. 1192–1210. DOI: `10.1016/j.apenergy.2019.01.064`.

[8] Georgios Mavromatidis, Kristina Orehounig, and Jan Carmeliet. “Design of distributed energy systems under uncertainty: A two-stage stochastic programming approach”. *Applied Energy* 222 (2018), pp. 932–950. DOI: `https://doi.org/10.1016/j.apenergy.2018.04.019`.

[9] Yun Yang, Shijie Zhang, and Yunhan Xiao. “Optimal design of distributed energy resource systems based on two-stage stochastic programming”. *Applied Thermal Engineering* 110 (2017), pp. 1358–1370. DOI: `https://doi.org/10.1016/j.applthermaleng.2016.09.049`.

[10] Zhe Zhou, Jianyun Zhang, Pei Liu, Zheng Li, Michael C. Georgiadis, and Efstratios N. Pistikopoulos. “A two-stage stochastic programming model for the optimal design of distributed energy systems”. *Applied Energy* 103 (2013), pp. 135–144. DOI: `https://doi.org/10.1016/j.apenergy.2012.09.019`.

[11] Jens Götze, Jonte Dancker, and Martin Wolter. “A general MILP based optimization framework to design Energy Hubs”. *at - Automatisierungstechnik* 67.11 (2019), pp. 958–971. DOI: `doi:10.1515/auto-2019-0059`.

[12] Araz Ashouri, Samuel S. Fux, Michael J. Benz, and Lino Guzzella. “Optimal design and operation of building services using mixed-integer linear programming techniques”. *Energy* 59 (2013), pp. 365–376. DOI: `https://doi.org/10.1016/j.energy.2013.06.053`.

[13] Can Li and Ignacio E. Grossmann. “A review of stochastic programming methods for optimization of process systems under uncertainty”. *Frontiers in Chemical Engineering* 2 (2020), p. 34.

[14] Rohit Kannan. “Algorithms, analysis and software for the global optimization of two-stage stochastic programs”. PhD thesis. Massachusetts Institute of Technology, 2018.

[15] Rohit Kannan and Paul I. Barton. “GOSSIP documentation” (2018).

[16] Agora Energiewende and Ember. *The European Power Sector in 2020: Up-to-Date Analysis on the Electricity Transition.* 2021. URL: https://ember-climate.org/wp-content/uploads/2021/01/Report-European-Power-Sector-in-2020.pdf.

[17] Mohammad Mohammadi, Younes Noorollahi, Behnam Mohammadi-ivatloo, and Hossein Yousefi. "Energy hub: From a model to a concept – A review". *Renewable and Sustainable Energy Reviews* 80 (2017), pp. 1512–1527. DOI: https://doi.org/10.1016/j.rser.2017.07.030.

[18] IEA. *Solar PV.* Tech. rep. International Energy Agency, Paris, 2020. URL: https://www.iea.org/reports/solar-pv.

[19] U.S. Department of Energy/Energy Efficiency and Renewable Energy. *Solar Radiation Basics.* URL: https://www.energy.gov/eere/solar/solar-radiation-basics (visited on 04/15/2021).

[20] Christian A. Gueymard. "Direct and indirect uncertainties in the prediction of tilted irradiance for solar engineering applications". *Solar Energy* 83.3 (2009), pp. 432–444. DOI: https://doi.org/10.1016/j.solener.2008.11.004.

[21] Satel-Light. *Satel-Light: Your site outdoor information.* URL: http://www.satel-light.com/indexs.html (visited on 04/15/2021).

[22] Tomas Markvart. *Solar electricity.* 2nd ed. Wiley, 2000.

[23] PVEducation. *IV curve.* URL: https://www.pveducation.org/pvcdrom/solar-cell-operation/iv-curve (visited on 04/28/2021).

[24] Sustainable Technologies Evaluation Program. *The effect of irradiance and temperature on the performance of photovoltaic modules.* Tech. rep. Toronto and Region Conservation Authority, 2017.

[25] Energy Sage. *Most Efficient Solar Panels: Solar panel efficiency explained.* URL: https://news.energysage.com/what-are-the-most-efficient-solar-panels-on-the-market/ (visited on 04/19/2021).

[26] PVEducation. *Degradation and failure modes.* URL: https://www.pveducation.org/pvcdrom/modules-and-arrays/degradation-and-failure-modes (visited on 04/28/2021).

[27] Andy Schell. *What are the best solar panels on the market?* Paradise Energy Solutions. URL: https://www.paradisesolarenergy.com/blog/what-are-the-best-solar-panels-on-the-market (visited on 04/28/2021).

[28] Mahesh M. Bandi. "Spectrum of wind power fluctuations". *Physical Review Letters* 118.2 (2017), p. 028301.

[29] IEA. *Renewables 2020.* Tech. rep. International Energy Agency, Paris, 2020. URL: https://www.iea.org/reports/renewables-2020.

[30] D. L. Elliott, C. G. Holladay, W. R. Barchet, H. P. Foote, and W. F. Sandusky. "Wind energy resource atlas of the United States" (1987). URL: https://www.osti.gov/biblio/6628633.

[31] Sathyajith Mathew. *Wind energy: fundamentals, resource analysis and economics.* Springer, 2006.

[32] R. Veena, S. Mathew, and M.I. Petra. "Artificially intelligent models for the site-specific performance of wind turbines". *International Journal of Energy and Environmental Engineering* 11.3 (2020), pp. 289–297.

[33] Iain Staffell and Richard Green. "How does wind farm performance decline with age?" *Renewable Energy* 66 (June 2014), pp. 775–786. DOI: 10.1016/j.renene.2013.10.041.

[34] Terrence Xu, Wei Wang, Mikhail L Gordin, Donghai Wang, and Daiwon Choi. "Lithium-ion batteries for stationary energy storage". *JOM* 62.9 (2010), pp. 24–30.

[35] Federico Baronti, N. Femia, Roberto Saletti, C. Visone, and Walter Zamboni. "Hysteresis modeling in li-Ion batteries". *IEEE Transactions on Magnetics* 50 (Nov. 2014), pp. 1–4. DOI: 10.1109/TMAG.2014.2323426.

[36] D. Linden and T.B. Reddy. *Handbook of batteries.* 3rd ed. McGraw-Hill, 2001.

[37] M. M. Kabir and Dervis Emre Demirocak. “Degradation mechanisms in Li-ion batteries: a state-of-the-art review”. *International Journal of Energy Research* 41.14 (2017), pp. 1963–1986. DOI: https://doi.org/10.1002/er.3762.

[38] Evelina Wikner and Torbjörn Thiringer. “Extending battery lifetime by avoiding high SOC”. *Applied Sciences* 8.10 (2018), p. 1825.

[39] Zachary Shahan. *38,000 Tesla Powerwall reservations in under a week (Tesla / Elon Musk transcript).* Cleantechnica. 2015. URL: https://cleantechnica.com/2015/05/07/38000-tesla-powerwall-reservations-in-under-a-week-tesla-elon-musk-transcript/ (visited on 04/28/2021).

[40] Paul I. Barton. “Mixed-Integer and Nonconvex Optimization”. *Notes* (2007).

[41] Stephen M. Robinson. *Numerical Optimization.* 2nd. 2006.

[42] Xiang Li, Asgeir Tomasgard, and Paul I. Barton. “Nonconvex Ggeneralized Benders Ddcomposition for stochastic separable mixed-integer nonlinear programs”. *Journal of Optimization Theory and Applications* 151 (2011), pp. 425–454. DOI: https://doi.org/10.1007/s10957-011-9888-1.

[43] Ruth Misener and Christodoulos A Floudas. “ANTIGONE: algorithms for continuous/integer global optimization of nonlinear equations”. *Journal of Global Optimization* 59.2-3 (2014), pp. 503–526.

[44] Martin Geidl and Göran Andersson. “Optimal power flow of multiple energy carriers”. *IEEE Transactions on Power Systems* 22.1 (2007), pp. 145–155. DOI: 10.1109/TPWRS.2006.888988.

[45] Georgios Mavromatidis, Kristina Orehounig, and Jan Carmeliet. “A review of uncertainty characterisation approaches for the optimal design of distributed energy systems”. *Renewable and Sustainable Energy Reviews* 88 (2018), pp. 258–277. DOI: https://doi.org/10.1016/j.rser.2018.02.021.

[46] Fernando Dominguez-Munoz, Jose M Cejudo-Lopez, Antonio Carrillo-Andres, and Manuel Gallardo-Salazar. “Selection of typical demand days for CHP optimization”. *Energy and Buildings* 43.11 (2011), pp. 3036–3043.

[47] Paolo Gabrielli, Alessandro Poluzzi, Gert Jan Kramer, Christopher Spiers, Marco Mazzotti, and Matteo Gazzani. “Seasonal energy storage for zero-emissions multi-energy systems via underground hydrogen storage”. *Renewable and Sustainable Energy Reviews* 121 (2020), p. 109629. DOI: https://doi.org/10.1016/j.rser.2019.109629.

[48] *Elkraft Avedøre 1MW.* Wind-turbine-models. 2015. URL: https://en.wind-turbine-models.com/turbines/1166-elkraft-aved-re-1mw (visited on 05/06/2021).

[49] Battery University. *BU-802b: What does elevated self-discharge do?* URL: https://batteryuniversity.com/learn/article/elevating_self_discharge (visited on 05/06/2021).

[50] Weather Spark. *Average weather in Berlin Germany.* URL: https://weatherspark.com/y/75981/Average-Weather-in-Berlin-Germany-Year-Round (visited on 04/15/2021).

[51] Muhamad Reza, AM Van Voorden, PH Schavemaker, GC Paap, and Lou Van der Sluis. “Implementation of renewable electrical energy generation in an urban distribution network: Impacts of energy storage and demand growth”. *17th International Conference on Electricity Distribution, CIRED* (2003).

[52] Energy Facts Norway - Norwegian Ministry of Petroleum and Energy. *The power market.* URL: https://energifaktanorge.no/en/norsk-energiforsyning/kraftmarkedet (visited on 04/16/2021).

[53] Fraunhofer Institute for Solar Energy Systems ISE. *Average spot market prices.* URL: https://energy-charts.info/charts/price_average/chart.html=en&c=DE&interval=month&year=2019 (visited on 04/16/2021).

[54] Tobias Bobmann and Iain Staffell. “The shape of future electricity demand: Exploring load curves in 2050s Germany and Britain”. *Energy* 90 (2015), pp. 1317–1333.

[55] Lazard. *Lazard´s Levelized Cost of Energy Analysis*. 2020. URL: `https://www.lazard.com/media/451419/lazards-levelized-cost-of-energy-version-140.pdf`.

[56] John Farrel. *Is bigger best in renewable energy?* Institute for Local Self-Reliance, 2016. URL: `https://ilsr.org/report-is-bigger-best/`.

[57] Mari E. Rugland. “Object-oriented framework for the optimization of flexible renewable energy systems”. MA thesis. Norwegian University of Science and Technology, 2021.

[58] Marshall Brian. *How lithium-ion batteries work*. Howstuffworks.

Appendices

A Principles of renewable energy technologies

In this appendix, brief overviews of the working principles of solar cells, pitch controlled wind turbines, and batteries are presented.

A.1 Photovoltaics

The concept of photovoltaics, illustrated in Figure A.1, works by three main mechanism:

1. Electron-hole creation by absorbing sunlight
2. Electron-hole separation to opposite sides
3. Electron-hole recombination through an external circuit

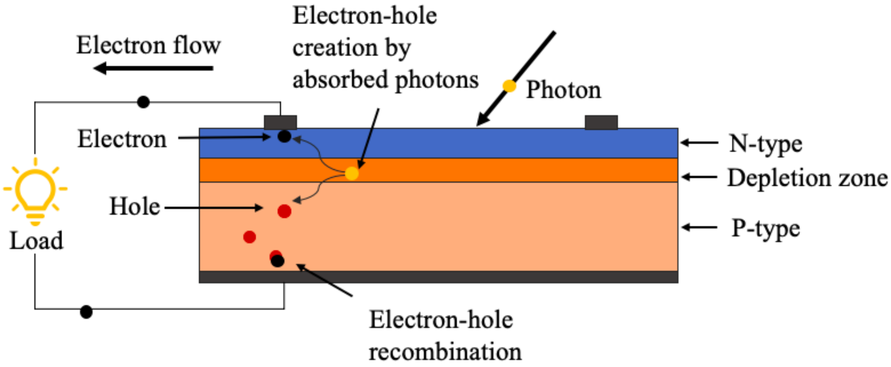


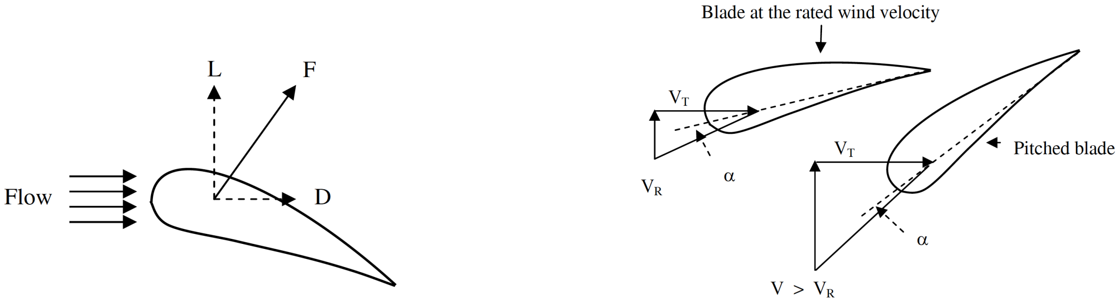
Figure A.1: Concept of photovoltaics.

Semiconductors are materials that are conductors under certain circumstances, for instance when the solar radiation creates a large enough energy field when used in solar cells. The solar cell consists of a p-type cell, doped with atoms of one less valence electron than the semiconductor material, making the cell positively charged. On the other hand, an n-type cell has a negative charge due to doping with atoms with one more valence electron. The solar cell joins the two types together in a pn-junction, which depletes the interface for free electrons and holes, and will only allow charge to move across it if an external voltage or electric field is present. When solar radiation is incident on the solar cell, the photons can give electrons in the depletion zone enough energy to move across the electric field in the junction and create electron-hole pairs. This leads to a voltage difference between the two sides and by connecting the two semiconducturs the electron can move towards the holes on the other side, creating electric current.

A.2 Rotational motion and pitch of rotor blades

The rotational motion of the wind turbine is produced when air flows over the curved blades and creates a lift [31], illustrated in Figure A.2a. The lift, the perpendicular component of the total force, F, relative to the air flow, is caused by a pressure difference on the top and bottom side of the blade.

In addition to the wind speed, this pressure difference is determined by the angle of attack on the blade, illustrated in Figure A.2b by the parameter α. Thereby, changing the pitch of the blades can be used to control the rotational motion. Specifically for pitch controlled wind turbines, the blade velocity, V_T, can be held constant if the wind velocity, V, becomes greater than the rated wind speed, V_R, by reducing the angle of attack.



(a) Force creation on blade

(b) Blade velocity by pitch control

Figure A.2: Creating rotational motion in a wind turbine [31]

A.3 Li-ion batteries

Li-ion batteries are a type of rechargeable batteries. Rechargeable batteries can be used to store and use energy by conversion between electrical energy and chemical energy. Charging the battery will convert electric energy to chemical energy in the cell, and during discharge, the energy can be extracted by conversion of chemical energy to electricity. Operations during charging and discharging of a lithium-ion battery are illustrated in Figure A.3.

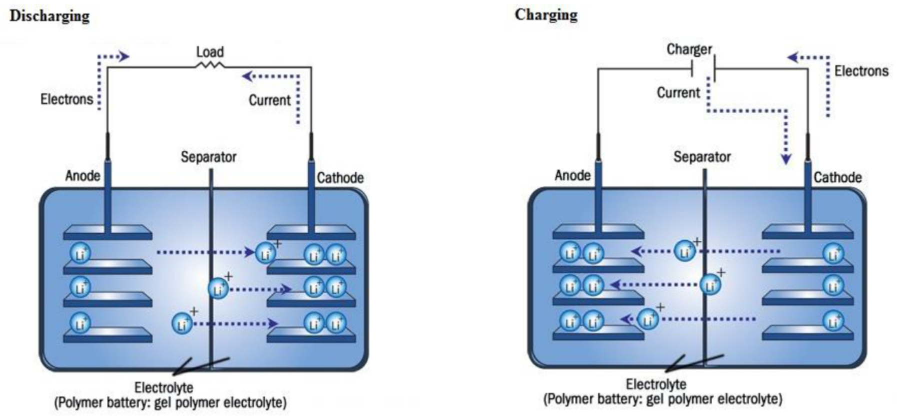


Figure A.3: Operation of a lithium-ion battery [58]

The Li-ion battery consists of cells, an anode and a cathode, divided by a separator. This separator only allows for transportation of lithium ions across it, while the electrons have to be transported through an external circuit. The transportation of Li-ions is governed by the potential differences between the two electrodes. A Li-ion will always seek its stable state in the cathode side, but applying an external voltage source will force the Li-ions over in an unstable state in the anode. Hence, charging the battery. Due to the separator, the battery will remain charged until the two electrodes are connected to a load, creating a path for the electrons and discharges the battery.

B | Model File

```cpp
// Flexible RES design problem, model file
#include <iostream>
#include <string>
#include <vector>
#include <fstream>
#include <sstream>
#include <map>
#include "definitions.hpp"
#include "CompGraph.hpp"
#include <cmath>
#include "inputmodel.hpp"

using namespace std;

int inputmodel(vector<double> &probabilities)
{
//--------------------------------------------------------------------
//Controls
bool setDesignsManually = false;        // For finding EEVP
vector<double> setDesign{3-1,8-1,5-1};  // Nominal design found in EVP
                                        // No. of chosen discrete val.
bool useBattery = true;                 // Battery in system

bool expectedValueProblem = false;      // EVP

bool printFile =false;                  // Print Scenarios
string uncertiantyFolder = "ScenarioSet4/";


//--------------------------------------------------------------------
// Input data and program set-up
int numScen         = 0;    //Number of Scenarios [-]
int numTimesteps  = 24;   //Number of timesteps [-]
double deltaT   = 1;    // length of a timestep [h]
int numParams   = 4;    //Number of uncertain parameters
int numTech     = 3;    //Number of technologies:
                                //1: PV. 2: WT. 3: BAT
int concount      = -1;
int varcount      = -1;
char clabel[75];

//--------------------------------------------------------------------
// Cost Parameters
double r = 0.06;                     //Interest rate [-]
vector<double> L{30,20,15};            //Lifetime [years]
vector<double> C{130,1200000,180000};  //Linear cost per unit size
                                       //$/{m^2,no.,Mwh}
vector<double> xi{0.05,0.05,0.02};     //Maintenance factor [-]
double budget = 20*1000000;            //Cap on construction costs [$]

//--------------------------------------------------------------------
// Discretization params
vector<int> numDiscrete{10,10,10};      //Number of discrete values
vector<double> Z_UB {72000,18,13.5};    //Upper bounds {m^2,no.,Mwh}
vector<double>Z_LB{0,0,0};              //Lower bounds {m^2,no.,Mwh}
```

```cpp

//--------------------------------------------------------------------
// Model Parameters: Grid
double FiT      = 70;   // Renewable Feed-in-tariff [$/MWh]
double FiT_extra  = 0.5*FiT;  // Extra earnings [$/MWh]
//--------------------------------------------------------------------
// Model Parameters: Renewable power gen
double eta_PV = 0.2;    //Efficiency PV [-]

double q_d  = 1         //Rated power for wind turbine [MW]
double W_d = 14;        //Rated wind velocity [m/s]
double W_min = 4;       //Cut-in wind velocity [m/s]
double W_max = 25;      //Cut-off wind velocity [m/s]

//--------------------------------------------------------------------
// Model Parameters: battery
double SoC_initial  = 50;     //50; Initial SoC of battery [%]
double SoC_min    = 10;     //10; min SoC [%]
double SoC_max    = 90;     //90; max SoC [%]
double eta_ES_storage = 0.998;  //Representing eergy storage self-
                                    // discharge losses
double eta_ES_ch    = 0.95;   //0.95; Battery charging losses
double eta_ES_dis   = 0.95;   //0.95; Battery discharging losses
double C_rate_max   = 0.25;   //0.25; Battery max C-rate

//--------------------------------------------------------------------
// Calculating Bounds
    double max_f_dem = 6;
    double max_f_PV = 1000;

double bound_PV = Z_UB[0]*eta_PV*max_f_PV/1000000*1.1;//MW, 5% margin
double bound_WT = Z_UB[1]*q_d*1.1;                  //MW, 5% margin
double bound_BAT = Z_UB[2]*C_rate_max*1.1;          //MW, 5% margin
double bound_DEM = max_f_dem*1.1;                       //MW, >5% margin

//--------------------------------------------------------------------
// Importing scenarios and uncertain data from csv-file
//  Iinitializing
int t=0;
int it_line=0;      //Counter for line-iterator
int it_cell=0;      //Counter for cell-iterator
ifstream file;
string filepath;
string cell;
string line;
string param = "param0";
string paramName;
map<string,vector<vector<double>>> uncertainParams; //Map for data

//  Scenario probabilities from csv to program
    //File with probabilities has to be called probabilities.csv
file.open(uncertiantyFolder+"probabilities.csv");
if (!file.is_open())
{
    cout<<"Error opening file for probabilities"<<endl;
    return -1;
}
while(getline(file, line))
{
    string comma(",");
    if(printFile){ cout<<line<<endl;}
    if(it_line!=0)
    {
        stringstream ss(line);
        while (getline(ss,cell,';'))
        {
            if(it_cell == 1)
            {
                if (cell.find(comma) != string::npos)
                {
                    probabilities.push_back(stod(cell.replace(cell.find(comma),
    comma.length(),".")));
                }
                else
                {
                    probabilities.push_back(stod(cell));
                }
                numScen++;
            }
            it_cell++;
        }
    }
    it_line++;
    it_cell=0;

}
file.close();
if (file.is_open())
{
    cout<<"Error closing file for probabilities"<<endl;
    return -1;
}
// Uncertain parameter values from csv to program
    // Have to be called param1.csv, param2.csv, param3.csv ..
vector<vector<double>> temp(numTimesteps,vector<double>(numScen));
for (int i=0;i<numParams;i++)
{
    it_line =0;
    param.pop_back();
    param+=to_string(i+1);
    filepath = uncertiantyFolder+param+".csv";
    file.open(filepath);
    string comma(",");
    if (!file.is_open())
    {
        cout<<"Error opening file for "<<param<<endl;
        return -1;
    }
    if (printFile){cout << param <<":"<<endl;}
    //Reading file and updating map
while(getline(file, line))
{
    if (printFile){cout <<line<<endl;}
    if(it_line!=1)
    {
    stringstream ss(line);
    while (getline(ss,cell,';'))
    {
        if(it_line==0 && it_cell==1)
        {
            paramName=cell;         //Parameter identifier in map
            uncertainParams.insert(pair<string,vector<vector<double>>>(paramName,
    temp));
        }
        else if (it_line>1 && it_cell==0)
        {
            t = stoi(cell);
        }
        else if(it_line>1 && it_cell>0 )
        {
        if (cell.find(comma) != string::npos)
        {
        uncertainParams[paramName][t-1][it_cell-1] = stod(cell.replace(cell.find(
    comma),comma.length(),"."));
        }
        else
        {
            uncertainParams[paramName][t-1][it_cell-1] = stod(cell);
        }
        }
        it_cell++;
    }
    }
    it_line++;
    it_cell=0;
}
    file.close();
    if (file.is_open())
    {
    cout<<"Error closing file for "<< paramName<<endl;
    return -1;
    }
}

if (printFile) // Printing data read from files
{
    cout <<"-------printing data---------"<<endl;
    cout<<"prob: ";
    for (int h=0;h<numScen;h++)
    {
        cout << "scen "<< h+1<<" "<< probabilities[h]<<endl;
    }
    map<string,vector<vector<double>>>::iterator it_map;
    for (it_map = uncertainParams.begin(); it_map != uncertainParams.end(); it_map
    ++)
    {
        cout << "Param: "<<it_map->first<<endl;   // string (key)
        for (int h=0;h<numScen;h++)
        {
            cout << " scen: "<<h+1<<endl;
            for (int t=0;t<numTimesteps;t++)
            {
                cout <<" "<< it_map->second[t][h]<< " ";  // string's value
            }
            cout << endl;
        }
    }
}

//-------------------------------------------------------------------
//Expected value problem
// --> Alter the first scenario of uncertainParams and probability
//     to represent EVP using expected values of uncertainty

if(expectedValueProblem)
{
    cout<<"EVP running.. \n Calculating EVPs.."<<endl;
    double avg;
    map<string,vector<vector<double>>>::iterator it_map;
    for (it_map = uncertainParams.begin(); it_map != uncertainParams.end(); it_map
    ++)
    {
        cout <<it_map->first<<": "<<endl;
        for (int t=0;t<numTimesteps;t++)
        {
             avg=0;
            for (int h=0;h<numScen;h++)
            {
                avg += probabilities[h]*it_map->second[t][h];
            }
            it_map->second[t][0]=avg;
            cout <<it_map->second[t][0]<<" ";
        }
        cout <<endl;
    }
    probabilities.erase(probabilities.begin()+1,probabilities.end());
    probabilities[0] = 1;
    numScen = 1;
}

```

```
//-------------------------------------------------------------------
// 1st stage variables and discretization

vector<vector<double>> z_d(numTech);
vector<vector<Variables>> z_b(numTech);
for (int tech =0;tech<numTech;tech++)
{
    z_d[tech].resize(numDiscrete[tech]);
    z_b[tech].resize(numDiscrete[tech]);
}
cout << "Discrete Sets of technologies:"<<endl;
for (int tech=0;tech<numTech;tech++)
{
    cout <<"Tech " <<tech+1<<": "<<endl;
    for (int n=0;n<numDiscrete[tech];n++)
    {
        if(!useBattery && tech==2)
        {
            continue;
        }
        z_d[tech][n] = Z_LB[tech] + n*Z_UB[tech]/(numDiscrete[tech]-1);
        cout << z_d[tech][n]<<"\t";

        sprintf(clabel, "z_b[%d][%d]",tech+1,n+1);
        z_b[tech][n].setIndependentVariable(++varcount,compgraph::BINARY,I(0,1)
    ,0.,-1,clabel);
    }
    cout<<endl;
}
cout<<endl;

cout<<"-- Discretization performed.. --"<<endl;
//-------------------------------------------------------------------
// 2nd Stage Variables

//  Declaring variables
//Renewable power for demand
vector<vector<Variables>>f_renewables(numTimesteps,
        vector<Variables>(numScen)); // [MW]

//vector<vector<Variables>>f_demand(numTimesteps,
//    vector<Variables>(numScen)); // [MW]

// PV
vector<vector<Variables>>f_PV(numTimesteps,
        vector<Variables>(numScen)); // [MW]
//WT
vector<vector<Variables>>f_WT(numTimesteps,
        vector<Variables>(numScen)); // [MW]
//grid
vector<vector<Variables>>f_grid_export(numTimesteps,
        vector<Variables>(numScen)); // [MW]

vector<vector<Variables>>f_grid_import(numTimesteps,
        vector<Variables>(numScen)); // [MW]
//Battery
vector<vector<Variables>>f_bat_charge(numTimesteps,
        vector<Variables>(numScen)); // [MW]
vector<vector<Variables>>f_bat_discharge(numTimesteps,
        vector<Variables>(numScen)); // [MW]
vector<vector<Variables>>E_BAT(numTimesteps,
        vector<Variables>(numScen)); // [%]

//  Setting variables
for (int t=0;t<numTimesteps;t++)
{
    for (int h=0;h<numScen;h++)
    {
        sprintf(clabel,"f_renewables[%d][%d]",t+1,h+1);
        f_renewables[t][h].setIndependentVariable(++varcount,compgraph::CONTINUOUS,
    I(0,(bound_WT+bound_PV)),0.,h+1,clabel);

        //sprintf(clabel,"f_demand[%d][%d]",t+1,h+1);
        //f_demand[t][h].setIndependentVariable(++varcount,compgraph::CONTINUOUS,I
    (0,bound_DEM),0.,h+1,clabel);

        sprintf(clabel,"f_PV[%d][%d]",t+1,h+1);
        f_PV[t][h].setIndependentVariable(++varcount,compgraph::CONTINUOUS,I(0,
    bound_PV),0.,h+1,clabel);

        sprintf(clabel,"f_WT[%d][%d]",t+1,h+1);
        f_WT[t][h].setIndependentVariable(++varcount,compgraph::CONTINUOUS,I(0,
    bound_WT),0.,h+1,clabel);

        sprintf(clabel,"f_grid_export[%d][%d]",t+1,h+1);
        f_grid_export[t][h].setIndependentVariable(++varcount,compgraph::CONTINUOUS
    ,I(0,(bound_WT+bound_PV)),0.,h+1,clabel);

        sprintf(clabel,"f_grid_import[%d][%d]",t+1,h+1);
        f_grid_import[t][h].setIndependentVariable(++varcount,compgraph::CONTINUOUS
    ,I(0,bound_DEM),0.,h+1,clabel);

        if(!useBattery) // Skip rest of loop
        {
            continue;
        }

        sprintf(clabel,"f_bat_charge[%d][%d]",t+1,h+1);
        f_bat_charge[t][h].setIndependentVariable(++varcount,compgraph::CONTINUOUS,
    I(0,bound_BAT),0.,h+1,clabel);

        sprintf(clabel,"f_bat_discharge[%d][%d]",t+1,h+1);
        f_bat_discharge[t][h].setIndependentVariable(++varcount,compgraph::
    CONTINUOUS,I(0,bound_BAT),0.,h+1,clabel);

        sprintf(clabel,"E_BAT[%d][%d]",t+1,h+1);
        E_BAT[t][h].setIndependentVariable(++varcount,compgraph::CONTINUOUS,I(0,
    Z_UB[2]*1.1),0.,h+1,clabel);
    }
}

//-------------------------------------------------------------------
//Design decisions: Linking 1st stage to second stage
vector<vector<Variables>>Z(numTech,vector<Variables>(numScen));
for (int tech =0;tech<numTech;tech++)
{
    if(!useBattery && tech ==2)
        {
            continue;
        }
    for (int h=0;h<numScen;h++)
    {
        sprintf(clabel,"Z[%d][%d]",tech,h);
        Z[tech][h].setIndependentVariable(++varcount,compgraph::CONTINUOUS,I(Z_LB[
    tech],Z_UB[tech]),0.,h+1,clabel);
    }
}
cout<<"-- 2nd stage varaiables declared.. --"<<endl;

//-------------------------------------------------------------------
// Objective function
vector<double> CRF(numTech);  // Capital recovery factor [-]
for (int tech=0;tech<numTech;tech++)
{
    if(!useBattery && tech==2)  // Skip battery
        {
            continue;
        }
    CRF[tech] = r*pow(r+1,L[tech])/(pow(r+1,L[tech])-1);
}
```

```

vector<Objective>obj(numScen);  // Total cost per scenario
for (int h=0; h<numScen;h++)
{
    obj[h]  = 0;
    // Investment cost
    for (int tech=0;tech<numTech;tech++)
    {
        if(!useBattery && tech==2)
        {
            continue;
        }
        // Annual investemnt + maintenance
        obj[h] += Z[tech][h]*C[tech]*CRF[tech]*(1+xi[tech]);
    }
    //Operating cost (revenue)
    for (int t=0 ; t<numTimesteps ; t++)
    {

        obj[h] += 365*uncertainParams["OC_GRID"][t][h]
                     *f_grid_import[t][h]*deltaT;
        obj[h] -= 365*f_grid_export[t][h]*FiT_extra*deltaT;
        obj[h] -= 365*(uncertainParams["L_DEM"][t][h]
                     -f_grid_import[t][h])*FiT*deltaT;

    }
    obj[h].setDependentVariable(++concount,compgraph::OBJ,true,h+1);
}
cout<<"-- Objective function set.. --"<<endl;

//--------------------------------------------------------------------
// 1st stage constraints
//  Only one 'true' binary variable per technology
vector<Constraints> con_binary(numTech);
for (int tech=0;tech<numTech;tech++)
{
    if(!useBattery && tech==2)
        {
            continue;
        }
    con_binary[tech] = -1;
    for (int n=0;n<numDiscrete[tech];n++)
    {
        con_binary[tech] += z_b[tech][n];
    }
    con_binary[tech].setDependentVariable(++concount,
                                          compgraph::EQUALITY,
                                          false,-1);
}

// 1st stage binary variable --> 2nd stage continuous
//  Sizes equal to 1st stage for all scenarios
vector<vector<Constraints>> con_designs(numTech,
        vector<Constraints>(numScen));
for (int tech=0;tech<numTech;tech++)
{
    if(!useBattery && tech==2)
        {
            continue;
        }
    for (int h=0;h<numScen;h++)
    {
        con_designs[tech][h] = -Z[tech][h];
        for (int n=0;n<numDiscrete[tech];n++)
        {
            con_designs[tech][h] += z_b[tech][n]*z_d[tech][n];
        }
        con_designs[tech][h].setDependentVariable(++concount,
                                                  compgraph::EQUALITY,
                                                  true,h+1);
    }
}

// Budget constraint

Constraints con_budget;
con_budget = -budget;
for (int tech=0;tech<numTech;tech++)
{
    for (int n=0;n<numDiscrete[tech];n++)
    {
        if(!useBattery && tech==2)
        {
            continue;
        }
        con_budget += z_b[tech][n]*z_d[tech][n]*C[tech];

    }
}
con_budget.setDependentVariable(++concount,compgraph::LEQ,false,-1);

//--------------------------------------------------------------------
//EEVP
if(setDesignsManually)
{
    vector<Constraints> con_set_designs(numTech);
    for (int tech=0;tech<numTech;tech++)
    {
        if(!useBattery && tech==2)
        {
            continue;
        }
        con_set_designs[tech] = -1;
        con_set_designs[tech] += z_b[tech][setDesign[tech]];

        con_set_designs[tech].setDependentVariable(++concount,
                                                   compgraph::EQUALITY,
                                                   false,-1);
    }
}
cout<<"-- 1st stage constraints set.. --"<<endl;

//--------------------------------------------------------------------
// Second stage constraints
//  Declaring constraints
//Energy hub
vector<vector<Constraints>> con_energyhub_eb(numTimesteps,
        vector<Constraints>(numScen));
vector<vector<Constraints>> con_renewables_eb(numTimesteps,
        vector<Constraints>(numScen));
vector<vector<Constraints>> con_demand(numTimesteps,
        vector<Constraints>(numScen));


// Renwable power generation
vector<vector<Constraints>> con_PV(numTimesteps,
        vector<Constraints>(numScen));
vector<vector<Constraints>> con_WT(numTimesteps,
        vector<Constraints>(numScen));

// Energy storage: Battery
vector<vector<Constraints>> con_BAT_eb(numTimesteps,
        vector<Constraints>(numScen));
vector<vector<Constraints>> con_BAT_SoC_max(numTimesteps,
        vector<Constraints>(numScen));
vector<vector<Constraints>> con_BAT_ch_max(numTimesteps,
        vector<Constraints>(numScen));
vector<vector<Constraints>> con_BAT_dis_max(numTimesteps,
        vector<Constraints>(numScen));
vector<Constraints> con_BAT_periodicity(numScen);
```

```

// Constraints
for (int h=0;h<numScen;h++)
{
    for (int t=0;t<numTimesteps;t++)
    {
        con_energyhub_eb[t][h] = 0;
        con_energyhub_eb[t][h] -= uncertainParams["L_DEM"][t][h];
        //con_energyhub_eb[t][h] -= f_demand[t][h];
        con_energyhub_eb[t][h] += f_renewables[t][h];
        con_energyhub_eb[t][h] += f_grid_import[t][h];
        // Eb2
        con_renewables_eb[t][h] = 0;
        con_renewables_eb[t][h] -= f_renewables[t][h];
        con_renewables_eb[t][h] -= f_grid_export[t][h];
        con_renewables_eb[t][h] += f_PV[t][h];
        con_renewables_eb[t][h] += f_WT[t][h];
        if(useBattery)
        {
            con_renewables_eb[t][h] -= f_bat_charge[t][h];
            con_renewables_eb[t][h] += f_bat_discharge[t][h];
        }

        // Demand flow --> uncertain parameter
        // con_demand[t][h] = -f_demand[t][h];
        // con_demand[t][h] += uncertainParams["L_DEM"][t][h];
        //PV
        con_PV[t][h] = -f_PV[t][h];
        con_PV[t][h] += eta_PV*uncertainParams["IR_PV"][t][h]
                              *Z[0][h]/1000000;

        // Wind
        con_WT[t][h] = -f_WT[t][h];

        if((uncertainParams["W_WT"][t][h] >= W_min) &&
           (uncertainParams["W_WT"][t][h] <= W_max))
        {
            if (uncertainParams["W_WT"][t][h] <= W_d)
            {

                con_WT[t][h] += Z[1][h]*q_d *
                (pow(uncertainParams["W_WT"][t][h],3) -
                 pow(W_min,3))/(pow(W_d,3)-pow(W_min,3));
            }
            else
            {
                con_WT[t][h] += Z[1][h]*q_d;
            }
        }
        else
        {
            con_WT[t][h] += Z[1][h]*0;
        }

        if(!useBattery)
        {
            continue;
        }
        // Energy balance for battery
        con_BAT_eb[t][h] = -E_BAT[t][h];
        if (t != 0)
        {
            con_BAT_eb[t][h] += E_BAT[t-1][h]*eta_ES_storage;
        }
        else
        {
            con_BAT_eb[t][h] += eta_ES_storage*(SoC_initial-SoC_min)
                                              *Z[2][h]/100;
        }
        con_BAT_eb[t][h] += f_bat_charge[t][h] * eta_ES_ch* deltaT;
        con_BAT_eb[t][h] -= deltaT* f_bat_discharge[t][h]/ eta_ES_dis;

            // State of charge constraints
        con_BAT_SoC_max[t][h] = E_BAT[t][h];
        con_BAT_SoC_max[t][h] -= (SoC_max-SoC_min)*Z[2][h]/100;

            //charging/discharging constraints
        con_BAT_ch_max[t][h] = -deltaT*C_rate_max*Z[2][h];
        con_BAT_dis_max[t][h] = -deltaT*C_rate_max*Z[2][h];
        con_BAT_ch_max[t][h] += f_bat_charge[t][h];
        con_BAT_dis_max[t][h] += f_bat_discharge[t][h];
    }
    if(!useBattery)
    {
        continue;
    }
    // Periodic boundary condition
    con_BAT_periodicity[h] = -E_BAT[numTimesteps-1][h];
    con_BAT_periodicity[h] += (SoC_initial-SoC_min)*Z[2][h]/100;
}

//  Setting constraints
for (int h=0;h<numScen;h++)
{
    for (int t=0;t<numTimesteps;t++)
    {
        // Energy Hub
        con_energyhub_eb[t][h].setDependentVariable(++concount,
                                    compgraph::EQUALITY,true,h+1);
        con_renewables_eb[t][h].setDependentVariable(++concount,
                                    compgraph::EQUALITY,true,h+1);

        //Demand
        //con_demand[t][h].setDependentVariable(++concount,
        //                          compgraph::EQUALITY,true,h+1);

        //PV
        con_PV[t][h].setDependentVariable(++concount,
                                         compgraph::EQUALITY,true,h+1);

        // Wind
        con_WT[t][h].setDependentVariable(++concount,
                                    compgraph::EQUALITY,true,h+1);

        // Battery energy storage
        if(!useBattery)
        {
            continue;
        }
            // Energy balance for battery
        con_BAT_eb[t][h].setDependentVariable(++concount,
                                    compgraph::EQUALITY,true,h+1);

            // State of charge constraints
        con_BAT_SoC_max[t][h].setDependentVariable(++concount,
                                    compgraph::LEQ,true,h+1);
            //Flows
        con_BAT_ch_max[t][h].setDependentVariable(++concount,
                                    compgraph::LEQ,true,h+1);
        con_BAT_dis_max[t][h].setDependentVariable(++concount,
                                    compgraph::LEQ,true,h+1);
    }
    if(!useBattery)
    {
        continue;
    }
    //Periodicity
    con_BAT_periodicity[h].setDependentVariable(++concount,
                                    compgraph::EQUALITY,true,h+1);
}
```

```
cout<<"-- 2nd stage constraints set.. --"<<endl;

cout<<"-- Running program.. --"<<endl;

return numScen;

}
```

C | Scenarios for Program

In this appendix, the scenarios of different realizations of uncertain parameters are presented. The assumed weighted probability of each scenario is given in Table C.1. The realizations for solar irradiance, wind speed, demand load and grid price patterns are given in Tables C.2, C.3, C.4 and C.5 respectively.

Table C.1: Weighted probabilities for scenarios

Scenario	1	2	3	4	5	6
Weight	0.125	0.125	0.125	0.125	0.0625	0.0625
Scenario	7	8	9	10	11	12
Weight	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625

Table C.2: Fluctuating patterns for irradiation [W/m^2]

t	**Scenario**											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	4	11	0	0	5	15	0	0	3	7	0
7	0	25	42	1	0	34	57	1	0	16	27	1
8	5	100	123	25	7	135	166	34	3	65	80	16
9	51	219	243	109	69	296	328	147	33	142	158	71
10	141	357	380	218	190	482	513	294	92	232	247	142
11	211	461	497	312	285	622	671	421	137	300	323	203
12	255	511	569	368	344	690	768	497	166	332	370	239
13	257	520	586	387	347	702	791	522	167	338	381	252
14	230	508	588	354	311	686	794	478	150	330	382	230
15	173	466	558	299	234	629	753	404	112	303	363	194
16	89	388	484	219	120	524	653	296	58	252	315	142
17	24	296	395	136	32	400	533	184	16	192	257	88
18	2	190	281	71	3	257	379	96	1	124	183	46
19	0	92	165	21	0	124	223	28	0	60	107	14
20	0	28	67	2	0	38	90	3	0	18	44	1
21	0	5	20	0	0	7	27	0	0	3	13	0
22	0	0	2	0	0	0	3	0	0	0	1	0
23	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0

Table C.3: Fluctuating patterns for wind speeds $[m/s^2]$

t	**Scenario**											
	1	2	3	4	5	6	7	8	9	10	11	12
1	11.3	14.0	4.0	9.0	7.4	9.1	2.6	5.9	15.3	18.9	5.4	12.2
2	12.2	11.6	5.1	7.7	7.9	7.5	3.3	5.0	16.4	15.6	6.8	10.4
3	10.5	14.9	5.1	9.7	6.8	9.7	3.3	6.3	14.1	20.1	6.8	13.1
4	11.3	14.0	5.1	11.9	7.4	9.1	3.3	7.7	15.3	18.9	6.8	16.0
5	12.2	10.0	6.2	9.7	7.9	6.5	4.0	6.3	16.4	13.5	8.4	13.1
6	11.3	6.5	5.1	9.0	7.4	4.2	3.3	5.9	15.3	8.8	6.8	12.2
7	10.5	8.6	5.6	11.9	6.8	5.6	3.7	7.7	14.1	11.6	7.6	16.0
8	12.2	7.9	7.4	9.7	7.9	5.1	4.8	6.3	16.4	10.6	10.0	13.1
9	10.5	5.9	8.0	9.0	6.8	3.8	5.2	5.9	14.1	7.9	10.8	12.2
10	11.3	7.2	10.0	9.0	7.4	4.7	6.5	5.9	15.3	9.7	13.5	12.2
11	11.3	5.9	10.7	11.9	7.4	3.8	6.9	7.7	15.3	7.9	14.4	16.0
12	8.1	6.5	12.1	9.0	5.3	4.2	7.9	5.9	10.9	8.8	16.3	12.2
13	8.1	5.3	12.8	9.7	5.3	3.4	8.3	6.3	10.9	7.1	17.3	13.1
14	10.5	3.5	12.8	11.9	6.8	2.3	8.3	7.7	14.1	4.8	17.3	16.0
15	10.5	7.2	10.7	11.1	6.8	4.7	6.9	7.2	14.1	9.7	14.4	15.0
16	8.1	5.9	10.0	9.7	5.3	3.8	6.5	6.3	10.9	7.9	13.5	13.1
17	6.6	7.2	10.7	9.7	4.3	4.7	6.9	6.3	8.9	9.7	14.4	13.1
18	9.7	5.9	10.7	7.7	6.3	3.8	6.9	5.0	13.0	7.9	14.4	10.4
19	11.3	6.5	5.1	6.5	7.4	4.2	3.3	4.2	15.3	8.8	6.8	8.7
20	10.5	10.8	4.0	6.5	6.8	7.0	2.6	4.2	14.1	14.6	5.4	8.7
21	11.3	13.2	4.0	5.3	7.4	8.6	2.6	3.4	15.3	17.8	5.4	7.1
22	11.3	13.2	5.6	5.3	7.4	8.6	3.7	3.4	15.3	17.8	7.6	7.1
23	10.5	14.0	4.5	5.3	6.8	9.1	2.9	3.4	14.1	18.9	6.1	7.1
24	12.2	13.2	4.0	7.7	7.9	8.6	2.6	5.0	16.4	17.8	5.4	10.4

Table C.4: Fluctuating patterns for demand load $[MW]$

t	**Scenario**											
	1	2	3	4	5	6	7	8	9	10	11	12
1	3.93	3.07	2.53	2.97	3.93	3.07	2.53	2.97	3.93	3.07	2.53	2.97
2	3.86	3.00	2.49	2.89	3.86	3.00	2.49	2.89	3.86	3.00	2.49	2.89
3	3.78	2.93	2.45	2.83	3.78	2.93	2.45	2.83	3.78	2.93	2.45	2.83
4	3.70	2.91	2.48	2.86	3.70	2.91	2.48	2.86	3.70	2.91	2.48	2.86
5	3.58	3.17	2.81	3.16	3.58	3.17	2.81	3.16	3.58	3.17	2.81	3.16
6	3.61	3.66	3.32	3.68	3.61	3.66	3.32	3.68	3.61	3.66	3.32	3.68
7	3.96	4.00	3.68	4.04	3.96	4.00	3.68	4.04	3.96	4.00	3.68	4.04
8	4.41	4.19	3.89	4.22	4.41	4.19	3.89	4.22	4.41	4.19	3.89	4.22
9	4.68	4.28	3.97	4.33	4.68	4.28	3.97	4.33	4.68	4.28	3.97	4.33
10	4.93	4.31	4.02	4.37	4.93	4.31	4.02	4.37	4.93	4.31	4.02	4.37
11	5.04	4.31	4.03	4.39	5.04	4.31	4.03	4.39	5.04	4.31	4.03	4.39
12	5.09	4.25	3.97	4.34	5.09	4.25	3.97	4.34	5.09	4.25	3.97	4.34
13	5.11	4.19	3.91	4.29	5.11	4.19	3.91	4.29	5.11	4.19	3.91	4.29
14	5.06	4.14	3.88	4.26	5.06	4.14	3.88	4.26	5.06	4.14	3.88	4.26
15	5.03	4.18	3.94	4.35	5.03	4.18	3.94	4.35	5.03	4.18	3.94	4.35
16	5.04	4.26	3.97	4.57	5.04	4.26	3.97	4.57	5.04	4.26	3.97	4.57
17	5.31	4.25	3.85	4.72	5.31	4.25	3.85	4.72	5.31	4.25	3.85	4.72
18	5.58	4.29	3.72	4.68	5.58	4.29	3.72	4.68	5.58	4.29	3.72	4.68
19	5.52	4.29	3.63	4.47	5.52	4.29	3.63	4.47	5.52	4.29	3.63	4.47
20	5.29	4.15	3.60	4.18	5.29	4.15	3.60	4.18	5.29	4.15	3.60	4.18
21	5.01	3.82	3.42	3.80	5.01	3.82	3.42	3.80	5.01	3.82	3.42	3.80
22	4.66	3.41	3.03	3.40	4.66	3.41	3.03	3.40	4.66	3.41	3.03	3.40
23	4.29	3.09	2.74	3.09	4.29	3.09	2.74	3.09	4.29	3.09	2.74	3.09
24	3.94	3.10	2.62	3.05	3.94	3.10	2.62	3.05	3.94	3.10	2.62	3.05

Table C.5: Fluctuating patterns for cost of grid import [$\$/MWh$]

t	**Scenario**											
	1	2	3	4	5	6	7	8	9	10	11	12
1	76	62	61	68	76	62	61	68	76	62	61	68
2	74	60	60	66	74	60	60	66	74	60	60	66
3	72	58	58	64	72	58	58	64	72	58	58	64
4	70	57	57	63	70	57	57	63	70	57	57	63
5	71	58	58	64	71	58	58	64	71	58	58	64
6	75	61	61	68	75	61	61	68	75	61	61	68
7	87	75	69	78	87	75	69	78	87	75	69	78
8	101	86	79	89	101	86	79	89	101	86	79	89
9	104	87	82	92	104	87	82	92	104	87	82	92
10	100	84	79	89	100	84	79	89	100	84	79	89
11	96	81	76	86	96	81	76	86	96	81	76	86
12	95	81	75	85	95	81	75	85	95	81	75	85
13	91	78	73	81	91	78	73	81	91	78	73	81
14	89	77	72	80	89	77	72	80	89	77	72	80
15	88	77	71	80	88	77	71	80	88	77	71	80
16	89	78	72	80	89	78	72	80	89	78	72	80
17	91	79	74	82	91	79	74	82	91	79	74	82
18	99	85	79	88	99	85	79	88	99	85	79	88
19	105	89	82	93	105	89	82	93	105	89	82	93
20	106	90	82	93	106	90	82	93	106	90	82	93
21	99	85	77	87	99	85	77	87	99	85	77	87
22	92	79	73	81	92	79	73	81	92	79	73	81
23	87	75	70	78	87	75	70	78	87	75	70	78
24	79	68	64	71	79	68	64	71	79	68	64	71

D | Program Output

This appendix presents the GOSSIP output of the stochastic program when running the program with two different algorithms: NGBD and ANTIGONE´s full space. It is noted that the second stage flow and state variables are omitted.

Output with NGBD

```
Discrete Sets of technologies:
Tech 1:
0 8000 16000 24000 32000 40000 48000 56000 64000 72000
Tech 2:
0 2 4 6 8 10 12 14 16 18
Tech 3:
0 1.5 3 4.5 6 7.5 9 10.5 12 13.5

-- Discretization performed.. --
-- 2nd stage varaiables declared.. --
-- Objective function set.. --
-- 1st stage constraints set.. --
-- 2nd stage constraints set.. --
-- Running program.. --

               Number of scenarios: 12

          Before Preprocessing:

                         Variables: 2370

                          Complicating: 30
                                    Binary: 30
                                   Integer: 0
                                Continuous: 0

                              Recourse: 2340
                                    Binary: 0
                                   Integer: 0
                                Continuous: 2340

                       Constraints: 2356

                           First-stage: 4
                              Recourse: 2352

          After Preprocessing:
```

```
                         Variables: 2382

                          Complicating: 30
                                    Binary: 30
                                Continuous: 0

                              Recourse: 2352
                                    Binary: 0
                                   Integer: 0
                                Continuous: 2352

                       Constraints: 2368

                           First-stage: 4
                                  Equality: 3
                                Inequality: 1

                              Recourse: 2364
                                  Equality: 1200
                                Inequality: 1164

                    Linear Constraints: 2368
                 Nonlinear Constraints: 0



Time for preprocessing: 0.06 seconds
         Reformulation: 0.00 seconds
        Simplification: 0.06 seconds
   Structure detection: 0.00 seconds

Cannot write full-space GAMS model when algorithm is NGBD

Running NGBD to solve the input problem


|It#|      LBD    |   UBDPB    |    UBD     | Gap % |Solver time|Wall Clock time|
|  0|  -6.0444e+06|  2.1035e+06|            |  ---  |       0.01|           1.86|
|  1|  -6.9719e+05|  3.0609e+05|            |  ---  |       0.02|           1.87|
|  2|  -6.6780e+05|  1.2018e+03|            |  ---  |       0.02|           1.88|
|  3|  -3.2074e+05| -7.2867e+04|            |  ---  |       0.03|           1.90|
|  4|  -2.3392e+05| -1.0644e+05|            |  ---  |       0.05|           1.92|
|  5|  -1.6706e+05| -1.3954e+05|            |  ---  |       0.06|           1.94|
|  6|  -1.5679e+05|            |            |  ---  |       0.08|           1.96|
|  7|  -1.3657e+05|            |            |  ---  |       0.10|           1.99|
|   |  -1.3657e+05|            | -1.3954e+05|0.00000|       0.11|           2.00|
|   |  -1.3657e+05| -1.1908e+05|            |0.00000|       0.11|           2.00|

Number of iterations: 8
Final LBD: -1.3657e+05
Final UBD: -1.3954e+05
Final UBDPB: -1.1908e+05
Final Relative Gap: 0.0000e+00

PP Calls: 1
```

```
PBP Calls: 8
FP Calls: 0
RMP Calls: 8
FBBT Calls: 0
OBBT Calls: 0

Model set up + generation time: 1.85

Total Solver Time: 0.11
PP Solver Time: 0.01
PBP Solver Time: 0.05
FP Solver Time: 0.00
RMP Solver Time: 0.05
IG Solver Time: 0.00
FBBT Solver Time: 0.00
OBBT Solver Time: 0.00

Total Wall Time: 2.00
PP Wall Time: 0.02
PBP Wall Time: 0.09
FP Wall Time: 0.00
RMP Wall Time: 0.05
IG Wall Time: 0.00
FBBT Wall Time: 0.00
OBBT Wall Time: 0.00

Best found solution:
z_b[1][1] = 0.0000
z_b[1][2] = 0.0000
z_b[1][3] = 0.0000
z_b[1][4] = 0.0000
z_b[1][5] = 0.0000
z_b[1][6] = 0.0000
z_b[1][7] = 1.0000
z_b[1][8] = 0.0000
z_b[1][9] = 0.0000
z_b[1][10] = 0.0000
z_b[2][1] = 0.0000
z_b[2][2] = 0.0000
z_b[2][3] = 0.0000
z_b[2][4] = 0.0000
z_b[2][5] = 0.0000
z_b[2][6] = 1.0000
z_b[2][7] = 0.0000
z_b[2][8] = 0.0000
z_b[2][9] = 0.0000
z_b[2][10] = 0.0000
z_b[3][1] = 0.0000
z_b[3][2] = 0.0000
z_b[3][3] = 0.0000
z_b[3][4] = 0.0000
z_b[3][5] = 0.0000
z_b[3][6] = 0.0000
z_b[3][7] = 1.0000
z_b[3][8] = 0.0000
```

```
z_b[3][9] = 0.0000
z_b[3][10] = 0.0000

Z[0][0] = 48000.0000
Z[0][1] = 48000.0000
Z[0][2] = 48000.0000
Z[0][3] = 48000.0000
Z[0][4] = 48000.0000
Z[0][5] = 48000.0000
Z[0][6] = 48000.0000
Z[0][7] = 48000.0000
Z[0][8] = 48000.0000
Z[0][9] = 48000.0000
Z[0][10] = 48000.0000
Z[0][11] = 48000.0000
Z[1][0] = 10.0000
Z[1][1] = 10.0000
Z[1][2] = 10.0000
Z[1][3] = 10.0000
Z[1][4] = 10.0000
Z[1][5] = 10.0000
Z[1][6] = 10.0000
Z[1][7] = 10.0000
Z[1][8] = 10.0000
Z[1][9] = 10.0000
Z[1][10] = 10.0000
Z[1][11] = 10.0000
Z[2][0] = 9.0000
Z[2][1] = 9.0000
Z[2][2] = 9.0000
Z[2][3] = 9.0000
Z[2][4] = 9.0000
Z[2][5] = 9.0000
Z[2][6] = 9.0000
Z[2][7] = 9.0000
Z[2][8] = 9.0000
Z[2][9] = 9.0000
Z[2][10] = 9.0000
Z[2][11] = 9.0000

Final wall clock time = 2.03
```

Output with ANTIGONE´s full space

```
Discrete Sets of technologies:
Tech 1:
0 8000 16000 24000 32000 40000 48000 56000 64000 72000
Tech 2:
0 2 4 6 8 10 12 14 16 18
Tech 3:
0 1.5 3 4.5 6 7.5 9 10.5 12 13.5

-- Discretization performed.. --
-- 2nd stage varaiables declared.. --
-- Objective function set.. --
-- 1st stage constraints set.. --
-- 2nd stage constraints set.. --
-- Running program.. --

               Number of scenarios: 12

          Before Preprocessing:

                         Variables: 2370

                          Complicating: 30
                                    Binary: 30
                                   Integer: 0
                                Continuous: 0

                              Recourse: 2340
                                    Binary: 0
                                   Integer: 0
                                Continuous: 2340

                       Constraints: 2356

                           First-stage: 4
                              Recourse: 2352

          After Preprocessing:

                         Variables: 2382

                          Complicating: 30
                                    Binary: 30
                                Continuous: 0

                              Recourse: 2352
                                    Binary: 0
                                   Integer: 0
                                Continuous: 2352

                       Constraints: 2368

                           First-stage: 4
                                  Equality: 3
                                Inequality: 1

                              Recourse: 2364
                                  Equality: 1200
                                Inequality: 1164



Solving the full space problem

Tried aggregator 2 times.
MIP Presolve eliminated 448 rows and 447 columns.
MIP Presolve modified 3 coefficients.
Aggregator did 289 substitutions.
```

```
Reduced MIP has 1631 rows, 1646 columns, and 5176 nonzeros.
Reduced MIP has 29 binaries, 0 generals, 0 SOSs, and 0 indicators.
Presolve time = 0.01 sec. (5.55 ticks)
Found incumbent of value 2771209.575000 after 0.01 sec. (7.21 ticks)
Probing time = 0.01 sec. (2.00 ticks)
Tried aggregator 1 time.
Reduced MIP has 1631 rows, 1646 columns, and 5176 nonzeros.
Reduced MIP has 29 binaries, 24 generals, 0 SOSs, and 0 indicators.
Presolve time = 0.00 sec. (1.91 ticks)
Probing time = 0.00 sec. (1.99 ticks)
Clique table members: 32.
MIP emphasis: balance optimality and feasibility.
MIP search method: dynamic search.
Parallel mode: none, using 1 thread.
Root relaxation solution time = 0.04 sec. (21.20 ticks)

        Nodes                                         Cuts/
   Node  Left   Objective  IInf   Best Integer   Best Bound     ItCnt   Gap

*     0+    0                    2771209.58   -5654588.52    983  304.05%
      0     0  -154033.10    30   2771209.58    -154033.10    983  105.56%
*     0+    0                     403352.71    -154033.10    983  138.19%
      0     0  -153950.62    32    403352.71      Cuts: 42    997  138.17%
*     0+    0                    -127763.39    -153950.62    997   20.50%
      0     0  -153763.49    31   -127763.39       Cuts: 5   1026   20.35%
      0     0  -153709.51    31   -127763.39       Cuts: 3   1037   20.31%
      0     0  -153709.51    32   -127763.39       Cuts: 2   1040   20.31%
      0     0  -153706.91    32   -127763.39      Cuts: 11   1043   20.31%
      0     2  -153706.91    32   -127763.39    -153659.25   1043   20.27%
Elapsed time = 0.17 sec. (110.10 ticks, tree = 0.00 MB, solutions = 3)
*     9     5      integral     0  -139541.1204  -147136.2604     1219    5.44%

GUB cover cuts applied:  2
Clique cuts applied:  1
Implied bound cuts applied:  23
Mixed integer rounding cuts applied:  4
Zero-half cuts applied:  1
Gomory fractional cuts applied:  2

Root node processing (before b&c):
  Real time             =    0.16 sec. (98.70 ticks)
Sequential b&c:
  Real time             =    0.01 sec. (7.48 ticks)
                          ------------
Total (root+branch&cut) =    0.16 sec. (106.18 ticks)
Best found objective value = -139541
Best found solution:
z_b[1][1] = -0.0000
z_b[1][2] = -0.0000
z_b[1][3] = -0.0000
z_b[1][4] = 0.0000
z_b[1][5] = -0.0000
z_b[1][6] = 0.0000
z_b[1][7] = 1.0000
z_b[1][8] = 0.0000
```

```
z_b[1][9] = 0.0000
z_b[1][10] = 0.0000
z_b[2][1] = -0.0000
z_b[2][2] = -0.0000
z_b[2][3] = -0.0000
z_b[2][4] = -0.0000
z_b[2][5] = -0.0000
z_b[2][6] = 1.0000
z_b[2][7] = 0.0000
z_b[2][8] = 0.0000
z_b[2][9] = 0.0000
z_b[2][10] = -0.0000
z_b[3][1] = -0.0000
z_b[3][2] = 0.0000
z_b[3][3] = -0.0000
z_b[3][4] = -0.0000
z_b[3][5] = -0.0000
z_b[3][6] = -0.0000
z_b[3][7] = 1.0000
z_b[3][8] = 0.0000
z_b[3][9] = 0.0000
z_b[3][10] = 0.0000

Z[0][0] = 48000.0000
Z[0][1] = 48000.0000
Z[0][2] = 48000.0000
Z[0][3] = 48000.0000
Z[0][4] = 48000.0000
Z[0][5] = 48000.0000
Z[0][6] = 48000.0000
Z[0][7] = 48000.0000
Z[0][8] = 48000.0000
Z[0][9] = 48000.0000
Z[0][10] = 48000.0000
Z[0][11] = 48000.0000
Z[1][0] = 10.0000
Z[1][1] = 10.0000
Z[1][2] = 10.0000
Z[1][3] = 10.0000
Z[1][4] = 10.0000
Z[1][5] = 10.0000
Z[1][6] = 10.0000
Z[1][7] = 10.0000
Z[1][8] = 10.0000
Z[1][9] = 10.0000
Z[1][10] = 10.0000
Z[1][11] = 10.0000
Z[2][0] = 9.0000
Z[2][1] = 9.0000
Z[2][2] = 9.0000
Z[2][3] = 9.0000
Z[2][4] = 9.0000
Z[2][5] = 9.0000
Z[2][6] = 9.0000
Z[2][7] = 9.0000
```

```
Z[2][8] = 9.0000
Z[2][9] = 9.0000
Z[2][10] = 9.0000
Z[2][11] = 9.0000
Final wall clock time = 2.53
```