PhD-FSTM-2021-030
The Faculty of Sciences, Technology and Medicine

DISSERTATION

Defence held on 26/05/2021 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN *INFORMATIQUE*

by

Nikolaos ANTONIADIS
Born on 7 July 1971 in Nea Filadelfeia Attikis, (Greece)

ENHANCING SMART GRID RESILIENCE AND RELIABILITY BY USING AND COMBINING SIMULATION AND OPTIMIZATION METHODS

Dissertation defence committee

Dr Yves Le Traon, dissertation supervisor
Professor, Université du Luxembourg

Dr Nicolas Navet, Chairman
Professor, Université du Luxembourg

Dr Olivier Barais, Vice Chairman
Professor, Université de Rennes 1, France

Dr Sylvain Kubler
Associate Professor, Université de Lorraine, France

Dr Maxime Cordy
Research Scientist, Université du Luxembourg

To my wife Kiki and my parents in law Antonis and Vagia, for their endless love, support, and encouragement.

πάντα κατ᾽ ἀριθμόν γίγνονται.
[everything is number]

Pythagoras 570–495 BC

Σοφώτατον χρόνος· ἀνευρίσκει γὰρ πάντα.
[Time is the wisest of all things that are;
for it brings everything to light.]

Thales 624–545 BC

I have no special talents. I am only passionately curious.

Albert Einstein

Nothing is better than reading and gaining more and more knowledge.

Stephen Hawking

Abstract

Modern electrical grids include numerous digital technologies for producing, transmitting, distributing, and supplying electricity. The electrical grids that achieve the most reliable, efficient, and less environmental impact operation using the above technologies combined with renewable energy sources are characterized as smart grids. The study of electricity networks aims at the continuous and uninterrupted production, transmission, and distribution of electricity under the safest operating conditions. Therefore, an electrical grid is designed and studied in a multifaceted way to highlight the weaknesses and reduce possible disturbances. One of the most critical disturbances that can occur in energy grids is overload. Overloads on an electrical system are dangerous, as they can cause overheating or an electric arc. Cables in an electrical grid have a maximum ampacity, *i.e.*, current capacity, that can safely flow. If an excessive number of devices, such as electric vehicles, are connected to a circuit, the electrical current will overheat the cables. If the cable insulation melts, an electric arc can be generated and cause a fire in the overheating area, even inside a wall. In order to avoid overloads, fuses are installed in the circuits. If the current exceeds a specific value, the fuse is activated, drops, and opens the circuit, thus interrupting electricity flow. However, even if they are below the safety limits, sustained overloads could also damage the wires.

Smart grid operators could change the state of each grid's fuse or could remotely curtail the over-producing/over-consuming users so that, with the minimum interruption, any potential overload could be prevented. Nevertheless, making the most appropriate decisions is a complicated decision-making task, mainly due to contractual and technical obligations.

The present dissertation studies the overloading prevention problem in terms of smart grids' reliability and resilience and evaluates real-world topology in a

Luxembourg city district. To this end, it suggests solution methods that can suggest optimal countermeasures to operators facing potential overloading incidents. Specifically, the dissertation has three main axes:

The first axis regards the deterministic overloading prevention problem. Given the topology and the energy data of a microgrid at the current time, the potential overloading incidents are detected, and the optimal countermeasures are calculated for the next measurement interval. The grid operators can apply the proposed actions to recover the grid from the disturbance. Into the thesis, the problem is defined and formulated as a Multiobjective Mixed Integer Quadratically Constrained Program. The dissertation also suggests a solution method using a combinatorial optimization approach with a state-of-the-art exact solver.

The second axis focuses on reliability analysis through simulation after a potential overloading incident. Smart grid operators would be of great use to ensure stability after a potential overload for a planning horizon, as the future electrical values are unknown. To evaluate the robustness of the topology reconfiguration after a disturbance, like an overload, reliability analysis through simulation is employed.

The third axis proposes the single-stage stochastic overloading prevention problem. It differs from the deterministic problem as the optimal countermeasures are calculated for a measurement horizon, *e.g.*, 24 h. The dissertation defines the corresponding single-stage stochastic program and proposes a simheuristic method to solve it.

Overall, this thesis presents a fully-edge study on reliability optimization for smart grids to provide the appropriate countermeasures after a potential overloading disturbance. The present approach has been developed in collaboration with an industrial partner and evaluated on real-world topology.

Keywords: Smart grids, Reliability, Resilience, Combinatorial Optimization, Integer Linear Programming, Simheuristic, Variable Neighborhood Search, Stochastic Optimization, Simulation

Acknowledgments

The Ph.D. experience goes beyond research, experimentations, and paper writing. It is indeed a challenging life experience that started in January 2017, and which outcome owes much to many people's support and help.

First of all, I would like to thank Prof. Dr. Yves Le Traon, who allowed me to pursue my Ph.D. studies in his group and under his supervision, for his consistent support, encouragement, and guidance throughout the past 4 years. I would also like to thank Dr. Maxime Cordy and Prof. Dr. Angelo Sifaleras, as they both helped to develop the presented contributions. I am also grateful to the chairman of my defense committee Prof. Dr. Nicolas Navet for his valuable suggestions, and his ingenious advice all these years. Additionally, I very much appreciate Prof. Dr. Olivier Barais, and Prof. Dr. Sylvain Kubler for kindly accepting to participate in my dissertation defense committee.

My special thanks go to my dear friend, Dr. Anestis Tsakmalis, for his advice and the plenty of passionate talks. I am pleased about the friendship we have built up during the years.

My sincere thanks go to Yves Reckinger and Robert Graglia from Creos for the many fruitful discussions and the time they collaborate with us. It was beneficial and rewarding for me to apply my research to a concrete industrial case.

I would also like to express my warm thanks to all current and former SerVal team members for the plenty of good coffee breaks, discussions, and the great times we had. I wish all of them the very best.

Finally, and more personally, I would like to express my heartfelt thanks to my family and my friends for their continuous and unconditional support during these last years.

The role they all played in this journey is more important than they can know.

Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms

GVNS	General Variable Neighborhood Search xv, 43
HDS	Hybrid Data Space xv, 91–94
HSA	Hybrid Simulation–Analytic xv, 35–37
IRP	Inventory Routing Problem xv, 38
MCC	Matthews Correlation Coefficient xv, 99
MCS	Monte Carlo Simulation v, xv, 10, 23–26, 36–38, 67, 68, 72, 74, 75, 86, 106
MIQCP	Mixed Integer Quadratically Constraint Program xv, 56, 58
MO-MIQCP	Multiobjective Mixed Integer Quadratically Constraint Program xv, 9, 10, 13, 51, 64, 106
NDDS	Non-ordered Discrete Data Space xv, 91, 92
NSMCS	Non-Sequential Monte Carlo Simulation xv, 24
OPP	Overloading Prevention Problem vii, xv, 9, 10, 13, 34, 55, 64, 106, 128, 129, 131, 133, 135, 137, 139, 141
PFSP	Permutation Flow-Shop Problem xv, 38
PLC	Power-Line Communication xv, 88, 90
PV	Photovoltaic xv, 96
RMSE	Root Mean Squared Error xv, 97
RVNS	Reduced Variable Neighborhood Search xv, 44, 74, 75
SAIDI	System Average Interruption Duration Index xv, 24, 69
SAIFI	System Average Interruption Frequency Index xv, 24, 69, 70
SAODI	System Average Overload Duration Index xv, 69–71
SAOFI	System Average Overload Frequency Index xv, 69, 71
SDP	Stochastic Dynamic Programming xv, 32
SE	Solution Evaluation xv, 34, 35
SG	Solution Generation xv, 35, 37
SMC	Surrogate Model Construction xv, 34–36
SMCS	Sequential Monte Carlo Simulation xv, 24
SO	Simulation-Optimization xv, 35, 37
SSSP	Single-State Stochastic Program xv, 10, 11, 13, 65, 67, 106

SVN	Singular Value Decomposition xv, 58
SVNS	Skewed Variable Neighborhood Search xv, 44, 45
TSP	Traveling Salesman Problem xv, 41
UOI	Unified Overload Index xv, 67, 68, 71–73, 75, 79, 81, 82, 86
VND	Variable Neighborhood Descent xv, 42, 43
VNS	Variable Neighborhood Search xv, 13, 40, 41, 44, 67, 74
VRP	Vehicle Routing Problem xv, 38, 44

Nomenclature

Indices

b	cabinet index, $b \in \{1, \dots, o\}$
f	fuse index, $f \in \{1, \dots, 2n\}$
i	cable index, $i \in \{1, \dots, n\}$
j	linear equation index, $j \in \{1, \dots, leq\}$
k	user index, $k \in \{1, \dots, m\}$

Parameters

δ	measurement frequency coefficient; e.g. $\frac{60}{15} = 4$, for 15 min interval
λ	maximum allowed current load percentage for all cables, e.g. 80%
aE_k	active energy for user k, $aE_k = aEC_k - aEP_k$, $aE_k \in \mathbb{R}$
aEC_k	active energy consumption for user k, $aEC_k \in \mathbb{R}_+$
aEP_k	active energy production for user k, $aEP_k \in \mathbb{R}_+$
$cblmo$	cable minutes of overload
$cblo$	number of cables overloaded
cc_{bf}	fuse cabinet indicator; 1 if fuse f belongs to the cabinet b, 0 otherwise
cl_i	maximum allowed current load in cable i, e.g. 100A
cmo	customer minutes of overload

co number of customers overloaded

cur_k amperage of user k, $cur_k = \frac{\sqrt{aE_k^2+rE_k^2}}{\sqrt{3}\cdot 230}$

$cust_i$ number of customers at cable i

$downt_{it}$ duration of overload at cable i on period t

$fail_{it}$ new overload at cable i on period t

I_R curtailed amperage for users, e.g. 20A

I_{LC} maximum allowed amperage for consumers, e.g. 32A

I_{LP} maximum allowed amperage for producers, e.g. 60A

leq number of linear equations, $leq \in \mathbb{N}^*$

lmo overload duration of customers

lo load of customers overloaded

lt total load

m number of users, $m \in \mathbb{N}^*$

n number of cables, $n \in \mathbb{N}^*$

o number of cabinets (including substations), $o \in \mathbb{N}^*$

Pl_i initial active energy for cable i, $Pl_i = \delta \sum_{k=1}^{m} uc_{ki} RaE_k$

Ql_i initial reactive energy for cable i, $Ql_i = \delta \sum_{k=1}^{m} uc_{ki} rE_k$

RaE_k real active energy consumption for user k,
$RaE_k = aE_k$, if $cur_k < I_{LC}$, (consumer) or $cur_k < I_{LP}$ (producer), and $RaE_k = RGaE_k$ otherwise

rE_k reactive energy for user k, $rE_k = rEC_k - rEP_k$, $rE_k \in \mathbb{R}$

rEC_k reactive energy consumption for user k, $rEC_k \in \mathbb{R}_+$

rEP_k reactive energy production for user k, $rEP_k \in \mathbb{R}_+$

$RGaE_k$ curtailed active energy for user k,
$RGaE_k = \sqrt{|230^2 \cdot 3 \cdot I_R^2 - rE_k^2|}$, $RGaE_k \in \mathbb{R}_+$

T time horizon; e.g. $4 \cdot 24 = 96$, for 24 hours and 15 min measurement interval

t time period

u_{it} overload indicator at cable i on period t

uc_{ki} user cable indicator; 1 if user k is connected with cable i, 0 otherwise

x_f^0 initial fuse state; 1 if fuse f is closed, and 0 otherwise;
if $f = 2i$, x_f^0 denotes the initial state of the *start* fuse of cable i,
else if $f = 2i + 1$, x_f^0 denotes the initial state of the *end* fuse of cable i

Variables

A_{jf} coefficient matrix element; for equation j and fuse f, $A_{jf} \in \{-1, 0, 1\}$

$dfcab_b$ cabinet visit indicator; 1 if $\sum_{f=1}^{2n} cc_{bf}|x_f - x_f^0| \geq 1$, 0 otherwise

l_i actual current load percentage, at cable i;
$l_i = \max(\frac{100\sqrt{wp_{2i}^2+wq_{2i}^2}}{230cl_i\sqrt{3}}, \frac{100\sqrt{wp_{2i+1}^2+wq_{2i+1}^2}}{230cl_i\sqrt{3}})$

P_j active load vector element; $P_j = Pl_i \cdot r_i$, if equation j is describing the current flow of cable i, and 0 otherwise, $P_j \in \mathbb{R}$

Q_j reactive load vector element; $Q_j = Ql_i \cdot r_i$, if equation j is describing the current flow of cable i, and 0 otherwise, $Q_j \in \mathbb{R}$

r_i reachability cable state; 1 if cable i is powered and 0 otherwise

wp_f actual active energy vector energy element for fuse f; $wp_f \in \mathbb{R}$

wq_f actual reactive energy vector energy element for fuse f; $wq_f \in \mathbb{R}$

x_f fuse state; 1 if fuse f is closed, and 0 otherwise;
if $f = 2i$, x_f denotes the current state of the *start* fuse of cable i,
else if $f = 2i + 1$, x_f denotes the current state of the *end* fuse of cable i

Part I

Prologue

1 Introduction

ἀρχὴ ἥμισυ παντὸς
[The beginning is half of the whole.]

(Plato, The Republic)

This chapter starts by exposing the context of this dissertation. Then, the aim and the objectives of this thesis are described. Finally, an overview of the contributions and thesis' structure of this dissertation is presented.

Contents

1.1 Context

The study of the modern electrical grids, also known as *smart grids*, aims at the continuous and uninterrupted production, transmission, and distribution of electricity under safe operation's optimal economic conditions. An electrical grid, to achieve the above goals, is designed and studied in many ways to identify its weaknesses and reduce the chances of possible disruptions. The study of electrical networks includes, among others, three significant chapters: the study of load flow, the study of short circuits, and the study of transient stability.

The transient stability study seeks to provide an answer to the following two key questions [142]:

- Does the system return to a state of equilibrium after a disturbance?
- How fast the switching procedures must work to isolate the part of the system in error status in time?

For the study of transient stability, several parameters are considered, such as the switching functions' activation time, the generator rotor's electromechanical oscillations parameters, the initial system conditions before the disturbance, and the differential equations describing modern and induction machines during the transitional period. The study of short circuits concerns the design of protection systems and the identification of each switch of the system's necessary functions so that with the least possible interruption of the services provided for consumption, the wrong parts of the system are disconnected, and thus their destruction is avoided [142]. Both the studies of short circuits and transient stability deal with an electrical system's behavior in the field of time, where the operation of the switching functions is defined.

Also, in an energy system, various disturbances can occur at any time, such as power bus errors, disconnection of generators, or sudden load changes, which can cause significant changes in the system's characteristics. The result of these changes is the redistribution of power flow and the prevalence of new voltage and current values resulting in a state of transient stability, with new values of currents and powers for the system, or a state of transient instability and collapse of the system [142].

One of the most critical issues in power grids [50] is overloading cables as they can harm distribution power lines. When an overload trips the fuse switch, the circuit opens and, consequently, flow and heating stop. Grid operators usually assume that the load rate above a predefined level on a cable entails a significant overload chance. Long-term overloads, however, can also damage cables even within the security limits and may cause energy grids to malfunction [148].

Specific counteractions can then be applied to reduce cable loads and, consequently, to prevent overload. The preferred solution consists of limiting the overproduction remotely (e.g., solar panels on a sunny day) or over-consumption of specific users (e.g., charging EVs); this countermeasure is commonly named *load curtailment* [131]. However, some users have contracts that prevent the operator from regulating their power capacity. Therefore, curtailment is not, in such cases, an option. More generally, if curtailments cannot result in a stable state, i.e., without risk of overloading, the operators have to reconfigure the topology of the grid by switching fuses, using the *intertrip* [20] method to shift reserves from one network to another, even if intertrip is complicated for the meshed low-voltage network [20]. Changing fuse states require technicians to visit the corresponding cabinets physically. Therefore, minimizing the number of visiting cabinets is an object of considerable solicitude to the grid operator to minimize a potential incident's restoration time. Another concern is the minimization of the number of fuses that have to be switched on or off, as the grid's configuration should remain nearly the same as its initial state, returning to the equilibrium state. Avoid disconnecting users, especially critical ones, such as patients, is a matter of great concern to the grid operators. Still, this may happen as a last resort to prevent cascading overloads [20] and to avoid any damage to the power line when there is an insufficient operating reserve. In this case, the number of disconnected users should remain minimal. The above countermeasures, including user curtailments and reconfiguration of the grid's topology, are generally applied in a short time, i.e., less than an hour. Even if the above counteractions are applied immediately after detecting a hazardous overloading incident, the recovery response solution could lead to another overloading incident, as the future demands are not known beforehand.

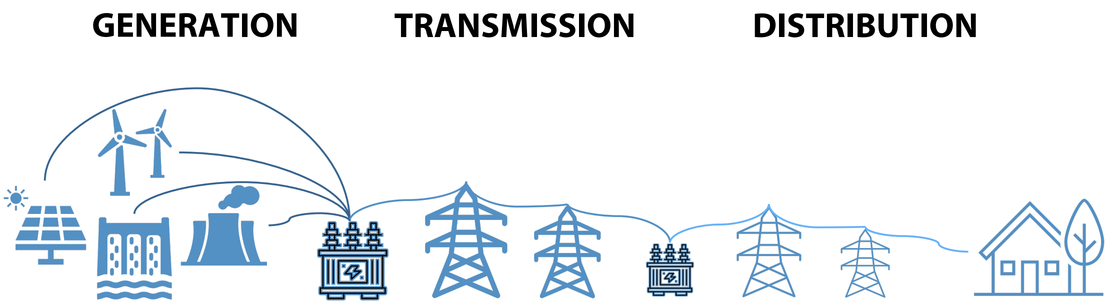


Figure 1.1: Diagram of an electric power system.

There are two types of networks in a power grid, depending on the electric power voltage that circulates, the *transmission network* and the *distribution network*. The transmission network transfers the electrical power generated at the traditional and renewable-energy power plants to the transmission substations, where transformers convert the low voltage of electricity into high. The transmission is done at high voltage to reduce power losses when the distances are long. Transmission lines can not directly supply consumers using low voltage but reach specific points, the transmission substations, where the voltage is reduced to medium voltage. Substations are nodes in the electricity network. From these points where the transmission substations are located, the distribution lines start, ending at the distribution substations where the medium voltage is lowered to low voltage, as shown in Figure 1.1. The distribution network includes the medium voltage distribution network that transfers electricity from the transmission substations to the distribution substations and to industries, and the low voltage distribution network [42] that transmits electricity from distribution substations to non-industrial customers. This low-voltage network is, in general, more complex and meshed than the medium-voltage one, and it is harder to track its disturbances. In Figure 1.2, an example of such a network is shown in the form of a multigraph, where its vertices are the substations and cabinets, the electrical enclosures which connect the distribution cables, and its edges are the power lines of the grid.

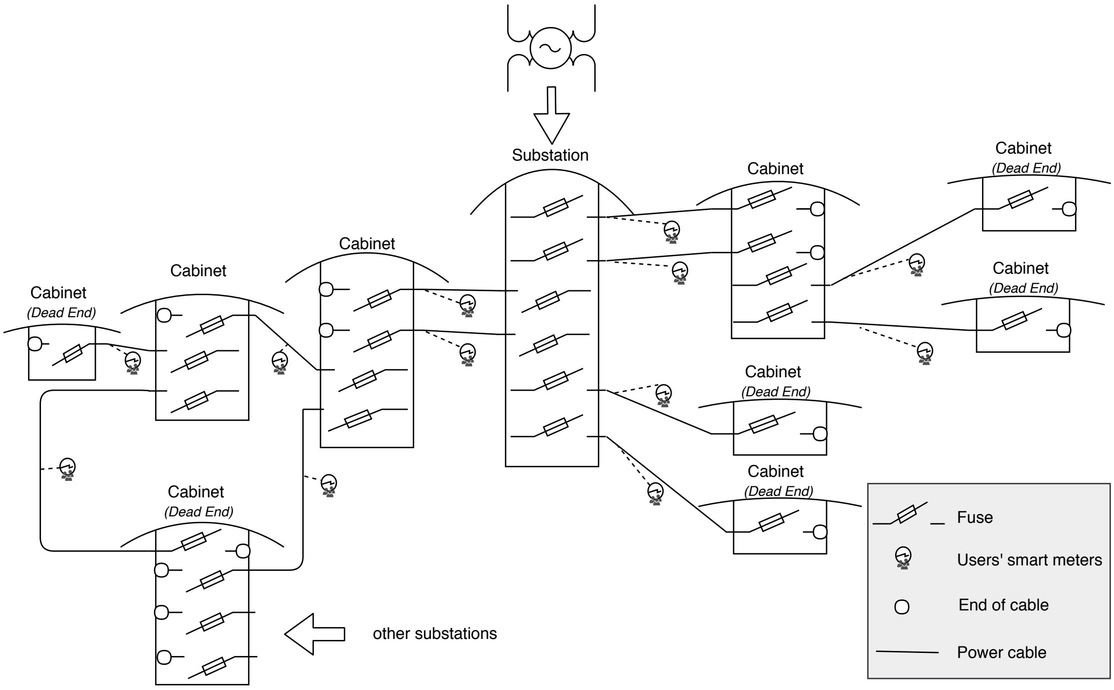


Figure 1.2: A low-voltage distribution network example.

Each distribution substation comprises a power subsystem that delivers electric energy to industrial and residential users through feeder pillars, i.e., cabinets and cables. Distribution cabinets control the distributed power and provide overload protection to the network lines through their fuses. Between the service cable and each user installation, a smart meter is installed to measure the electric consumption and manage loads through its relay trigger. The number of connected components of the multigraph mentioned above is equal to the number of distribution substations, meaning that each service cable can only be connected to precisely one substation. Every cable starts from a fuse in a cabinet and ends in another fuse in another cabinet. If the ending cabinet of the cable does not have any cable that starts from it, it is called *dead end*. The state of each fuse can be either open or closed; this information, combined with the grid's topology, can be used to determine the reachability of each cable on the network from each one's substation. The consumption values for each user are given through its smart meter. Each cable's current load summarizes the production and consumption values of all the users on this cable.

From the discussion so far, it is understood that for a particular mode of operation of an electrical grid, the dynamic control of all possible disturbances, including short circuits, which can lead to network instability, is a complicated and computationally costly process that requires the calculation of a set of detailed operational parameters. During the static study of a grid, time-related electrical parameters are not considered. The static study of a possible disturbance is done with the help of a load flow analysis algorithm. In general, the problem of load flow is to calculate[1] the power flows and voltages in a network when the conditions of the balances are determined [142]. During the static analysis of a network, several possible disturbances can be studied, such as overloads. Load flow analysis is achieved by applying iterative algorithms that require a lot of computational resources. The current load of each cable can also be approximated, in a fraction of time, using methods such as described in [74]. Accordingly, the load percentage of a cable is obtained by dividing its current load by its maximum ampacity multiplied by one hundred. Then the cable is at risk of *overloading* if its current load is over a predefined threshold.

Moreover, modern electricity systems' mission is to produce, transmit, and distribute electricity at the lowest possible cost and with a reasonable level of quality and continuity of supply to their customers. The quality requirement refers to the need to satisfy the specifications for the supply of electrical power to customers with a frequency and supply voltage within the prescribed limits. The design engineers of electricity generation and transmission systems have in the past taken seriously the requirements of a reasonable level of continuity and quality of supply in the future, which is why electricity systems include, from the design phase, many additional physical and functional elements that supply customers with increased level power, continuity security, and higher quality level.

[1] During the load flow analysis procedure for the *slack/swing bus*, the voltage measurement $|V|$, assumed $|V| = 1.0$ if not specified, and the angle δ, assumed $\delta = 0°$ if not specified, are given, and the generated active and reactive power P_G and Q_G, are calculated respectively. For the *load/PQ bus*, the active, P_G, and reactive, Q_G, power produced are specified, while the voltage measurement $|V|$ and the angle δ are calculated. For the *generator/machine/PV bus*, the generated active power P_G and the voltage measurement, $|V|$, are given, and the generated reactive power Q_G and the angle δ are calculated. For the *voltage controlled bus*, the active, P_G, the reactive power, Q_G, and the voltage measurement, $|V|$, are specified, and the transformation ratio α and the angle δ are calculated [142].

One indication of this interest is the redundancy in all functional areas of the systems and many forms, such as reserves in power plants, interconnections with neighboring networks or countries, other transmission - distribution networks, and simple or complex alternative supply in distribution systems. These excess elements are installed because the basic design philosophy recognizes and predicts the system components' possible failures and the need to put them out of order for their planned preventive maintenance. Consideration of all these requirements in feasibility studies, planning, design, and operation of an electrical system is usually defined by the term *reliability assessment* [38, 39]. Moreover, there is ongoing demand from the grid operators for more reliable smart grids towards the "self-healing" grid [134], automatically responding to problems and minimizing disturbances. Therefore, if the recovery response solution could be tested for its efficacy over the next day, the smart grid operators would be of great use to ensure the solution is as robust as possible.

1.2 Aim and Objectives

The demand for a "self-healing" grid requires novel tools to provide a resilient and reliable power grid to its users. The overall aim of this thesis is to present a fully-edge study on reliability optimization for smart grids to provide, fast, the most appropriate countermeasures after a potential overloading disturbance. The research studies simulation, optimization methods, and their combination and suggests implementations, evaluated in a real-world smart grid topology.

The following objectives have been established to achieve the thesis aim:

- To explore the overloading disturbances in a power grid.
- To identify the counteractions to recover from an overloading incident.
- To describe the Deterministic OPP and formulate it as a Multiobjective Mixed Integer Quadratically Constraint Program (MO-MIQCP).
- To propose a solution method using a combinatorial optimization approach with a state-of-the-art exact solver.

- To examine reliability assessment on power grids.
- To apply MCS to estimate the reliability for a planning horizon with unknown future electrical values.
- To describe the Stochastic OPP and formulate it as a Single-State Stochastic Program (SSSP).
- To propose a *simheuristic* approach to solve the Stochastic OPP.
- To propose a software monitoring and alerting system to reduce false positive alarms for meter reading failures, based on live machine learning techniques.
- To critically evaluate the proposed solution methods in a real-world smart grid topology.

1.3 Contribution

This dissertation fulfills the aim mentioned above by extending the applicability of simulation and mathematical optimization to the domain of power grids. In the following, a short overview is provided about each contribution.

Deterministic Overloading Prevention Problem. Given the topology and the energy data of a microgrid at the current time, the potential overloading incidents are detected, and the optimal countermeasures are calculated for the next measurement interval. The grid operators can apply the proposed actions to recover the grid from the disturbance. Into this contribution, the problem is defined and formulated as a MO-MIQCP. This contribution also suggests a solution method using a combinatorial optimization approach with a state-of-the-art exact solver.

This contribution is based on the work that has been presented in the following paper:

- Antoniadis N., Cordy M., Sifaleras A., Le Traon Y. (2020) "Preventing Overloading Incidents on Smart Grids: A Multiobjective Combinatorial

Optimization Approach". In: Dorronsoro B., Ruiz P., de la Torre J., Urda D., Talbi EG. (eds) *Optimization and Learning*. OLA 2020. Communications in Computer and Information Science, vol 1173. Springer, Cham.

Reliability Analysis through Simulation. After a potential overloading incident, smart grid operators would be of great use to ensure stability after a potential overload for a planning horizon, as the future electrical values are unknown. Into this contribution, reliability analysis through simulation is employed to evaluate the topology reconfiguration's robustness after a disturbance, like an overload.

This contribution is based on the work that has been presented in the following paper:

- Antoniadis N., Cordy M., Sifaleras A., Le Traon Y. (2021) *A variable neighborhood search simheuristic algorithm for reliability optimization of smart grids under uncertainty*. Manuscript submitted for publication.

Single-stage Stochastic Overloading Prevention Problem. It differs from the deterministic problem as the optimal countermeasures are calculated for a measurement horizon, *e.g.*, 24 h. This contribution defines the corresponding SSSP and proposes a simheuristic method to solve it.

This contribution is based on the work that has been presented in the following paper:

- Antoniadis N., Cordy M., Sifaleras A., Le Traon Y. (2021) *A variable neighborhood search simheuristic algorithm for reliability optimization of smart grids under uncertainty*. Manuscript submitted for publication.

Smart Meter Communication Monitoring. In this derivative contribution, a novel software monitoring, and alerting system is introduced based on live machine learning techniques. More specifically, we suggest using ND-trees to learn for each smart meter a failure pattern over time, which then acts as a profile for the respective smart meter. This profile is used to decide, in real-time, if an alarm should be raised or if the reading error can be considered as "normal". This contribution can help grid operators decide when reading failures can be considered uncritical and ensure that the electrical data are clear from any false reading failures.

This derivative contribution is based on the work that has been presented in the following paper:

- Antoniadis N., Cordy M., Le Traon Y. (2021) *Intelligent Smart Meter Communication Monitoring by Learning Failure Patterns using ND-Trees*. Manuscript in preparation.

1.4 Thesis Structure

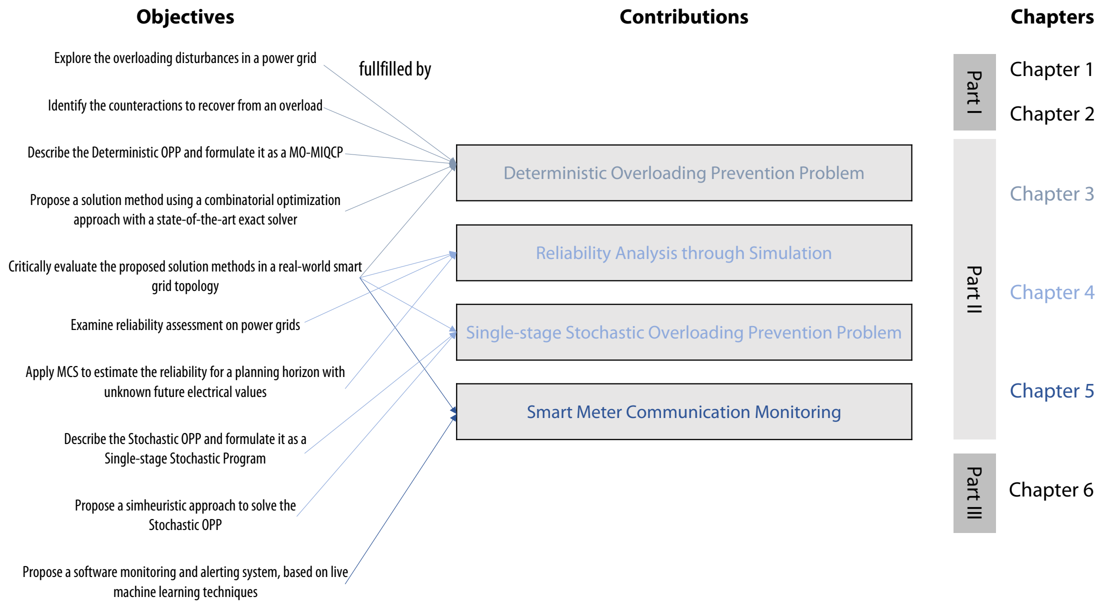


Figure 1.3: Thesis structure.

This thesis encompasses three parts. The first part introduces the technical background of this dissertation and the related work. Then, the central part of this manuscript fulfills the thesis' objectives, presented in Section 1.2, by extending the applicability of simulation and mathematical optimization to the domain of power grids. Each of the chapters starts with an introduction to and a motivation for fulfilling the thesis' objectives. In this way, each of these chapters can be read independently. Finally, a third part presents the conclusion of this dissertation. Figure 1.3 depicts the structure of the dissertation along with the contributions made in this thesis and shows which of the objectives described in Section 1.2 are fulfilled. In the following, an overview of each part is provided.

Part I: Prologue. This part is composed of the present Chapter 1, which introduces the context of this thesis, and Chapter 2 which presents the technical background and the related work regarding this dissertation.

Part II: Enhancing Smart Grid Resilience and Reliability. This is the part of the contribution. It defines the core and derivative concepts and foundations of this thesis. More detailed, in Chapter 3, the Deterministic OPP is defined, and formulated a MO-MIQCP. Then, a solution method using a combinatorial optimization approach with a state-of-the-art exact solver is suggested. This approach is evaluated using a real-word topology [34], showing that the proposed method can suggest optimal countermeasures to grid operators who are facing potential overloading incidents. In Chapter 4, reliability analysis through simulation is discussed and employed to evaluate the robustness of applying the suggested countermeasures, which calculated by the solution method suggested in Chapter 3, for a planning horizon, e.g., the next 24h. Then, to achieve a more robust configuration, an SSSP is proposed to suggest such a grid topology configuration, after a potential overload, to ensure stability for a planning horizon. A simheuristic approach based on a Variable Neighborhood Search (VNS) metaheuristic is presented to solve the problem mentioned above. After evaluating the two approaches, it is shown that especially the simheuristic, the proposed grid topology configuration can lead to a highly possible reliable grid for the planning horizon. At the end of Part II, in Chapter 5 a machine learning-based approach is presented that can continuously learn the "normal" pattern of the smart meter communication to help grid operators to decide if reading failures have happened, that could indicate severe disturbances in the grid, or there happened due to certain conditions, like noisy solar panels. The evaluation of this approach shows that it can significantly reduce false negatives, avoid unobserved, critical for the smart grid, communication errors, and false positives, limiting the number of sending technicians unnecessarily to investigate why a smart meter is not reachable. This approach can also help electrical data to be clear from any false reading failures.

Part III: Epilogue. This part gives the conclusion of the thesis. In Chapter 6 the concluding points are explained, and possible future research directions are discussed.

2 Background and Related Work

> If we knew what were doing, it wouldn't be called research.
>
> *(Albert Einstein)*

This chapter presents the technical background related to smart grid resilience and reliabily, simulation and simulation-optimization methods. Along with the background, this chapter provides an overview of works on related research topics. Special care was taken to exhaustively cover all the published studies until the time of writing.

Contents

2.1 Stability in Electrical Grids

The analysis of possible disruptions for electrical networks is the latest years, a growing field of research due to the imperative needs of uninterrupted energy production within a competitive environment. The energy production's specifications require minimizing the chances of failure to cover loads either locally or generically. They also presuppose the production of constant frequency and voltage energy and the electrical system's ability to remain in the permanent working condition under extraordinary conditions. The competitive environment and the liberalization of the energy market have led to interconnected electrical networks that are more stable. However, at the same time, due to competition, they have to operate increasingly close to their limits, making them more vulnerable.

Nowadays, the design and implementation of stable and durable electrical systems disturbances is now a separate field in electrical engineering. The stability, safety, reliability, and optimal economical operation of electrical networks is the issue. Generally, *stability* is defined as the ability of the system to remain in functional equilibrium or synchronization during disturbances [98]. Three types of stability are distinguished: stability during a permanent operating mode, dynamic, and transient. Stability during a permanent operating mode refers to a synchronous machine's response to a progressively increasing load. Dynamic stability refers to the system's response to small disturbances that cause oscillations, while transient one refers to the system's response to large disturbances that affect machine torsional moments, angles, and power transfers [98].

The dynamic stability study refers to either increasing amplitude oscillations in the system, which lead either to dynamic instability or to dynamic stability due to decreasing amplitude oscillations. The system's response to a dynamic stability disturbance may not be perceived for 10 to 20 seconds, while the system's response to transient phenomena is usually perceived within the first second [98]. Kundur *et al.* [90], define the operating states for an electric power system into five categories: normal, alert, emergency, extreme, and restorative, illustrated in Figure 2.1. During the normal state, the power system operates securely. If the system's state variables, such as voltage and current, are over particular but still acceptable limits, the system enters into the alert state. The system enters into an

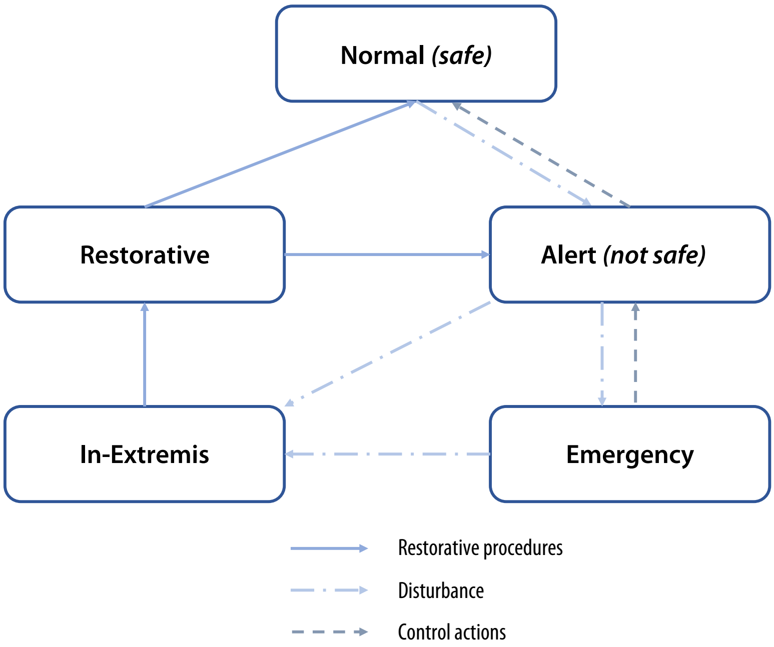


Figure 2.1: Operating states for electrical power systems.

emergency state if the system cannot withstand the potential overload anymore. An extreme state occurs when the severity of the disturbance increases, while the restoration and reconnection procedure is represented by the restorative state [2, 41, 90].

2.2 Preventing Overloading Incidents

Even though the prevention of overloading incidents concerns the grid operators, this problem is not studied enough. Several research works investigate how to prevent overload incidents using demand response programs. To the best of our knowledge, there is no detailed work that examines the overloading prevention problem in respect to demand response for both producers and consumers and grid reconfiguration at the same time.

Ramaswamy and Deconinck [113] define the grid reconfiguration problem as a multiobjective non-convex one and argue that a genetic algorithm is probably a suitable optimization method to solve it. Han and Piette [64] describe different incentive-based demand response programs; usually reducing demands with a financial benefit for the customers. They present different direct load control methods, interruptible/curtailable rates, emergency demand response programs, capacity market programs, and demand bidding/buyback programs. To prevent overloads, Bollen [20] present different curtailment schemes averting the operating reserve from getting insufficient, that could lead to overloads. In this work, curtailment's general directions are given without giving many details about modeling and solving the curtailment problem. Furthermore, Simao *et al.* [131] formulate the problem of planning short-term load curtailment in a dense urban area as a stochastic mixed-integer optimization problem. They implement three approximation policies and test them with a baseline policy where all curtailable loads are curtailed to the maximum amount possible. Even if short-term planning is implemented in their work, overloads are allowed, and the curtailment applies only to consumers of the grid.

In addition to the previous studies, in [55], Goyal *et al.* propose a control strategy of a Battery Storage Unit (BSU) to prevent overloading in an autonomous hybrid microgrid by setting it online, depending on the frequency signal. Xyngi and Popov [147] are describing an algorithm for coordinating relays on a Dutch medium voltage network to solve the false tripping problem that happens when the relay does not recognize the fault current direction. Correa *et al.* [33] present a binary programming model for online coordination of directional overload relay problems in interconnected power distribution and sub-transmission systems. In [93] an adaptive overcurrent protection strategy for a microgrid network is presented. They treat the overcurrent relays' protection coordination as a linear programming problem for the different operation states. An artificial neural network is trained with real-time measurements to identify whether there is a fault on the line segment. Recently, Dehghanpour *et al.* [36] presented the optimal coordination of directional overcurrent relays in microgrids as a non-linear programming problem. They combined Cuckoo Optimization Algorithm with linear programming as a hybrid optimization algorithm to optimize the coordination protection of directional overcurrent relays in microgrids. In addition to the previous studies, Pashajavid *et al.* [103] present an overload management strategy that controls the supporting floating batteries in an autonomous microgrid and decides any possible connection between it and its neighboring microgrids by monitoring the microgrids' frequency. However, in their work, no demand response program is considered.

Furthermore, Shahnia *et al.* [126] developed a dynamic multi-criteria decision-making algorithm to manage microgrid overloads. They also deploy a cloud theory-based probabilistic analysis to contemplate the uncertainties in the considered distribution network. Nevertheless, they were not considering reactive power in their approach to define overloading. Recently, Babalola *et al.* [10] proposed a multi-agent algorithm that does not require load shedding to prevent cascading failures, such as overloaded lines after a contingency occurs. Nonetheless, their work is focused on power generators only.

2.3 Reliability Analysis in Electrical Grids

Generally, the term *reliability* is used to denote the overall ability of a system to perform its intended function [19, 25, 32, 132]. Electrical systems are prime examples of systems with an expected high level of operational reliability. Modern social conditions demand a high priority in achieving a satisfactory operational reliability level and the rapid reconnection of the supplied power after a power outage. In most countries, the average duration of power outages for a consumer is 2-3 hours per year. Thus, reliability analysis is one of the most critical stages of forecasting, planning, design, operation, and electrical systems maintenance. For many decades a satisfactory level of reliability was achieved through empirical methods and rules. However, as systems grew in size and complexity, the need for rigorous analyzes grew, and in recent years, standard principles and methods for calculating operating reliability have been applied to every area of electrical reliability studies of power systems. The actual level of operational reliability of a customer varies from region to region, while different transmission or distribution networks display different levels of reliability. It is easy to see that to achieve the desired level of reliability at the customer level, each system's higher levels must have an even higher reliability level.

In a quantitative sense, reliability can deal with understanding and decisions making while dealing with complex situations. The term *reliability* is used, in this dissertation, to denote, quantitatively, the ability of the smart grid to avoid overloading events. The application of quantitative methods of calculating operating reliability is the only way all conflicting and uncertain factors affecting electrical systems' operation can be considered individually. The reliability analysis results can be correlated with the financial aspect of the system's planning, design, and operation. Simultaneously, it is evident that the corresponding effects that arise significantly determine the system's present and future development. Modern electricity systems' economic operation is a significant problem because many and peculiar factors influence it. Coincidentally, systems' role is crucial in economic and social life and its inhabitants' quality of life. Therefore, there is increased pressure from many social groups and organizations that have not existed during the past decades. Therefore, it is necessary to consider with great care the various areas of problems related to the reliable operation of current electrical systems and to recognize the need to develop methodologies for calculating the reliability indicators of systems and data collection procedures required by historical their functional characteristics.

The modeling of the electrical systems' operational reliability and the calculation of their reliability indicators has been the subject of research [4, 16, 17, 18, 19]. The purpose of these publications is to present the numerical analysis techniques that realistically reflect these systems' operational behavior and their impact on customers. These techniques calculate an appropriate set of performance reliability indicators for each analyzed system's load balances and allow studies to be performed to evaluate alternative systems reinforcement designs.

2.3.1 Methodology

One of the objectives of the programming of electrical systems is to strengthen them with equipment, such as production units, transmission lines, and transformers, necessary to supply economically and reliably a projected level of load demand. These two objectives are conflicting because the higher level of reliability requires the system's strengthening, which results in the increased cost of supplying customers' demands. The scholars' responsibility is to achieve the best degree of correlation between price and level of reliability by considering the uncertainties in the change in loads and the indicators of system data availability.

Over the past decades, researchers of electrical systems have traditionally used deterministic criteria [16, 39]. A studied system's performance is calculated for several different scenarios representing functional conditions considered critical and reasonable in such a causal process. For example, a scenario can be defined by the annual peak of system load demand, the loss of any transmission line, and the loss of the system's largest production unit. The design of a system is considered acceptable if it can feed the load in any considered scenario. In such cases, the most economical method is chosen from those considered adequate according to the deterministic criteria used.

Deterministic criteria present a significant number of attractive modeling features. An understandable and straightforward implementation process while dealing with critical and non-functional conditions is usually compatible with past principles to design a reliable system. However, these criteria present specific modeling limitations that have been recognized in the past, such as:

- the difficulty of determining the level of operational unreliability of a system that can be damaged according to more than one scenario
- the difficulty of identifying a system that has a more significant margin of safety than two or more systems that meet all criteria and scenarios
- the creation of a non-economical design for the studied systems in comparison with the corresponding optimized methods that result considering the possible occurrence of each scenario and the financial implications of the reduced operating efficiency according to the relevant scenarios

The probabilistic methodology is the alternative modeling methodology in which the stochastic features and functional processes are fully represented. For many decades the principles of modeling and the practical advantages of probabilistic methods have been successfully recognized and promoted [16, 39]. However, such a transition from deterministic to probabilistic methods is relatively slow. It is due to objective difficulties, the most important of which are the following:

(i) *Data Collection*: Statistics of system equipment failure are based on historical records, which usually have significant deficiencies and include many errors, such as human and entry errors. At the same time, older files are not stored on computers, and therefore there is increased effort to collect and analyze them.

(ii) *Modeling of Probabilistic Phenomena*: Probabilistic models of representing faults in system equipment can be quite tricky. They may require laborious calculations and depend significantly on the type of equipment, their age, and their functional characteristics.

(iii) *System Response Modeling*: The probabilistic analysis and the calculation of probabilistic indicators are performed by applying computational models. Such models can accurately simulate the system's functional characteristics in a larger number of operating conditions. The development, control, and implementation of such computing tools is a significant investment in technological thinking and computer equipment.

However, it is evident that despite the above difficulties, the possible analysis of the electrical systems' operational reliability has begun to become more and more enjoyable, both for the designers and the operators of such systems. Several computer programs have been developed with increasingly detailed modeling and approximation methods, intended for general or specific applications. In this direction, enormous difficulties had to be faced in development, which appears in modeling, information gathering, and computer simulation. The fundamental problem has proven to be to strike a balance between the most realistic model possible and the corresponding volume of calculations that need to be solved so that operators can obtain results within acceptable time limits.

The application of probabilistic techniques for calculating the operating reliability of electrical systems can be divided into the following main areas of interest:

- Electrical power generation capacity
- Transmission and distribution of electrical power
- A combined system of electrical power generation and transmission

- Electrical supply systems in industrial installations

The first area concerns the power generation system. Its primary purpose is to determine the amount of electricity generation needed to meet, with a reasonable degree of safety, customer demand requirements. The transmission network must carefully connect to the generation system to allow power to flow from the sources to the distribution networks that supply the customers' loads with direct and often radial routing. The transmission and distribution networks must also maintain the system's quality requirements: the system voltage balances' satisfactory level and the loading of the branches with power flows less than or equal to their thermal limits. Finally, power supply systems in industrial installations are exceptional cases of distribution systems with particular functional characteristics that significantly affect their reliability level.

Different models have been proposed to determine the effects of random disturbances, like overloads, on the reliability assessment of a smart grid system. These models can be grouped into either *analytical* or *MCS* methods, as have been presented in the international literature [4, 16, 17, 18, 19, 24, 28, 38, 39, 63, 119, 132].

Analytical methods, are widely developed in North America and several European countries, based on detecting and analyzing system conditions that may lead to a fault condition. Analytical methods include [132, 138]:

- *State space* method using Markov processes, where all possible system states are enumerated; consequently, this approach can be regarded as the most direct approach to calculating reliability indices [132].

- *Network reduction* method, where the serial and parallel elements of a system are reduced to a single equivalent element; hence the reliability of the reduced system equals the reliability of the initial system [138].

- *Conditional probability* method, where the complex, not serial, or parallel, elements of a system are simplified, via the conditional probability concept, to a combination of serial and parallel structures; then the network reduction method can be used to find the reliability of the system [132].

- *Cut-set* and *tie-set* method. The minimal cut set contains a set of system components that, when they fail, would cause the failure of the system. On the opposite, the minimal tie set consists of a set of system components that, when successful, would lead to system success. Then, the reliability indices can be calculated by the probabilities of failures or successes [77, 132].

The above analytical approaches could be used when the problems are relatively easy to be modeled and solved. However, in complex models, MCS should be employed to determine the effect of random failures on the reliability of the system, as it is more flexible in dealing with challenging operating conditions and system considerations [132]. Thus, using simulation, we can estimate the reliability indices by constructing realizations of the stochastic values. MCS can be categorized as *non-sequential* [107], and *sequential* [106]. In Non-Sequential Monte Carlo Simulation (NSMCS), the states of the system are randomly sampled, while in Sequential Monte Carlo Simulation (SMCS), the system states are simulated in chronological order, providing statistically more reliable information [62].

Reliability, as defined above, is concerned with the ability of a power system to avoid overloading incidents. To quantify the expected reliability, indices are used to express the probabilistic measures of the examined system. Some of the most common reliability indices [25, 32, 43], introduced by [79], are the System Average Interruption Frequency Index (SAIFI), the System Average Interruption Duration Index (SAIDI), the Customer Average Interruption Duration Index (CAIDI), the Average System Interruption Frequency Index (ASIFI), and the Average System Interruption Duration Index (ASIDI).

MCS method, which began to be applied mainly in Italy, France, and Brazil and has been extended to all countries of the world, is based on the simulation methodology, which simulates the system's operation during the considered analysis period. Both methods have respective advantages and disadvantages. The analytical approach can detect rare but significant potential faults and can be considered a standard procedure for identifying specific systems' flaws. The MCS method often handles the frequent failures in the system components, which may not cause system failure because the system's smooth operation is not disturbed. This method should also be applied in all cases where the chronological order of the considered contingencies needs to be analyzed, such as when accurate modeling of hydropower plants' operation is required, reservoir model consideration, and complex interdependencies between different system parameters. It is essential to understand that the computational effort needed for the calculations applying the MCS method depends little on the system's size concerning the analytical methodology's corresponding dependence. The reliability assessment part of the present dissertation is focused on the MCS method because the electrical systems' characteristics require detailed and complex modeling that could not be achieved using the analytical process.

2.3.2 Basic Principles of the MCS

Simulation, in its broadest sense, is a technique of forming experimental samples in a system model. The simulation method is used, showing advantages over other methods. In cases where the system being analyzed is so complicated that it cannot be described by a set of mathematical equations or when a mathematical model can be formed to represent the system but is not possible to achieve a solution by narrowly analytical methods. The simulation makes it possible to study and experiment, taking into account detailed observations of the simulated system. During the simulation design phase, the knowledge gained often indicates the proposed changes in operating practices applied to the system.

The MCS method is straightforward to apply, especially in complex systems, and very large to be solved with the analytical process. However, most issues of systems reliability analysis result in simulations to determine rare cases. From this point of view, the MCS leads to a colossal waste of computation time, which is the main disadvantage of the method. The MCS method's application presupposes creating the random variables' appropriate values using the corresponding distribution function proposed for the considered model. This procedure can be achieved systematically for each variable by first generating evenly distributed random numbers from 0 to 1 and then, employing appropriate transformations, generating the corresponding set of random numbers with the given distribution function. The production of evenly distributed random numbers is a significant step in the MCS. Before the discovery of the first computers, many attempts were made to construct lists of random numbers. Today, software companies supply their programs with internal subroutines for random number generation. All uniformly distributed random number generation techniques are based on calculations using retrograde residual calculus relations, mod m, resulting from a linear transformation. The most widely used method of generating random numbers today is the multiplication convergence method, in which the remainders of the successive forces of a number X are the sequential numbers in a series of random numbers [4, 63, 119].

One of the most common methods of generating random numbers starts from an initial value X_0 and calculates successive values $X_n, n \geq 1$:

$$X_n = aX_{n-1} \bmod m \tag{2.1}$$

where a and m are positive integer numbers.

A general method of generating random numbers from a given distribution function considers a distribution function F of the continuous random variable u and looks for the value of x, so that $F(x) = u$. This value of the variable x is found from the calculation of the inverse function:

$$x = F^{-1}(u) \tag{2.2}$$

which means that if $(u_1, u_2, \ldots, u_n)$ is a set of values of the variable u, the corresponding set of values obtained assuming the above equation is:

$$x_i = F^{-1}(u_i), i = 1, 2, \ldots, n \tag{2.3}$$

Applying Equation 2.2 produces random numbers that are evenly distributed and correspond to a considered probabilistic distribution. It is necessary to determine the inverse function from the distribution function, and for this reason, this method is known as the *inverse transform method* [121] can be applied very easily in cases where the inverse distribution function has an analytical expression.

The simulation of a probabilistic model involves creating the model's stochastic mechanisms, and all observations of the model flow over time. Depending on the purpose of the simulation development, some parameters need to be defined. However, the change of the model over time often involves a complex logical structure of the elements that make it up, and it is not always obvious to be able to monitor this change to determine the required parameters. A general rule has been developed based on the idea of discrete events and has its objective to monitor the model's flow over time and determine the numerical values of the required parameters. This methodology is called Discrete Event Simulation (DES) [12].

The main parameters of a DES are variables and events [9]. The development of a simulation requires the constant monitoring of some variables. There are three types of variables used:

- *Time variable* t: Refers to the total time elapsed since the simulation's start.
- *Statistical counter variables*: Determine the frequency of events that occur during time t.
- *System state variables*: Describe the state of the system at time t.

Each time a "contingency" occurs, the values of the above variables change or are renewed. At the same time, all the required data are collected as an output of the process. A list of contingencies is kept to determine when the next event will occur. This list records all the nearest future contingencies and the time they are scheduled to occur. Whenever a contingency "occurs", all state variables and variable counters return to their original value while the relevant data is being collected.

A simulation study is commonly used to determine the value of a parameter θ associated with a stochastic model. A simulation of the relevant system results in calculating a random variable X whose expected mean value is the parameter θ. A second simulation, which means a second flow, results in a new and independent random variable having an average value θ. This calculation process continues until k flows are completed and k random variables $X_1, X_2, \ldots, X_k$ are found, which are all uniformly distributed with mean θ. The mean of these k values is used as the estimator or approximate value of the parameter θ. A related problem of any simulation process is the decision related to the time it must stop, which means that the appropriate value of k must be decided. The quality of the estimator of θ can be used as a criterion for this decision. A confidence interval can also be found in which the parameter θ lies with some uncertainty level.

If $X_1, X_2, \dots, X_k$ are random independent variables that have the same distribution function while θ and σ^2 are respectively their mean and variance, the arithmetic mean and the variance of the sample are defined as:

$$\overline{X} = \sum_{i=1}^{n} \frac{X_i}{n} \tag{2.4}$$

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})}{n-1} \tag{2.5}$$

and can be considered as impartial estimators of the parameters θ and σ^2, respectively.

A simulation process is continued until k values are generated to estimate the mean value θ for which the coefficient of variation $\beta = s/(\sqrt{k}\bar{X})$ must be less than an acceptable value d. When the sample size is small, the standard deviation of the sample s may not be a good estimator of σ. Therefore, a small number of initial values are generated to estimate the sample's standard deviation and the corresponding coefficient of variability. This process follows the following steps:

- Selection of an acceptable value d of the coefficient of variability β.
- Generation of a small number of initial values for estimating the coefficient of variability β.
- Continue producing new values until k values have been generated for which $\beta < d$ where β is the coefficient of variation based on these k values.
- The estimator of θ is given by $\overline{X} = \sum_{i=1}^{k} X_i/k$

2.3.3 Variance Reduction

In the simulation method, various variance reduction techniques have been developed, aiming to reduce the uncertainty associated with the samples' finite size and, therefore, allow the simulation to be terminated faster, without compromising the accuracy of the results. The following variance reduction techniques are mainly used in the reliability analyzes of the operation of electricity systems [4, 9, 63, 119, 120]:

(i) *Antithetic Variates*. For the estimation of $\theta = \mathbb{E}(X)$, the variables X_1 and X_2 are produced, which are uniformly distributed random variables and have an average value θ. Then it holds that:

$$\operatorname{Var}\left(\frac{X_1 + X_2}{2}\right) = \frac{1}{4}[\operatorname{Var}(X_1) + \operatorname{Var}(X_2) + 2\operatorname{Cov}(X_1 + X_2)] \tag{2.6}$$

From this expression it is clear that the variance $\operatorname{Var}\left((X_1 + X_2)/2\right)$ is smaller than $\operatorname{Var}(X_1)$, if the variables X_1 and X_2 are negatively correlated.
The above findings lead to a simulation process where the generated random numbers $U_1, U_2, \dots, U_m$, which are evenly distributed in the interval $(0, 1)$, are used to calculate the random variable X_1 while the set of random numbers $(1 - U_1), (1 - U_2), \dots, (1 - U_m)$ is used to calculate the variable X_2 so that it is negatively correlated with the random variable X_1. This process has a double benefit because the calculated estimator has reduced variance, and the calculation time is significantly less because no additional time is needed to generate a second set of random numbers.

(ii) *Control Variates*. The control variate technique considers a random variable Y, which has a known mean value equal to $\mathbb{E}(Y) = \mu_Y$, which is related to the random variable X and uses the variable $Z = X + c(Y - \mu_Y)$, which also is an unbiased estimator of the parameter θ; $\mathbb{E}(Z) = \mathbb{E}(X) = \theta$. The constant c is determined so that the variance of the random variable Z takes a minimum value.

$$\operatorname{Var}(Z) = \operatorname{Var}(X) + c^2 \operatorname{Var}(Y) + 2c \operatorname{Cov}(X, Y) \tag{2.7}$$

The minimum value of constant c, ($c = c^*$), is calculated as

$$c^* = -\frac{\text{Cov}(X, Y)}{\text{Var}(Y)} \tag{2.8}$$

while the minimum value of the estimator variance is equal to:

$$\text{Var}[X + c^*(Y - \mu_Y)] = \text{Var}(X) - \frac{[\text{Cov}(X, Y)]^2}{\text{Var}(Y)} \tag{2.9}$$

The variable Y is called the control variate for the estimator of the variable X of the simulation. This technique is quite efficient because the c^* parameter takes negative or positive values when the variables X and Y are positively or negatively correlated, respectively. If X and Y are assumed to be positively correlated, X receives large values when Y receives large values and vice versa. The indication of a large or small value means that the corresponding variable's value is greater than its known mean value. If a simulation flow results in a large, or small, value of Y, then X takes a correspondingly higher, or lower, value than the mean value of θ. Therefore, it is desirable to correct it by decreasing, or increasing, the X's estimator value, which is achieved when the parameter c^* is negative, or positive, respectively. Similar reasoning is inferred when the variables X and Y are negatively correlated.
The variance of the variable Y, $\text{Var}(Y)$, is known before the start of the simulation while $\text{Cov}(X, Y)$ is almost always unknown because the objective purpose of the simulation is to estimate the unknown value of $\mathbb{E}(X)$. It is evident that the value of c^* is not known before the simulation process begins. However, because the maximum gain of using the variable control technique is achieved by selecting a value c close to c^*, a substantial improvement in the estimation of the variable X is achieved by considering the results of a small preliminary number of simulations. The value found is used for the remaining number of simulations until the convergence criteria are met.

(iii) *Variance Reduction by Conditioning.*

The calculation of expected values by conditioning is a classic technique in probability theory. When $\mathbb{E}(X)$ needs to be calculated, sometimes it is useful to suppose there is a second random variable Y, so that:

$$\mathbb{E}(X) = \mathbb{E}\big(\mathbb{E}(X|Y)\big) \tag{2.10}$$

The variance can also be computed like:

$$\mathrm{Var}(X) = \mathbb{E}\big(\,\mathrm{Var}(X|Y)\big) + \mathrm{Var}\left(\mathbb{E}(X|Y)\right) \tag{2.11}$$

As all the values in Equation 2.11 are non-negative, we can have:

$$\mathrm{Var}(X) \geq \mathrm{Var}\left(\mathbb{E}(X|Y)\right) \tag{2.12}$$

that leads to variance reduction by conditioning.
The use of this reduction is useful when the goal is to estimate $\theta = \mathbb{E}(X)$, and there is such another random variable so that the value $\mathbb{E}(X|Y = y)$ is known. Equation 2.12 shows that $\mathbb{E}(X|Y)$ is an unbiased estimator for $/theta$, superior to the estimator X.

(iv) *Stratified Sampling.*
Supposedly, we want to estimate $\mathbb{E}(X)$, and, somehow, a random variable X is related to another random variable Y, that can take discrete values y_j with known possibility. Thus, Y follows a discrete distribution with known probability function:

$$PY = y_j = p_j, \ j = 1, \ \dots, \ m \tag{2.13}$$

Using conditioning, we have:

$$\mathbb{E}(X) = \sum_{j=1}^{m} \mathbb{E}(X|Y = y_j)p_j \tag{2.14}$$

Simulation can be applied to estimate the values of $\mathbb{E}(X|Y = y_j)$, $j = 1, \dots, m$, so that using Equation 2.14, $\mathbb{E}(X)$ can be estimated, as:

$$\mathcal{E} = \sum_{j=1}^{m} \overline{X_j} p_j \tag{2.15}$$

where $\overline{X_j}$ is the average of the np_j observed values of X generated conditional on $Y = y_j$. The unbiased estimator $\mathcal{E}$ is called a *stratified sampling* estimator of $\mathbb{E}(X)$.

2.4 Stochastic Programming

When decision problems are modeled and solved deterministic, in many extreme and exceptional cases, the expected values of the variables are often excluded and taken into account. According to Masse [143], such a thing involves danger, as something that is considered a detail at the time of analysis, maybe later crucial. Masse also mentions that looking towards a causal future is overly optimistic and lacks flexibility. It is done so clearly that deterministic models fail to represent the future satisfactorily and their accuracy decreases as the subject in question grow the time horizon of the problem. Therefore, the concept of stochastic programming is introduced, which is a modeling method for problem optimization involving variables and constraints with a vital element of uncertainty. This uncertainty stems from the lack of reliable data, from measurement errors or parameters containing future information. It is obvious that stochastic programming is a category of problems and not a method of solving them. Many of the models belonging to this category can be solved either with mathematical programming tools, like the one presented in Chapter 4, or Stochastic Dynamic Programming (SDP). The stochastic programming problems can be categorized as:

- *Single-stage Stochastic Programming.* The single-stage stochastic optimization problems refer to the problems where the decision is taken instantaneously, and no correction to the random variables is obtained into account.

- *Two-stage Stochastic Programming*. In them, some decisions are taken deterministically within a specific one period of time, that is, in the first stage, after the end of which they occur random events. A set of retrospective decisions can be made at the second stage to compensate for any adverse effect it has observed since the first decision was taken.

- *Multi-stage Stochastic Programming*. A generalization of the two-stage model is the multi-stage model. In this case, each stage consists of a decision followed by a series of uncertain parameters over time.

Random parameters in stochastic programming are displayed as scenarios. These scenarios are shared as part of their stochastic information and create a graph called a scenario tree. Power system design problems have always been a field of programming applications under uncertainty. Thus the use of stochastic programming is considered useful and often imperative.

2.4.1 Stochastic Programming in Power Grids

Stochastic parameters that appear in the power grid problems [37, 51, 70, 95, 99, 116, 123, 133, 140, 145] refer, among other things, to load demand, hydroelectric dam reservoir inputs, fuel prices, emission allowance prices, unit availability, unscheduled outages, and staff availability.

The above parameters affect the programming according to time horizon of this. In the power grid problems, the time horizon programming is divided into the following categories:

- very long term (5–15 years)

- long term (2–5 years)

- medium term (1 month–2 years)

- short term (1 week–1 month)

- very short term (up to 1 week)

It is indicatively stated that the stochasticity of load demand influences each time horizon's decisions, although it seems that its influence is more significant in very long-term (network expansion) and very short-term models. On the other hand, the stochasticity of hydroelectric dam reservoir inputs, due to their annual cycle of changes, seems to have a more significant impact on the medium-term models. While providing satisfactory solutions to small-scale problems, traditional optimization methods often prove to be inadequate when applied to larger problems. That is why in the present dissertation, described in Chapter 4 it was chosen to model the problem of Stochastic OPP to be done using stochastic single-stage programming. The stochastic parameter, the effect of which in planning is considered, is the possibility of unplanned outages due to potential overloads.

2.5 Simulation-based Optimization

Simulation techniques can model complex systems, although they cannot be used by themselves as an optimization tool [86]. If we combine simulation approaches with optimization techniques, like metaheuristics, the hybrid simulation-optimization (Sim-Opt) can handle the uncertainty in stochastic optimization problems [1, 45]. There are many ways to combine simulation with optimization in simulation-optimization approaches; the major categories are [45]:

- *Solution Evaluation* (SE) approaches. Evaluation Function (EF) and Surrogate Model Construction (SMC) compose the SE approaches. A representing simulation model of the system is developed in these approaches, and different solutions are evaluated. Thus, these approaches are focused on optimizing a simulation model, while the purpose of a simulation is to evaluate the performance of solutions (simulation optimization). The drawback of these approaches is that evaluating various solutions through simulation can be computationally intensive.

- *Solution Generation* (SG) approaches. In some problems, the solution could be chosen without the simulation outcome be needed. A representing analytical model of the system is formulated and solved in such approaches, and different solutions are simulated, *optimization-based simulation*. The purpose of a simulation is to compute all the interesting variables and not to evaluate the solutions. Although these methods can be very effective in the above problems, they may not be effective in other problems.

- *Analytical Model Enhancement* (AME) approaches. In these approaches, usually, the analytical model is hybridized with SE approaches. Simulation results could be another enhancement of the analytical model.

While the first stochastic-approximation methods were suggested about seventy years ago, Simulation-Optimization (SO) techniques [86] have been used for over twenty-five years. In the last twenty years, the SO field has thrived, as it embraced computational power, simulation software, and novel, sophisticated optimization methods, such as hybrid metaheuristics. An illustration of the SO approach is given in Figure 2.2, where in order to find near-optimal solutions to stochastic optimization problems, simulation and optimization methods interact.

Both simulation and optimization communities, autonomously from each other, developed simulation-optimization approaches [45]. The optimization community formed Hybrid Simulation–Analytic (HSA) models/modeling [127], while the simulation community developed the SO approach–concentrated on the optimization of simulation models.

Concerning SO, Swisher *et al.* [135] describe it as a "structured approach to determine optimal input parameter values, where optimal is measured by a function of output variables–steady-state or transient–associated with a simulation model." Therefore, the optimization method evaluates a given solution's performance by utilizing the outputs from the simulation paradigm, and its inputs consist of a collection of decisions. Based on the past, the optimization method, including the current evaluations, chooses a new collection of input values. Consequently, the simulation paradigm acts as an EF of the optimization procedure [5]. A metamodel, or better SMC, may be constructed, by simulating various col-

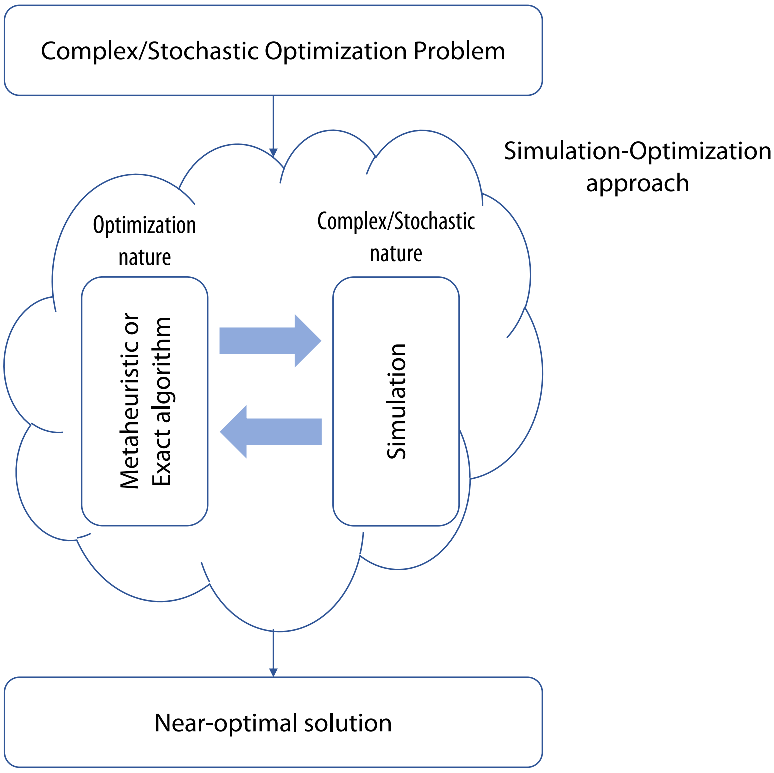


Figure 2.2: Overview of the Sim-Opt approach, adapted from [86]

lections of input values, each being a solution to the problem. Then, classical optimization methods can be used, rather than simulation, to solve SMC [13]. SMC's solution is considered, after this procedure, as an approximate solution to the original problem. In the literature [86], different works exist considering both SMC and EF approaches [48, 137, 144]. OptQuest, a proprietary simulation optimization engine [7, 52] combines both these methods; the SMC part uses a neural network, while scatter search and tabu search metaheuristics are used in the EF part.

An HSA model, as stated by Shanthikumar and Sargen [127], is "a mathematical model which combines identifiable simulation and analytic models," since HSA modeling "consists of building independent analytic and simulation models of the total system, developing their solution procedures, and using their solution procedures together for problem-solving." Any stochastic program with sampled scenarios that have been obtained via MCS is an example of an HSA paradigm.

Thus, in HSA, simulation is used to generate part of the solution (SG) or enhance the analytical model (AME) and not to evaluate the solutions' feasibility or quality. Simulation is used in AME to refine the parameters of an analytical paradigm that is specific to the problem. Since the optimization part is independent of simulation, AME's strategy is less simulation-intensive than EF [86]. Simulation-optimization approaches differ in their focus.For the optimization-driven ones (simulation-based optimization), simulation operates as an 'auxiliary' agent, while the optimization algorithm acts as the 'driving' agent. Conversely, for the simulation-driven ones (optimization-based simulation) optimization, occasionally find, at least, near-optimal, simulation parameters' values, while simulation, as the 'driving' agent, reproduces a random system's behavior [86].

2.6 Simheuristic Algorithms

Simheuristics, or *simheuristic algorithms* [30, 86], as a simulation-optimization approach, combine simulation with metaheuristics to solve stochastic combinatorial optimization problems. They are usually optimization-driven, but, according to their implementation, can be classified as AME (HSA) or EF (SO).

Metaheuristic approaches typically address real-world sized instances of Combinatorial Optimization Problems (COPs). Similarly, real-life stochastic COPs can be naturally addressed by combining metaheuristics and simulation techniques in any variation. A simheuristic algorithm is a simulation-optimization method to adeptly dive into a stochastic COP instance containing stochastic elements, located either in the constraints' set or objective function. Olafsson [100] highlighted the importance of eliminating the differences between theoretical analysis, exact methods, and practical problems. Even though simheuristics can also be applied to complex deterministic problems, this dissertation focuses on stochastic COPs.

In simulation-optimization literature, there are several examples of applying simheuristics to different domains. By combining MCS with routing metaheuristics, Gonzalez *et al.* [54] solved the Arc Routing Problem (ARP) with stochastic demands, while Juan *et al.* [82] solved the Vehicle Routing Problem (VRP) with stochastic demands, and Juan *et al.* [87] solved the Inventory Routing Problem (IRP) with stochastic demands and stock-outs. Additionally, the Permutation Flow-Shop Problem (PFSP) with stochastic processing times was solved by Juan *et al.* [85], who combined a scheduling metaheuristic with MCS. Also, Cabrera *et al.* [26] introduce solving COPs with probabilistic constraints by combining a metaheuristic with DES, where time conditions the random behavior.

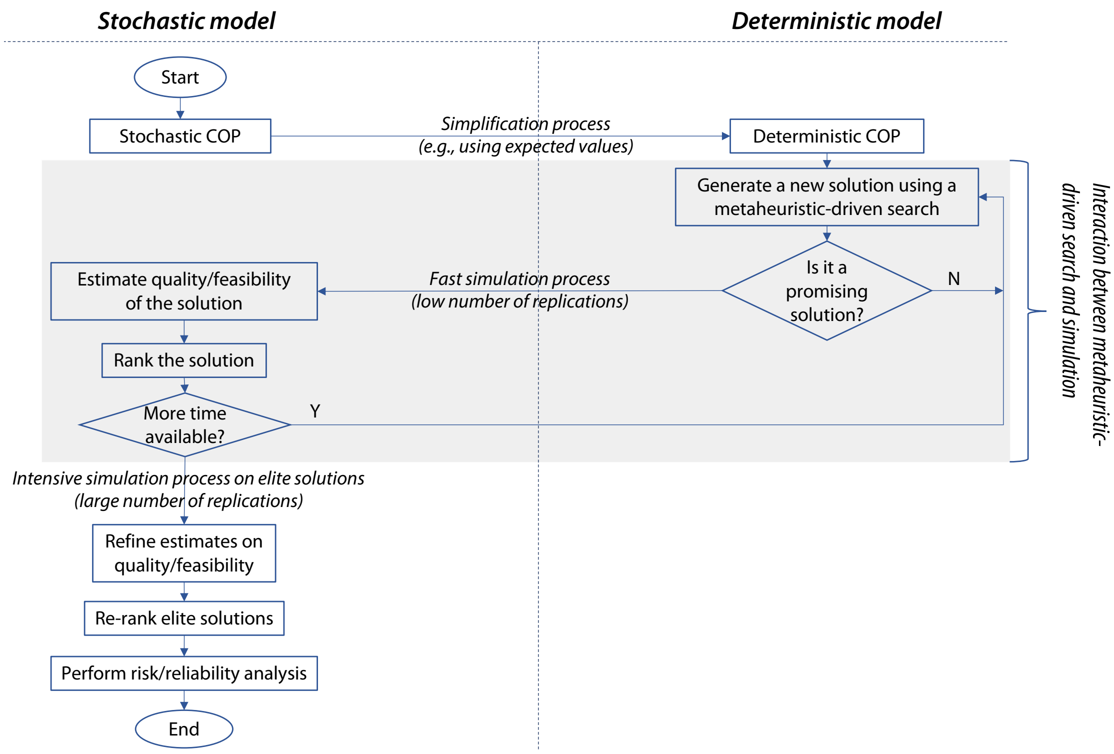


Figure 2.3: General scheme of simheuristics, adapted from [86]

In most practical applications, with moderate variance scenarios, it is safe to assume for simheuristic algorithms that the best solutions for the deterministic COP are probably the best solutions for the analogous stochastic one, without implying that is mandatory. Notice that, in extreme uncertainty scenarios, optimization methods should not be applied, as individual results can be remarkably disparate. Thus, given this 'relationship assumption' [86], a few high-quality solutions from the deterministic COP can be generated to effectuate several 'promising' solutions for the corresponding stochastic problem. Given a stochastic COP instance, a simplification process, as delineated in Figure 2.3, considers the corresponding deterministic problem, *e.g.*, by using expected values. Then, a metaheuristic-driven algorithm tries efficiently to find feasible, high-quality solutions for the deterministic COP. The simheuristic algorithm estimates or assesses the feasibility and the quality of the 'promising' solutions using simulation, fuzzy logic, or dynamic programming methods. However, the simulation method should be preferred, as, in this method, a best-fit theoretical or empirical probability distribution can pattern randomness instead of taking normal or exponential behavior, like other methods [86]. Additionally, a reduced number of replications are performed during the simulation to allow metaheuristic to perform an intensive search. A ranked list of the stochastic problem's elite solutions is keeped by using the estimated values of the simulation. Depending on the simulation's estimated values, metaheuristic can intensify exploration of promising searching areas. Larger number of simulations can be performed, after the metaheuristic's maximum allowed time is over, for a reduced batch of elite solutions, and, consequently, the new estimates can be used to re-rank the solutions.

Notice that additional information on the probability distribution of each solution's quality can be extracted from the final step's long simulations. Then, the decision-making procedure can include reliability/risk analysis criteria by using the above corresponding information. Thus, decision-makers may also be interested in analyzing the values' probability distribution with similar expected values to choose a lower risk solution over a risky, better-expected value [86]. Consequently, one of the critical simheuristics' advantages is the risk analysis competency, as metaheuristics can generate a plethora of high-quality alternative solutions and as simulation can provide a random sampling of observations for each proposed solution.

To solve stochastic COPs, metaheuristics, considered primarily to deal with deterministic problems, can be extended using simulation. Simheuristics is an instigating alternative to combined metaheuristics-simulation methods as they can provide near-optimal and high-quality solutions to stochastic and NP-hard problems in good computing times [86]. Additionally, obtaining an approximate solution to a real-world's precise model is better than taking the simplistic model's optimal solution.

Application areas of simheuristics include transportation and logistics [57, 83, 84, 87, 111, 117, 141], finance [101], healthcare [46], waste collection [56, 59], and cloud computing [96]. For real-world complex stochastic optimization problems, simheuristics should be considered a "first-resort" method [30], as it can handle reality in uncertain problems by simulation modeling, it can assess risk with ease, and a post-run simulation output analysis can be made. In this dissertation, particularly in Chapter 4, simulation drives the optimization in a sense that, after adaptive sampling, each candidate solution is evaluated, and the optimization algorithm can drive the search process.

2.7 Variable Neighborhood Search

The VNS proposed by Mladenovic and Hansen [14, 68, 69, 97, 130] is an efficient and straightforward metaheuristic method for solving various types of combinatorial optimization problems. Examines remote areas, or neighborhoods, of the current solution and moves to a new solution if and only if the objective function improves. Local search methods are applied repeatedly to find solutions in the neighborhood near the local. VNS is designed to approach solutions to discrete and continuous optimization problems and, aims to solve linear problems, integer programming problems, mixed-integer problems, nonlinear problems, and similar ones. VNS systematically changes the search into two phases. A descent is first performed to find a local optimal and finally a phase of successive perturbations to escape from the respective valley and explore more remote areas. VNS is based on the following concepts [65]:

- The local minimum concerning one neighborhood structure is not necessarily a local minimum in another neighborhood structure.
- A global minimum is a local minimum concerning all possible neighborhood structures.

- In many problems, the local minima concerning one or more neighborhoods are relatively close to each other.

The local search uses heuristic algorithms such as *2-opt*[1] heuristic. The goal is to improve the existing solution through a set of rules that define the respective neighborhood. Thus arises the set of permissible steps from which we choose the solution that presents the greatest improvement over the existing one. If no improvement is found, the algorithm is trapped in a local optimum and can not take any steps. VNS method aims to release these local optimal [91]. Therefore, in addition to providing very good solutions, often in simpler ways than other methods, VNS shows us due to its very good performance, which, in turn, can lead to more efficient and more sophisticated implementations. VNS's scope of applications is growing rapidly in number and covers many areas such as integral programming, vehicle routing, network design, location, artificial intelligence, engineering, concentration problems, biology, telecommunications design. Additionally, the use of VNS in a simheuristic has been used in many recent projects [57, 58, 101, 110].

2.7.1 Basic Variable Neighborhood Search

The Basic Variable Neighborhood Search (BVNS) takes as input an initial solution x the number of k_{max} neighborhoods available, and the termination criterion that usually in this type of algorithm is a time limit that we allow the algorithm to return the best improvement has been found within this time frame. k first takes the value 1, which means that we start from the first neighborhood we have defined. Then shaking takes place during which some random, but in any case, proper steps are taken to escape from the local best and explore other solutions. The algorithm attempts to apply the method of better improvement based on the neighborhood k in which we are. If no improvement is found, k

[1] Croes [35] first proposed the *2-opt* technique where he defines it as an optimized solution for the Traveling Salesman Problem (TSP) for both symmetric and asymmetric instances, the latter requiring more work. There is no guarantee that this technique will find a global optimum answer; instead, as it is a heuristic, the returning answer is usually said to be 2-optimal. It aims to gradually improve an initially given, feasible answer, local search, until it reaches a local optimum and no more improvements, called "inversions", can be made.

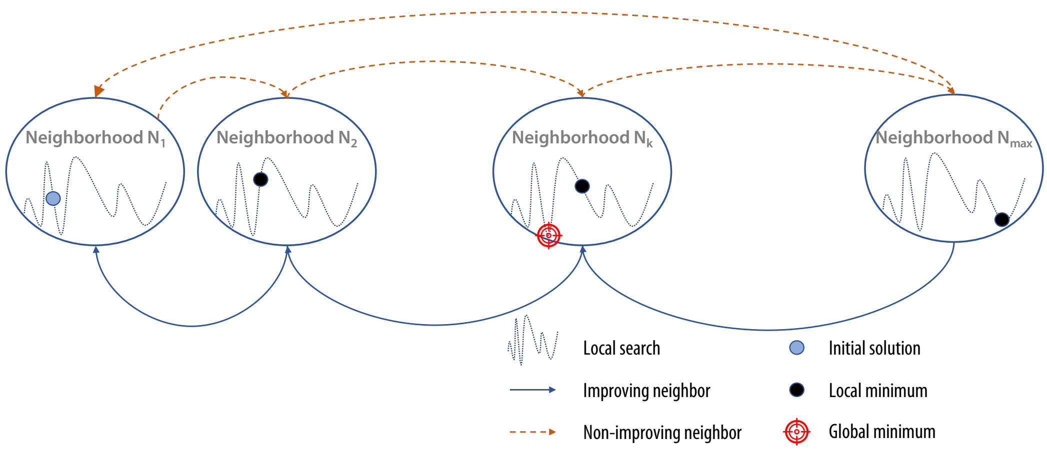


Figure 2.4: Schematic representation of the BVNS.

increases by 1, *i.e.*, we go to the next available neighborhood, starting from the beginning, making a disturbance and optimization based on this neighborhood. If an improvement is found, then k becomes 1 again, *i.e.*, we return to the first neighborhood and start from the beginning. This procedure continues until we reach the time limit set from the beginning and the best solution found up to that point is returned. BVNS is described in Algorithm 1, and a visual representation of its operation is given in Figure 2.4.

2.7.2 Variable Neighborhood Descent

The Variable Neighborhood Descent (VND) is an optimization method that deterministically combines many neighborhoods to find better solutions to our problem. Like BVNS, it starts by looking at optimization in the first neighborhood, and if no improvement is found, it goes to the next one. If it finds improvement in a neighborhood, it makes the step required for improvement and returns to the first neighborhood. The difference with the BVNS is that the perturbation process is not interfered with, and thus the VND is a completely deterministic methodology with stops when the termination criterion is met, which is not the

Algorithm 1 Steps of the BVNS.

Require: x : initial solution
k_{max} : number of neighborhoods
t_{max} : maximum CPU time allowed

repeat
  $k \leftarrow 1$
  repeat
    $x' \leftarrow \textsc{Shake}(x, k)$ ▷ *Shaking*
    $x'' \leftarrow \textsc{BestImprovement}(x')$ ▷ *Local search*
    $\textsc{NeighborhoodChange}(x, x'', k)$ ▷ *Change neighborhood*
  until $k = k_{max}$
  $t \leftarrow \textsc{CpuTime}()$
until $t > t_{max}$

execution time but not the execution time finding any improvement [68]. VND is described in the Algorithm 2.

Algorithm 2 Steps of the basic VND.

Require: x : initial solution
ℓ_{max} : number of neighborhoods

$\ell \leftarrow 1$
repeat
  $x' \leftarrow \arg\min_{y \in N_\ell(x)} f(x)$ ▷ *Find the best neighbor in* $N_\ell(x)$
  $\textsc{NeighborhoodChange}(x, x', \ell)$ ▷ *Change neighborhood*
until $\ell = \ell_{max}$

2.7.3 General Variable Neighborhood Search

The General Variable Neighborhood Search (GVNS) is the method that results if we replace the BestImprovement optimization step in the BVNS, described in Section 2.7.1, using the VND algorithm, described in Section 2.7.2. This way, we will have more efficient but maybe slower optimization. In each iteration, the current candidate solution is optimized not by using a neighborhood but by the VND, which does more in-depth optimization by considering a sufficient number of neighborhoods before stopping. Algorithm 3 describes the GVNS method [67].

Algorithm 3 Steps of the GVNS.

Require: x : initial solution
k_{max} : number of VNS neighborhoods
ℓ_{max} : number of local search neighborhoods
t_{max} : maximum CPU time allowed

```
repeat
    k ← 1
    repeat
        x' ← Shake(x, k)                              ▷ Shaking
        x'' ← VND(x', ℓ_max)                          ▷ Local search
        NeighborhoodChange(x, x'', k)                 ▷ Change neighborhood
    until k = k_max
    t ← CpuTime()
until t > t_max
```

2.7.4 Reduced Variable Neighborhood Search

The Reduced Variable Neighborhood Search (RVNS) is a variant that is more popular when dealing with substantial problems [66]. In problems of this magnitude, the optimization phase will undoubtedly cost us a lot of computing time, and we will not be able to find a sufficient number of areas of the total space of feasible solutions due to the time constraints we place on the various variants of VNS algorithms. For this reason, the RVNS lacks the optimization phase. It starts with an initial solution either randomly or using a construction heuristic algorithm. Then, as in BVNS, described in Section 2.7.1, the shake is done, but there is no optimization, resulting in a series of random movements. In each repetition of the shake, the necessary check is made for whether any improvement was found. The procedure is described in Algorithm 4.

2.7.5 Skewed Variable Neighborhood Search

This variant aims to explore very remote areas within the space of feasible solutions. Skewed Variable Neighborhood Search (SVNS) [68] measures the distance between two candidate solutions with a measurement function $\rho(x, x')$ and the help of parameter α, which will allow the jump to other more remote areas with the appropriate value selection. Especially in the VRPs in which we start the

Algorithm 4 Steps of the RVNS.

Require: x : initial solution
k_{max} : number of neighborhoods
t_{max} : maximum CPU time allowed

```
repeat
    k ← 1
    repeat
        x′ ← Shake(x, k)                                  ▷ Shaking
        NeighborhoodChange(x, x″, k)                      ▷ Change neighborhood
    until k = k_max
    t ← CpuTime()
until t > t_max
```

exploration having a quality initial solution and not some random one, this method may not give good results since the optimal solution of the problem will not be in a very remote area but will have several standard features with our initial solution. Algorithm 5 describes the SVNS.

Algorithm 5 Steps of the SVNS.

Require: x : initial solution
k_{max} : number of VNS neighborhoods
α : parameter
t_{max} : maximum CPU time allowed

```
repeat
    k ← 1
    repeat
        x′ ← Shake(x, k)                                  ▷ Shaking
        x″ ← FirstImprovement(x′)                         ▷ First descent heuristic
        if f(x′) < f(x) then                              ▷ Keep the best of the solutions x, and x′
            x ← x′
        if f(x″) − αρ(x, x″) < f(x) then                  ▷ Change neighborhood
            x ← x″
            k ← 1
        else
            k ← k + 1
    until k = k_max
    t ← CpuTime()
until t > t_max
```

2.8 Failure Patterns in Smart Meters' Communication

Despite the fact that learning failure patterns in the communication of smart meters are a definite concern of grid operators, as of today, this problem is understudied. However, lots of work investigates how failures in smart grids can be efficiently detected. Calderaro *et al.* [27] proposed a model for performing fault diagnosis despite erroneous data transmissions. Equally important is the approach of Jiang *et al.* [81] that suggested a data-driven method for fault detection, identification, and location in smart grids. Moreover, Jiang and Zeng [76] proposed a probabilistic graphical model to detect line outages in power grids. In our approach, described in Chapter 5, we utilize machine learning and individual profiles to successfully classify the smart meters' behaviour.

Machine learning techniques have been widely applied to solve smart grid problems. Kher *et al.* [88] suggested a machine learning model to monitor suspicious activities or malicious attacks. They proposed leveraging the real-time data collected from the grid. By the same token, Valdes *et al.* [139] suggested to use machine learning to distinguish normal, fault, and attack states in an electrical substation, by profiling the customer consumption per day. Similarly, Hartmann *et al.* [75] also proposed to use machine learning, to profile smart meters' behaviour. In their approach, they suggested to use Gaussian mixture models to detect non-technical losses in real-time. In like manner, in the domain of intrusion detection systems, to learn the standard behaviour of data transfers in a network and to find discrepancies that suggest an intrusion, Berthier *et al.* [15] proposed a machine learning technique.

Furthermore, Ruzelli *et al.* [122] suggested that the machine behaviour and disturbance in the environment can be learnt and stored in profiles. These profiles are then used to create signatures of electrical appliances to detect their operating status. Correspondingly, in our work, we suggest creating profiles to identify the normal communication behaviour of smart meters. In addition to the multivariate profiles approach, firstly used on [75], our implementation uses ND-trees to encode the failure patterns and verify if the new behaviour is expected or not, based on the computed profile.

ND-trees are mainly used to search through non-ordered data spaces[29, 109]. Moreover, Kobe *et al.* [89] proposed to repurpose ND-trees for k-nearest neighbour search. In our approach, we choose the ND-tree technique as it can quickly search the huge amount of historical data, allowing the profile to be calculated in near real time.

Part II

Enhancing Smart Grid Resilience and Reliability

3 Deterministic Overloading Prevention Problem

> No problem can be solved from the same level of consciousness that created it.
>
> *(Albert Einstein)*

Given the topology and the energy data of a microgrid at the current time, the potential overloading incidents are detected, and the optimal countermeasures are calculated for the next measurement interval. The grid operators can apply the proposed actions to recover the grid from the disturbance. Into this chapter, the problem is defined and formulated as a MO-MIQCP. This chapter also suggests a solution method using a combinatorial optimization approach with a state-of-the-art exact solver.
This chapter is based on work that has been published in the following paper:

- *"Preventing Overloading Incidents on Smart Grids: A Multiobjective Combinatorial Optimization Approach" (OLA 2020)*
  Antoniadis N., Cordy M., Sifaleras A., Le Traon Y.

Contents

3.1 Introduction

The so-called smart grid paradigm has been motivated by the need to manage today's power grids' growing complexity. It aims to keep up with growing energy demand, for example, by integrating renewable energy or providing innovative services, primarily driven by sensors and two-way communications between smart meters and electricity suppliers. Part of the smart grid power system [42] in Luxembourg [34], the low voltage distribution network, carries electricity from distribution transformers to smart meters of end customers. This low-voltage network is generally more complex and meshed than the medium-voltage one, and it is more difficult to trace its perturbations. Each distribution substation comprises a power supply system that delivers electricity to industrial and residential users through power supplies, *i.e.*, cabinets, and cables. Distribution cabinets control distributed power and provide overload protection to mains lines through their fuses. A smart meter is installed between the service cable and each user's installation to measure power consumption and manage loads through their relay, triggering the function. The number of connected components of the multigraph 1.2 is equal to the number of distribution substations, which means that each service cable can only be connected to one substation precisely. All cables start from one fuse in one cabinet and end in another fuse in another cabinet. If the end cabinet of the cable does not have any cable starting from it, it is called *dead end*. Each fuse's status can be opened or closed; this information, combined with the network topology, can be used to determine the accessibility of each cable in the network from each substation. The current load of each cable can be approximated using methods such as presented in [74], by summarizing the production and consumption values of all users of this cable, that given through their smart meters. Consequently, a cable's load percentage is obtained by dividing its current load by its maximum amplitude multiplied by one hundred. Then the cable runs the risk of *overload* if its current load exceeds a predefined threshold.

3.1.1 Preventing an overloading incident

Grid operators typically consider a risk of overloading incident when the current load percentage on a cable exceeds a predefined threshold set by the grid operator. Then, they can apply different countermeasures to reduce cable loads, thereby avoiding overloading to occur. The preferred solution consists of limiting the over-production remotely, *e.g.*, solar panels on a sunny day, or over-consumption of specific users, *e.g.*, charging EVs; this countermeasure is commonly named *load curtailment* [131]. However, some users have contracts that prevent the operator from regulating their power capacity. Therefore, curtailment is not, in such cases, an option. More generally, if curtailments cannot result in a stable state, *i.e.*, without risk of overloading, the operators have to reconfigure the topology of the grid by switching fuses, using the *intertrip* [20] method to shift reserves from one network to another, even if intertrip is complicated for the meshed low-voltage network [20].

Changing fuse states require technicians to visit the corresponding cabinets physically. Therefore, minimizing the number of visiting cabinets is an object of considerable solicitude to the grid operator to minimize a potential incident's restoration time. Another concern is the minimization of the number of fuses that have to be switched on or off, as the grid's configuration should remain nearly the same as its initial state.

Avoid disconnecting users, especially critical ones, such as patients, is a matter of great concern to the grid operators. Still, this may happen as a last resort to prevent cascading overloads [20] and to avoid any damage to the power line when there is an insufficient operating reserve. In this case, the number of disconnected users should remain minimal.

3.1.2 Contribution

Given the above requirements, finding the ideal solution(s) to prevent overloading incident is a daunting decision-making task that humans can hardly solve without support. Therefore, in this chapter, we propose a multiobjective combinatorial optimization approach to define, model, and solve the overloading prevention problem for a low-voltage network. Our approach can also model a medium-voltage smart grid or a standalone microgrid with minimal changes. Mathematical optimization methods have been successfully applied to solve a wide range of decision problems [129], including in the energy industry, see Section 2.2).

Given the physical network data, *i.e.*, substations, cabinets, cables, connections between cabinets, that are assumed to remain constant, the initial state of the fuses, and the power values from the users' smart meters, we approximate the current load percentage on each cable by solving a linear system described in [74]. To create the matrices defining this linear system, we compute the reachable cables from every substation based on the fuses' state and the physical network data. We also detect parallel cables, *i.e.*, multiple edges in the grid's multigraph, since computations involving those are slightly different.

Once the risk of overload is detected, *i.e.*, the approximated current load percentage exceeds the predefined threshold; we store the fuses' current states and the smart meter values. Then we solve our optimization model to suggest the most appropriate countermeasures. Curtailment of compliant users is first attempted.

If this curtailment cannot establish a stable state, we should take the second action to switch fuses on or off. On every possible change of fuses' state, a new linear system has to be defined and solved to approximate the current loads on the cables. Moreover, simultaneously connecting multiple substations should be avoided, as we cannot calculate the power flow cycles between substations; otherwise, the load calculation could return a wrong result [74]. In the end, our solution aims to maximize the number of connected users while minimizing the number of visited cabinets and the number of changes applied to fuses.

We evaluate the applicability of our approach through a benchmark set comprising ten grid topologies for five substations; similar to an area of a small village in Luxembourg, and another set containing a gradually increasing number of substations, by steps of five, from ten to fifty; similar to an area of a medium-size city in Luxembourg. The topologies are generated by a tool we developed based on Creos Luxembourg S.A.'s real-world statistics, the only grid operator in Luxembourg and our project partner. Our results show that our approach can suggest solutions for all topologies in due time, up to about 15 min. Moreover, a detailed analysis of the curtailed and disconnected users reveals that curtailment alone is not enough to prevent overloading incidents, emphasizing the need for automated solutions to reconfigure the grid and more sophisticated demand response programs. The remainder of this chapter is structured as follows. Section 3.2 provides the mathematical model for this work. Then, in Section 3.3, we detail the implementation of our proposed solution method, which is evaluated in Section 3.4. Finally, we conclude in Section 3.5.

3.2 Mathematical Model

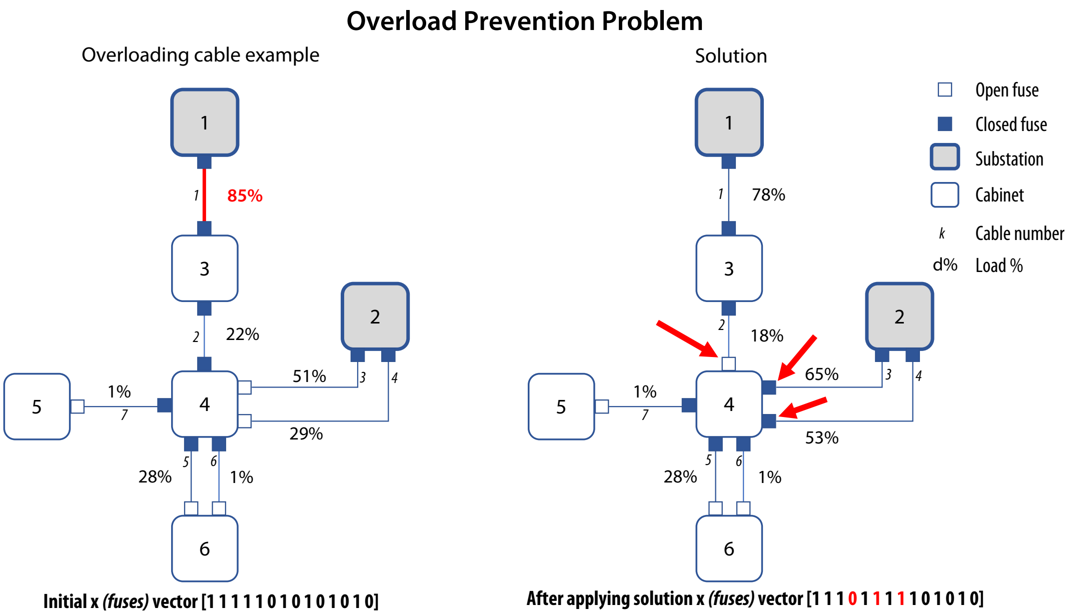


Figure 3.1: Overload Prevention Problem example.

The OPP can be defined on a complete undirected multigraph $G = (V, E)$. The set $V = \{1, 2, \ldots, o\}$ is the vertex set, *i.e.*, the set of the cabinets of the grid, $E = \{(i, j) \in V^2, i \neq j\}$ is the multiple edge set, *i.e.*, the multiset of the cables that connect the cabinets of the grid. An example of the problem is illustrated in Figure 3.1, while in Figure 3.2 the associated multigraph of the example is pre-

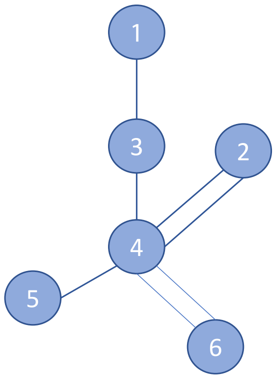


Figure 3.2: Associated multigraph of the example.

sented. The problem, we previously described, can be modeled as a Mixed Integer Quadratically Constraint Program (MIQCP) formulation as follows:

$$\max \sum_{i=1}^{n} r_i \sum_{k=1}^{m} uc_{ki} \tag{3.1}$$

$$\min \sum_{b=1}^{o} dfcab_b \tag{3.2}$$

$$\min \sum_{f=1}^{2n} |x_f - x_f^0| \tag{3.3}$$

subject to:

$$A \cdot wp = P \tag{3.4}$$

$$A \cdot wq = Q \tag{3.5}$$

$$l_i < \lambda, \forall i \in \{1, \dots, n\} \tag{3.6}$$

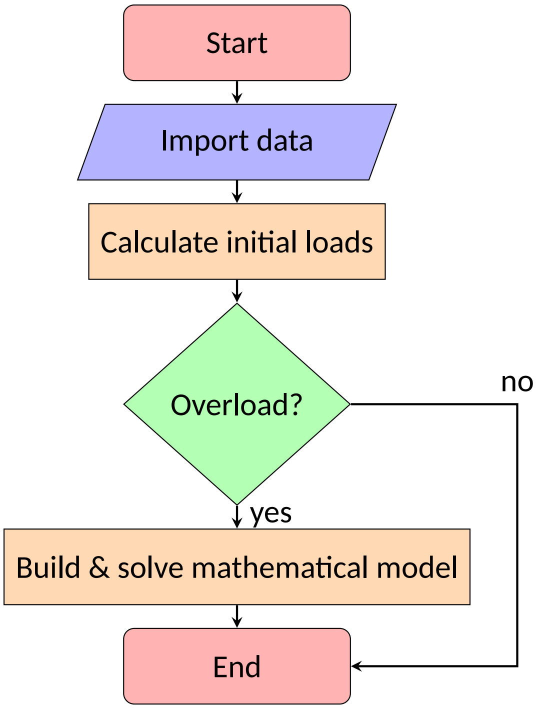


Figure 3.3: Outline of the solving procedure.

Given G, the first objective, Equation 3.1, defines the fuses' state to maximize the serviced users of the grid. Simultaneously, the second objective, Equation 3.2, sets the state of each fuse to minimize the number of visiting cabinets. According to Creos Luxembourg SA, the cost of reconfiguration is nearly analogous to the number of the cabinets the technicians have to visit. The third objective, Equation 3.3, minimizes the number of fuses' changes to keep the initial fuses' state as much as possible. An outline of the solving procedure is presented in Figure 3.3.

Curtailment policy to the users is applied when any producer or consumer has amperage over I_{LP} and I_{LC}, respectively. Equation 3.4 and Equation 3.5 approximate the current loads, as in [74]. To avoid overload cables, the constraint described in Equation 3.6 limit the current load percentage on each cable under the predefined threshold. The notation used is presented in the Nomenclature .

3.3 Implementation

As the problem above is formulated as a MIQCP, a state-of-the-art mathematical programming solver, Gurobi [61], is chosen to address it. Smart grid data are imported to our program, and a pre-computational phase is taking place. Vectors, substations, cabinets, and the edges, cables are stored, and the multigraph of the smart grid is created. Additionally, the initial fuses' states, the smart meters, their connecting cables, and their consumption and production values are being read and stored. Moreover, the entire set of cycles of the multigraph [80, 105] are being found, eliminating any connections between substations and investigating any multi-edges, multiple cables between cabinets, on the graph. During this pre-processing phase, the dead-ends cabinets are also defined to help us compute the load, using the depth-first-search algorithm [136], and stored. Having this information about the topology, we construct the potential linear equations assuming that all the fuses are closed. This phase ends by calculating the loads [74] by using Singular Value Decomposition (SVN) [53] for solving the over-determined linear system of equations and check if the initial state has any overloaded cables or not. If an overload is inspected, then the variables are being initialized and, using the Depth First Search (DFS) algorithm [136], the reachability vector r is constructed. After the reachability cable state is initialized, we can create the actual linear equations, the cable, cabinet, dead-end, and circle ones [74].

3.3.1 Linear transformation

As Gurobi does not support quadratic equality constraints, we need to transform the constraints, expressed by the Equation 3.4 and Equation 3.5 into a linear form. Firstly, we rewrite the constraints, expressed by the Equation 3.4 and Equation 3.5 as:

$$P_j = \sum_{f=1}^{2n} A_{jf} w p_f, \forall j \in \{1, \dots, leq\} \tag{3.7}$$

$$Q_j = \sum_{f=1}^{2n} A_{jf} wq_f, \forall j \in \{1, \dots, leq\} \tag{3.8}$$

We introduce, for each quadratic term in the above summations, new variables $zp_{jf} = A_{jf} wp_f$ and $zq_{jf} = A_{jf} wq_f$. As $A_{jf} \in \{-1, 0, 1\}$:

$$zp_{jf} = \begin{cases} -wp_f, & A_{jf} = -1 \\ 0, & A_{jf} = 0 \\ wp_f, & A_{jf} = 1 \end{cases} \tag{3.9}$$

$$zq_{jf} = \begin{cases} -wq_f, & A_{jf} = -1 \\ 0, & A_{jf} = 0 \\ wq_f, & A_{jf} = 1 \end{cases} \tag{3.10}$$

Using the Equation 3.9 and the Equation 3.10 we can rewrite the Equation 3.7 and the Equation 3.8 as:

$$P_j = \sum_{f=1}^{2n} zp_{jf}, \forall j \in \{1, \dots, leq\} \tag{3.11}$$

$$Q_j = \sum_{f=1}^{2n} zq_{jf}, \forall j \in \{1, \dots, leq\} \tag{3.12}$$

To be able to compute the zp_{jf} and zq_{jf}, we need to binary transform the above piecewise functions using indicator constraints [21]. Thus, for every coefficient matrix element, we introduce three additional variables as:

$$-1y_{jf1} + 0y_{jf2} + 1y_{jf3} = A_{jf}, \tag{3.13}$$

$$y_{jf1} + y_{jf2} + y_{jf3} = 1 \tag{3.14}$$

$$\forall j \in \{1, \dots, leq\}, \forall f \in \{1, \dots, 2n\}, y_{jf1}, y_{jf2}, y_{jf3} \in \{0, 1\}$$

In Equation 3.13 it is ensured that A_{jf} can only take values from its domain where the Equation 3.14 ensures that, only one variable could take value one. Using the Equations 3.13 and the Equation 3.14, the Equation 3.9 and the Equation 3.10 become:

$$zp_{jf} = \begin{cases} -wp_f, & y_{jf1} = 1 \\ 0, & y_{jf2} = 1 \\ wp_f, & y_{jf3} = 1 \end{cases} \tag{3.15}$$

$$zq_{jf} = \begin{cases} -wq_f, & y_{jf1} = 1 \\ 0, & y_{jf2} = 1 \\ wq_f, & y_{jf3} = 1 \end{cases} \tag{3.16}$$

3.3.2 Solving model

The final step is to calculate the difference between the initial and the current state of each fuse. Moreover, the binary cabinet visit indicator for each cabinet is computed. To solve the model, we are using the lexicographic approach [22] for the objectives to reach any Pareto optimal solution by assigning a priority to each objective and then optimizing the objectives in decreasing priority order. At each step, the current objective is optimized, and a constraint is introduced to guarantee that the higher-priority objective functions preserve their optimal value [22, 61]. We are specifying a complete order of importance along with our partner, Creos Luxembourg SA. After getting the preference information, our first objective, Equation 3.1, has the highest importance, the second one, Equation 3.2, has lower importance and, the third one, Equation (3.3), has the least importance.

3.4 Evaluation

Our method has to provide solutions to the overloading prevention problem sufficiently fast in order to be practicable. According to our partner Creos Luxembourg SA, the computation time should not exceed 15 minutes (which corresponds to the interval of time between two smart meter data reports). Hence, our first research question concerns the scalability of our approach concerning increasingly-large grids.

Our primary focus is to analyze the presented solution qualitatively: how much our approach manages to satisfy the requirements of not disconnecting, if possible, the users. The absolute numbers, of course, depend on the particular cases considered. Therefore, our second research question concerns a relative analysis: how well different curtailment policies allow avoiding user disconnections in different overload scenarios.

3.4.1 Dataset and experimental setup

A topology generation software tool to evaluate our proposed method was first developed. Using this tool, we create ten realistic smart grid graphs based on real topology data. In each instance, we consider five substations to answer the second research question, topologies that resemble a small village in Luxembourg. For every grid graph, we consider 216 scenarios[1] as a combination of different percentage of overload producers, overload consumers, producers, and consumers that can be curtailed. Moreover, as we do not notice a significant difference when changing the percentage of overloading and curtailment in timing, we study the scalability only concerning the grid size by creating another nine realistic smart grid graphs. Each instance contains a gradually increasing number of substations by five, from ten to fifty.

On these graphs, three to four cabinets are connected on each substation, where the number of cabinets is uniformly random. Two of these cabinets are connected by two edges (cables) to the substation. Under the first level of cabinets, three to five cabinets are connected, where the number of cabinets is uniformly random. Under the second level of cabinets, zero to two cabinets are connected, where the number of cabinets is uniformly random. During the experiments' creation, it is explicitly assumed that one cabinet, either on the second or the third level of the graph, is connected to another substation's cabinet so that intertrip can be applied. The material, size, and maximum ampacity are generated uniformly randomly from real data for each cable. On each cable, up to 21 smart meters are connected. The smart meters were sampled from a uniform discrete distribution with the range $[0, 21]$.

We analyzed the historical data we acquired from Creos Luxembourg SA to create consumption and production energy data. More specifically, we analyzed for the 215 consumers and the seven producers, the four electrical values from their smart meters, active energy consumption and production, and reactive energy consumption and production. The data consisted of 9 months of measurements, with 96 measurements per day. Mean and standard deviation and min-

[1] Interested readers may find all the presented results for the 216 instances from http://tiny.cc/ola2020_antoniadis

imum and maximum value for each user were computed to produce their consumption and production profiles. For each smart meter, a random profile is selected and, from the corresponding distribution, an electrical value is generated. Additionally, at most, 10% of the users are selected to produce energy. To create a different percentage of overloaded and curtailed users, we shuffle the producers and consumers vectors using the Fisher-Yates algorithm [40, 47]. Then, we pick the corresponding number of users from the shuffled vectors.

A *soft curtailment* [20] is applied if a producer overpasses the threshold of 60 A, *i.e.*, 80% of 75 A, the typical roof-top solar panel installation amperage, or if a consumer overpasses the threshold of 32 A, *i.e.*, 80% of 40 A, the typical amperage supplied by residential meters. If a producer or a consumer is picked for curtailment, its active energy is limited to 20 A; a value picked together with Creos Luxembourg SA. The experiments were conducted on a standard MacBook Pro with a 2.6 GHz Intel Core i7 processor, macOS Mojave 10.14.6 operating system, 16 GB 2133 MHz LPDDR3 memory using Java JDK 1.8.0-162 and Gurobi Optimizer 8.1.1 – Academic Version [61].

3.4.2 Results and discussion

In what follows, each experiment was run for ten times, for the ten different topologies, and the 216 different scenarios. The average time and the 99% confidence interval for the 21600 experiments (5 substations), was found to be equal to *6.363 sec ± 1.527 sec*.

We observe that our method can propose solutions quickly to help grid operators to prevent overloading incidents. To check if our method could be applied to a larger scale smart grid, we create and test nine different topologies, and the results of these experiments are shown in Figure 3.4. Indeed, even in the most complex case, that resembles the size of a medium-size city in Luxembourg, our approach finds a solution in about the allowed time (15 min).

Moreover, we notice that when the size of the graph doubles, the average computation time is approximately five times higher. For the second research ques-

[2] *POv*: overload producers' percentage, *COv*: overload consumers' percentage, *PCur*: curtailed producers' percentage, *CCur*: curtailed consumers' percentage, *ConU*: connected users percentage, *VisCab*: cabinets to visit percentage, *FusesCh*: fuses changed percentage

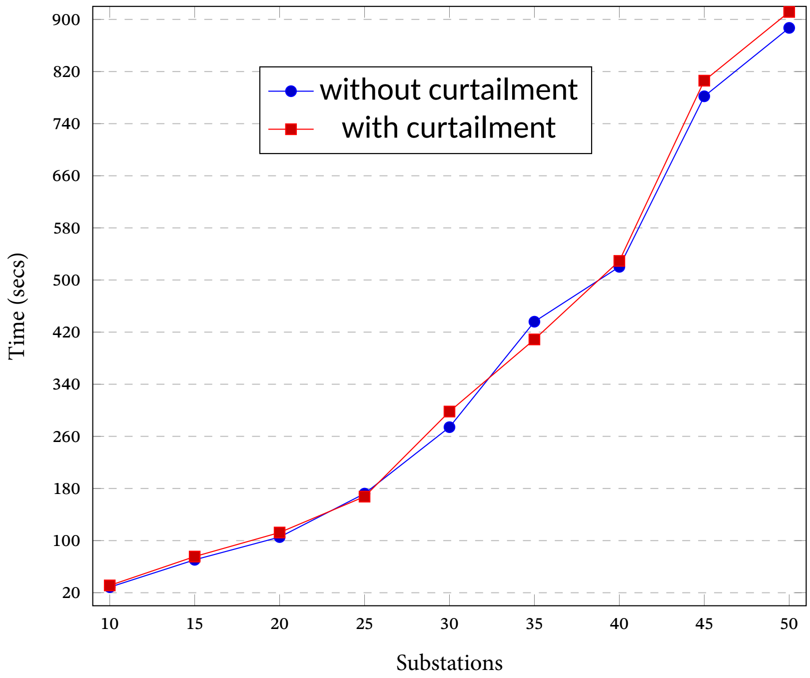


Figure 3.4: Different topologies and computational time.

POv[2]	COv[2]	PCur[2]	CCur[2]	ConU[2]	VisCab[2]	FusesCh[2]
0-25%	0-10%	100%	100%	100%	6.98%	3.47%
0%	25%	-	100%	99.96%	7.04%	3.5%
0%	50%	-	100%	95.07%	11.55%	5.5%
25%	25%	100%	100%	99.8%	7.18%	3.56%
10%	50%	50%	100%	94.7%	12%	5.78%
50%	10%	100%	100%	99.94%	7.02%	3.49%
10%	0%	0%	-	94.57%	10.75%	5.37%
10%	0%	50%	-	97.68%	8.47%	4.19%
50%	25%	100%	100%	99.35%	7.67%	3.76%
50%	25%	50%	50%	61.57%	31.6%	17.18%
50%	25%	0%	0%	41.77%	35.83%	23.06%

Table 3.1: Sample results of our method (5 substations).

tion, as shown in Table 3.1, and detailed in Appendix A, if the percentage of overloaded consumers remains at most 10%, while the percentage of overloaded producers remains at most 25%, and curtailment is applied for all the users of the grid, no disconnection is needed. Nonetheless, the percentages of cabinets to visit and changed fuses remain low, 6.98% and 3.47%, respectively. On the opposite, 5.43% ± 0.93% of the users should be disconnected to prevent overload if 10% of the producers are overloaded, and no curtailment policy is applied. From our findings, it is shown that curtailment policies lead to fewer disconnections to prevent overloads. Additionally, fewer cabinet visits and fewer changed fuses are needed, avoiding additional costs for the electrical companies while keeping the grid in a stable configured state as possible. Nevertheless, in the long term, electrical companies should increase their operational reserves to decrease the possibility of disconnections [20]. Moreover, solar panel producers should install batteries to minimize their losses due to the curtailment policies [20].

3.5 Conclusion

In this chapter, the definition of the OPP in smart grids and its formulation as a MO-MIQCP is presented, and a solution method using a state-of-the-art exact solver is suggested. It is shown that this approach can be included in the grid operator's decision-making process as it can successfully and rapidly help to prevent challenging overloading incidents in a smart grid of about the size of a medium city in Luxembourg, minimizing the disconnections of the grid's users.

Our method has been integrated into a grid visualization tool that allows operators to observe the grid cable states, detect (risk of) overloading incident, and call our algorithm to find appropriate countermeasures.

4 Reliability Analysis through Simulation and Stochastic Optimization

True optimization is the revolutionary contribution of modern research to decision processes.

(George Dantzig)

Into this chapter, reliability analysis through simulation is employed to evaluate the topology reconfiguration's robustness after a disturbance, like an overload. Then, in this chapter, the Single-State Stochastic Program (SSSP) is defined, where the optimal countermeasures are calculated for a measurement horizon, e.g., 24 h, and a simheuristic method is proposed to solve it.
This chapter is based on work that has been published in the following paper:

- *"A variable neighborhood search simheuristic algorithm for reliability optimization of smart grids under uncertainty." Manuscript submitted for publication. Antoniadis N., Cordy M., Sifaleras A., Le Traon Y.*

Contents

4.1 Introduction

The increased complexity of the energy grids motivates the idea of a smarter grid. There is a way that they can deal with the growing demand for energy and providing innovative services [6].

One of the most critical issues in power grids is overloading cables as they can harm distribution power lines. With fuse switches, an overload trips the fuse, causing the circuit to open and, consequently, stopping flow and heating. Grid operators usually assume that the load rate above a predefined level on a cable entails a significant overload chance. Long-term overloads, however, can also damage cables even within the security limits and may cause energy grids to malfunction [148].

Specific counteractions can then be applied to reduce cable loads, preventing overload, including the curtailment of over-production or over-usage by individual users.

Some consumers, however, have contracts that prohibit the operator from controlling its power capacity. In these cases, therefore, a restriction is not a choice. More generally, when such constraints do not effectuate a stable state, *i.e.*, without the possibility of a surge, operators need to reconfigure the grid topology by swapping fuses to transfer reserves between networks even when the intertrip for the meshed low voltage network is complicated.

Technicians, if remotely switching is not possible, have to physically visit the correct cabinets if the fuse states are to be changed. Therefore, the minimization of the number of visitor booths is an area of considerable concern for the grid operator, which minimizes the recovery time of a possible incident.

It is a matter of significant interest for grid operators to avoid disconnecting users, critical ones, such as hospitals. However, if there is an inadequate operating reserve, this will happen to prevent cascading overloads to avoid harm to the line as a last resort. In this case, there will remain a small number of disconnected users.

The above countermeasures, including user curtailments and reconfiguration of the grid's topology, are generally applied in a short time, *i.e.*, less than an hour. Although, even if the above counteractions are applied immediately after detecting a risky overloading incident, the recovery response solution could lead to another overloading incident, as the future demands are not known beforehand. Moreover, there is ongoing demand from the grid operators for more reliable smart grids towards the "self-healing" grid [134], in the sense of automatically respond to problems and minimize disturbances. Therefore, if the recovery response solution could be tested for its efficacy over the next day, the smart grid operators would be of great use to ensure the solution is as robust as possible.

4.1.1 Contribution

In this chapter, two methods capable of calculating a smart grid's short-term expected reliability after an overloading event are presented. Firstly, correspondingly to the above IEEE reliability indices, three customer-based, three cable-based, and two load-based reliability indices, estimating overloading incidents are presented. After a potential overload event, the optimal and sub-optimal solutions are calculated, as described in [6]. For every solution, using Monte Carlo Simulation (MCS), the eight reliability indices could be estimated. Along with the number of fuses' changes, a Unified Overload Index (UOI) is calculated as a normalized weighted sum of the values. By using different vectors of weights, the decision-maker could investigate different aspects of the reliability assessment.

As the simulation is not an optimization tool [86], we propose a SSSP to optimize the reconfiguration of the grid topology and to ensure the smart grid could remain stable for the next 24 h. We propose to solve it through the *simheuristics* method, which combines simulation with metaheuristics [118](see also 2.6) to solve Stochastic Combinatorial Optimization Problems (COPs). For the metaheuristic part, we use Variable Neighborhood Search (VNS), in which the intensification of search and the diversification of local optimum solution is based on the systematic change of neighborhoods (see also 2.7). Additionally, we apply the control variate reduction technique (see also 2.3.3) to the MCS to reduce the number of simulations.

The remainder of this chapter is structured as follows. Section 4.2 provides the mathematical model for the stochastic program of this work. Then, in Section 4.3, we detail the implementation of our proposed solution method, which is evaluated in Section 4.4. Finally, we conclude the chapter in Section 4.5.

4.2 Materials and Methods

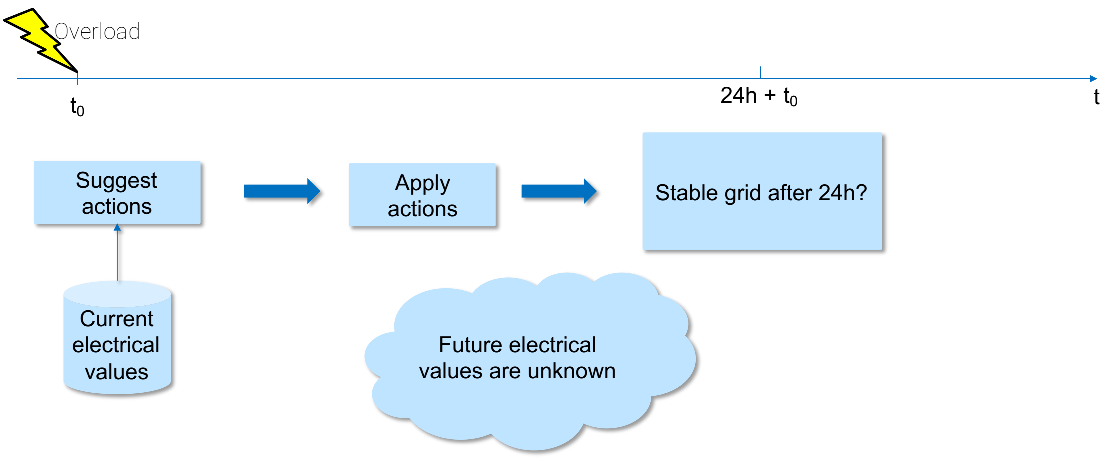


Figure 4.1: Check for stability over time.

The scope of our work is to predict the stability of a smart grid after a potential overload. The outline of the procedure is presented in Figure 4.1. From the given electrical values that have been read by the smart meters, if a potential overload has occurred, our proposed methods suggest the proper actions that have to be applied by the smart grid technicians. Our goal is to propose a robust solution for the next day under uncertainty, as the future electrical values are unknown. To do so, we propose two methods. The first one extends the work presented in [6] by including an MCS step for predicting the electrical values after a potential overload. As the simulation is not an optimization tool, in the second method, we use a single-stage stochastic program to optimize the grid topology's reconfiguration to ensure the smart grid could remain stable for the next 24 h by hybridizing simulation with a metaheuristic. To evaluate the solutions' quality, we propose a UOI as described in Section 4.2.1.

4.2.1 Unified Overload Index

In order to estimate the reliability after a potential overloading situation, we propose, in accordance with the state-of-the-art IEEE reliability indices [79] the following eight indices (see also Nomenclature):

- System Average Overload Frequency Index (SAOFI), like System Average Interruption Frequency Index (SAIFI) [79]. It expresses how often a *customer* experiences a sustained overload over a predefined period of time *(overloads/day)*.

$$SAOFI = \frac{co}{m} \tag{4.1}$$

$$co = \sum_{i=0}^{n} \sum_{t=0}^{T} fail_{it} cust_i \tag{4.2}$$

$$fail_{it} = \begin{cases} 1 & (u_{i(t-1)} = 0 \wedge u_{it} = 1 \wedge t > 0) \vee (u_{i0} = 1 \wedge t = 0) \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

$$u_{it} = \begin{cases} 1 & ld_{it} > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

- System Average Overload Duration Index (SAODI), like System Average Interruption Duration Index (SAIDI) [79]. It expresses the total duration of overload for a *customer* during a predefined period of time *(mins/day)*.

$$SAODI = \frac{cmo}{m} \tag{4.5}$$

- Customer Average Overload Duration Index (CAODI), like Customer Average Interruption Duration Index (CAIDI) [79]. It expresses the average time required to restore service on a *customer (mins/day)*.

$$CAODI = \frac{cmo}{co} \tag{4.6}$$

$$cmo = \sum_{i=0}^{n} \sum_{t=0}^{T} downt_{it} cust_i \tag{4.7}$$

$$downt_{it} = 15u_{it} \tag{4.8}$$

- Cable System Average Overload Frequency Index (CBLSAOFI), like SAIFI [79]. It expresses how often a *cable* experiences a sustained overload over a predefined period of time *(overloads/day)*.

$$CBLSAOFI = \frac{cblo}{n} \tag{4.9}$$

$$cblo = \sum_{i=0}^{n} \sum_{t=0}^{T} fail_{it} \tag{4.10}$$

- Cable System Average Overload Duration Index (CBLSAODI), like SAODI [79]. It expresses the total duration of overload for a *cable* during a predefined period of time *(mins/day)*.

$$CBLSAODI = \frac{cblmo}{n} \tag{4.11}$$

$$cblmo = \sum_{i=0}^{n} \sum_{t=0}^{T} downt_{it} \tag{4.12}$$

- Cable Average Overload Duration Index (CBLAODI), like CAODI [79]. It expresses the average time required to restore service on a *cable (mins/day)*.

$$CBLAODI = \frac{cblmo}{cblo} \tag{4.13}$$

- Average System Overload Frequency Index (ASOFI), like Average System Interruption Frequency Index (ASIFI) [79]. It is similar to SAOFI but it is based on load *(overloads/day)*.

$$ASOFI = \frac{lo}{lt} \tag{4.14}$$

$$lo = \sum_{i=0}^{n} \sum_{t=0}^{T} fail_{it} ld_{it} \tag{4.15}$$

$$lt = \sum_{i=0}^{n} \sum_{t=0}^{T} ld_{it} \tag{4.16}$$

- Average System Overload Duration Index (ASODI), like Average System Interruption Duration Index (ASIDI) [79]. It is similar to SAODI but it is based on load *(mins/day)*.

$$ASODI = \frac{lmo}{lt} \tag{4.17}$$

$$lmo = \sum_{i=0}^{n} \sum_{t=0}^{T} downt_{it} ld_{it} \tag{4.18}$$

In accordance with our industrial partner Creos Luxembourg SA, the weighted mean of the above indices is defined as the UOI. UOI can be used as a standard metric for the decision-makers to help them understand the impact of every proposed solution to the grid.

$$\begin{aligned} UOI = {} & w_{SAOFI}\frac{SAOFI}{\frac{T}{2}+1} + w_{SAODI}\frac{SAODI}{15(T+1)} + w_{CAODI}\frac{CAODI}{15(T+1)} \\ & + w_{CBLSAOFI}\frac{CBLSAOFI}{\frac{T}{2}+1} + w_{CBLSAODI}\frac{CBLSAODI}{15(T+1)} + w_{CBLAODI}\frac{CBLAODI}{15(T+1)} \\ & + w_{ASOFI}\frac{ASOFI}{\frac{T}{2}+1} + w_{ASODI}\frac{ASODI}{15(T+1)} \end{aligned} \tag{4.19}$$

$$w_{SAOFI} + w_{SAODI} + w_{CAODI} + w_{CBLSAOFI} + w_{CBLSAODI} + w_{CBLAODI} + w_{ASOFI} + w_{ASODI} = 1 \quad (4.20)$$

4.2.2 Simulation approach

In our first method, after a potential overload, at most five best solutions are picked from the method suggested in [6], and presented in Chapter 3, the proposed actions are applied, and the future electrical values are simulated, from the corresponding historical Gaussian distribution, for the next 24 hours. For every solution, using MCS, the eight reliability indices are estimated. During the MCS procedure, the *control variates* variance reduction method (see also Section 2.3.3) is applied by using the geometric mean of the UOI as the control variate, as shown in Equation 4.23. The UOI is calculated as a normalized weighted sum of the estimated values and the number of fuses' changes, as presented in Equation 4.22. The quality of each proposed solution is denoted by the calculated UOI (see also Section 2.3). In Figure 4.2 the outline of the proposed method is presented.

4.2.3 Simheuristic approach

Instead of hoping to find a good solution, after applying the proposed actions, in our second method, using the UOI, we try to find the actions to increase the expected reliability for the next 24 hours. The UOI is calculated as a normalized weighted mean of the aforementioned reliability indices, as presented in Section 4.2.1. If we denote the random electrical data that is available only after the decision is made with ξ, we need to minimize the random UOI function $UOI(x, \xi)$. Since we cannot directly minimize $UOI(x, \xi)$, we alternatively minimize the expected value, $\mathbb{E}[UOI(x, \xi)]$. The notation used is presented in the Nomenclature , while the corresponding single-stage stochastic optimization problem [128] becomes:

$$\min \mathbb{E}[UOI(x, \xi)] \quad (4.21)$$

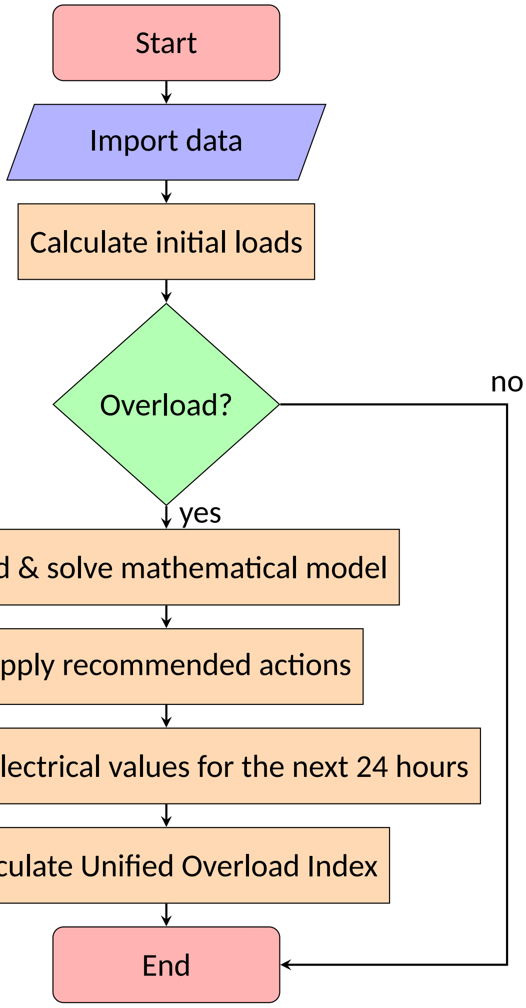


Figure 4.2: Outline of the simulation procedure.

If in equation (4.20) we set all the weights equal to $\frac{1}{8}$, UOI is the arithmetic mean of the normalized indices:

$$UOI^a = \frac{1}{8}\Bigg[\frac{SAOFI}{\frac{T}{2}+1} + \frac{SAODI}{15(T+1)} + \frac{CAODI}{15(T+1)} + \frac{CBLSAOFI}{\frac{T}{2}+1} + \frac{CBLSAODI}{15(T+1)} + \frac{CBLAODI}{15(T+1)} + \frac{ASOFI}{\frac{T}{2}+1} + \frac{ASODI}{15(T+1)}\Bigg] \tag{4.22}$$

We can also calculate UOI as the geometric mean of the normalized indices as:

$$UOI^{g} = \left[\frac{SAOFI}{\frac{T}{2}+1} \cdot \frac{SAODI}{15(T+1)} \cdot \frac{CAODI}{15(T+1)} \cdot \frac{CBLSAOFI}{\frac{T}{2}+1} \cdot \frac{CBLSAODI}{15(T+1)} \cdot \frac{CBLAODI}{15(T+1)} \right. \\ \left. \cdot \frac{ASOFI}{\frac{T}{2}+1} \cdot \frac{ASODI}{15(T+1)} \right]^{\frac{1}{8}} \tag{4.23}$$

We suppose that the random variable ξ, which represents the realizations of the electrical values, has a given Gaussian distribution, *i.e.*, it takes values $\xi_1, ..., \xi_K$, with respective probabilities $p_1, ..., p_K$, where the K considered scenarios represent historical data. In the case of finitely many scenarios, it is possible to model the stochastic program as a deterministic optimization problem by writing the expected value $\mathbb{E}[UOI(x, \xi)]$ as the weighted sum:

$$\mathbb{E}[UOI(x, \xi)] = \sum_{k=1}^{K} p_k UOI(x, \xi_k) \tag{4.24}$$

Our simheuristic approach is aimed to solve the problem (Equation 4.21) using the general solving scheme that has been proposed by [86]. We use the following simheuristic algorithm (Algorithm 6 and Algorithm 7). For the metaheuristic part of the simheuristic, we picked the Reduced Variable Neighborhood Search (RVNS), a VNS variant (see also Section 2.7 and Section 2.7.4). It could quickly reach reasonable quality solutions for large instances, without applying an iterative improvement procedure, as the Basic Variable Neighborhood Search (BVNS) (see also Section 2.7.1), but it only explores randomly different neighborhoods. The most computationally intensive part of our algorithms is the fitness calculation, in which MCS is used. Therefore, we keep a tabu list to avoid calculating any candidate solution's fitness more than once. In the first step (Algorithm 6), our Sim-RVNS algorithm takes as input the initial state of the fuses when the potential overload has occurred. The number of neighborhoods, and the maximum number of iterations without change of the best solutions, and the maximum number of the elite solutions are also given, initially. After that, the

RVNS procedure continues until the stopping criterion, *i.e.*, the maximum number of iterations is reached. In the end, Algorithm 6 returns the list of the elite solutions found after fast simulations.

In the second step, Algorithm 7 takes as an input the elite solutions are found in the first step (Algorithm 6). For every elite solution, a long simulation is performed, and the tuple of the best solutions is updated. In the end, Algorithm 7 returns the tuple of the best solutions, containing the configuration with the corresponding, expected UOI.

As time is a determinant factor for our problem, we need to reduce the number of simulations needed. Therefore, we use a variance reduction technique, the control variate [120]. The variance reduction techniques minimize the MCS estimator's standard error by making the estimator more deterministic. A raw yet insightful approximation achieves this by the control variate method. As the covariance between the UOI^a (Equation 4.22) and UOI^g (Equation 4.23) is high, we pick the geometric mean as our control variate (Algorithm 8). Additionally, we use a modified adaptive sampling method, described in [104], by introducing a geometrically increasing sample size schedule having a dynamic rate to reduce the number of samples needed for our experiments (Algorithm 8). The fitness function is described in Algorithm 8.

4.3 Experimental evaluation

We evaluate the capability of the two methods we propose to calculate the short-term expected reliability of a smart grid after an overloading event. We consider a real-world topology from a neighborhood in Luxembourg city and real prosumption data, which are used to estimate future energy demands and supply. Both of the approaches deploy the above dataset to simulate the electrical values for the day-ahead from a given date and time. Our research question concerns the comparison of the expected reliability, in quantitative measures, of each approach, for the next day. The proposed methodology, as well as the details on the dataset and its features, are provided below.

Algorithm 6 Sim-RVNS for the reliability optimization problem
Step 1: RVNS with fast MCS

```
Require: initial state
         k_max : number of neighborhoods
         maxIter : maximum number of iterations without change of the best solution
         maxEliteSolSize : number of elite solutions
bestSol ← initial state
TabuList ← {bestSol}
EliteSol ← ∅
iter ← 0
Fast ← True
repeat
    k ← 1
    while k ≤ k_max do
        newSol ← SHAKE(bestSol, k)
        if newSol ∉ TabuList then                      ▷ if newSol has been already checked
            TabuList ← TabuList ∪ newSol
            if SIZE_OF(EliteSol) < maxEliteSolSize then
                if newSol ∉ EliteSol then
                    EliteSolutions ← EliteSol ∪ newSol
            else
                WorstSol ← WORSTSOL(EliteSol)
                if FITNESS(newSol, Fast) < FITNESS(WorstSol, Fast) then
                    if newSol ∉ EliteSol then
                        EliteSol ← EliteSol \ WorstSol
                        EliteSol ← EliteSol ∪ newSol
            if FITNESS(newSol, Fast) < FITNESS(bestSol, Fast) then
                bestSol ← newSol
                iter ← 0
                break
            else
                k ← k + 1
        else
            k ← k + 1
    iter ← iter + 1
until iter > maxIter
return EliteSol
```

Algorithm 7 Sim-RVNS for the reliability optimization problem
Step 2 : Long MCS for the elite solutions

Require: *EliteSol* from the Algorithm 6
$BestResults \leftarrow \emptyset$
$Fast \leftarrow$ `False`
for all $sol \in EliteSol$ **do**
  $BestResults \leftarrow BestResults \cup (sol, \text{FITNESS}(sol, Fast))$
return *BestResults*

4.3.1 Prosumption data and topology

The dataset covers 711 different customers' profiles, average and standard deviation of active and reactive consumption demand, as well as active and reactive production supply data, is provided by Creos Luxembourg S.A. The data consisted of 12 months of values, with 96 measurements per day. The topology dataset consisted of 23 cabinets, 3 of which are substations, 31 cables, and 219 smart meters, extracted from a real neighborhood in Luxembourg, which is also provided by Creos Luxembourg SA.

As it is unlikely we have more than 25% of overloaded prosumers on a grid and, if a curtailment policy can be applied, it is also unlikely we could curtail home residents, we created four different case realistic scenarios as a combination of different percentage of overloaded and curtailed prosumers:

Table 4.1: Case scenarios.

Scenario	Percentage of overloaded prosumers	Percentage of curtailed prosumers
1	10%	0%
2	10%	25%
3	25%	0%
4	25%	25%

For each one of the 219 smart meters, five random profiles are picked from the historical dataset. Depending on each scenario's percentage of overloaded and curtailed customers, the equivalent number of smart meters is picked uniformly

Algorithm 8 Function FITNESS($sol, Fast$)

Fitness function with adaptive simulations and control variates

Require: sol : solution to check, $Fast$: flag for fast simulations
N_{quick} : number of fast simulations, , c_1 : default multiplier (1.1)
c_0 : minimum multiplier (1.05), c_h : maximum multiplier (3)
UOI_s^a : Arithmetic mean of UOI (Equation 4.22) for experiment s, based in sol
UOI_s^g : Geometric mean of UOI (Equation 4.23) for experiment s, based in sol
$maxiter$: Maximum iterations for the same number of simulations, *e.g.*, 5

if $Fast$ **then**
    $innerError \leftarrow relaxedFastStoppingCriterion$
    $outerError \leftarrow fastStoppingCriterion$
else
    $innerError \leftarrow relaxedLongStoppingCriterion$
    $outerError \leftarrow longStoppingCriterion$
Calculate initial load (from current electrical values)
$amean \leftarrow \frac{1}{N_{quick}} \sum_{s=1}^{N_{quick}} UOI_s^a$
$gmean \leftarrow \frac{1}{N_{quick}} \sum_{s=1}^{N_{quick}} UOI_s^g$
$cov \leftarrow \frac{1}{N_{quick}-1} \sum_{s=1}^{N_{quick}} \left[(UOI_s^a - amean) \cdot \left(UOI_s^g - gmean\right)\right]$
$var \leftarrow \frac{1}{N_{quick}-1} \sum_{s=1}^{N_{quick}} \left(UOI_s^g - gmean\right)^2$
$c^* \leftarrow -\frac{cov}{var}$
$\tau \leftarrow 0; N \leftarrow N_{quick}$
$\mathbb{E}(interval) \leftarrow \infty$
while $\mathbb{E}(interval) < outerError$ **do**
    if $\tau = 1$ **then**
        $c_1 \leftarrow \min(2c_1 - 1, c_h)$
    else
        if $\tau = maxiter$ **then**
            $c_1 \leftarrow \max(c_0, \frac{c_1+1}{2})$
    $N \leftarrow \lceil c_1 N \rceil$
    $\tau \leftarrow 0$
    while $(\tau < maxiter) \wedge (\mathbb{E}(interval) < innerError)$ **do**
        $gmean \leftarrow \frac{1}{N} \sum_{s=1}^{N} UOI_s^g$
        $\mathbb{E}(UOI) \leftarrow \frac{1}{N} \sum_{s=1}^{N} \left[UOI_s^a + c^* \left(UOI_s^g - gmean\right)\right]$
        $\mathbb{E}(interval) \leftarrow z_a \cdot \frac{\sqrt{\frac{1}{N-1} \sum_{s=1}^{N} \left[UOI_s^a + c^* (UOI_s^g - gmean) - \mathbb{E}(UOI)\right]^2}}{\sqrt{N}}$
        $\tau \leftarrow \tau + 1$
return $\mathbb{E}(UOI)$

randomly, and initial consumption and production energy data are created from the corresponding Gaussian distribution. Thus, our experimental dataset consists of 20 different energy data files. Five different initial consumption and production energy data are created for every smart meter. Curtailment policy to the customers is applied if a prosumer overpasses the threshold of 60 A, *i.e.*, 80% of 75 A, the typical roof-top solar panel installation amperage, or if a consumer overpasses the threshold of 32 A, *i.e.*, 80% of 40 A, the typical amperage supplied by residential meters. If a prosumer or a consumer is picked for curtailment, its active energy is limited to 20 A; a value picked together with Creos Luxembourg SA. The "good" UOI level is set to 0.006, meaning less than 0.25 overloads per day, 5 mins of interruptions per day, and 15 mins of restoration per day, while the "poor" UOI level is set to 0.013, meaning over than 0.5 overloads per day, 15 mins of overloads per day, and 30 mins of restoration per day, as suggested by Creos Luxembourg SA.

The experiments[1] were conducted on a standard MacBook Pro with a 2.6 GHz Intel Core i7 processor, macOS Catalina 10.15.3 operating system, 16 GB 2133 MHz LPDDR3 memory using Java JDK 1.8.0-162 and Gurobi Optimizer 9.0.1 – Academic Version [60].

4.3.2 Experimental setup

To estimate the reliability indices, for our first approach, we first apply, for each one of the 20 instances, the method described in [6]. For each one of the optimal and sub-optimal results and the next 96 quarters of an hour, *i.e.*, 24h, we run adaptive simulations as described in Algorithm 8 to estimate the future electrical values of the customers. We then solve the corresponding linear systems to get the current load of each cable and calculate the indices and the UOI. The average time and the 95% confidence interval for the experiments were found to be equal to 444 sec +/- 448 sec. The minimum and the maximum time was found to be 0.5 sec and 14,930 sec, respectively.

[1] Interested readers may find all the presented results along with the input data files from `http://tiny.cc/SimheuristicRVNS`

In our second approach, we investigate a single-stage stochastic program that gives the impact of the solutions to the grid's reliability level. We apply the simheuristic algorithm, described previously in Algorithms 6 and 7. Neighborhood search or local search is considered a highly efficient metaheuristic mechanism for solving many problems of satisfaction with constraints and optimization. In defining a neighborhood and starting with an initial solution, local search is gradually exploring the current solution's neighborhood for improvement. In this way, one of its neighbors (often improving) replaces the current solution iteratively until a specific stop criterion has been met [94]. We define five neighborhoods, based on the one-flip heuristic, where a flip means assigning the opposite state to a variable, *i.e.*, negation, of the given solution in the algorithm's metaheuristic part as follows:

- N_1 *One-flip (substation)*. In this method, we flip each fuse that changes the substation, which powers every cable, if possible, in our current solution and appends that to the neighborhood list.

- N_2 *One-flip (parallel)*. In this method, we flip each fuse in parallel cables, in such a way that the cables become parallel or not, in our current solution and append that to the neighborhood list.

- N_3 *One-flip (end fuses)*. In this method, we flip each end fuse in every cable in our current solution and append that to the neighborhood list.

- N_4 *One-flip (start fuses)*. In this method, we flip each start fuse in every cable in our current solution and append that to the neighbourhood list.

- N_5 *One-flip (all fuses)*. In this method, we flip each fuse in every cable in our current solution and append that to the neighbourhood list.

Table 4.2 shows the possible neighborhoods for the presented example in Figure 3.1.

Table 4.2: Neighborhood example.

Neighborhood	**Current solution**	**Possible Neighborhood list** *applying one-flip*
N_1	[1 1 1 1 1 0 1 0 1 0 1 0 1 0]	[1 1 1 ? 1 ? 1 ? 1 0 1 0 1 0]
N_2	[1 1 1 1 1 0 1 0 1 0 1 0 1 0]	[1 1 1 1 ? ? ? ? ? ? ? ? 1 0]
N_3	[1 1 1 1 1 0 1 0 1 0 1 0 1 0]	[1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ?]
N_4	[1 1 1 1 1 0 1 0 1 0 1 0 1 0]	[? 1 ? 1 ? 0 ? 0 ? 0 ? 0 ? 0]
N_5	[1 1 1 1 1 0 1 0 1 0 1 0 1 0]	[? ? ? ? ? ? ? ? ? ? ? ? ? ?]

The number of elite solutions is set to five, while the maximum number of iterations without change of the best solution, so far, is set to 80. We use a tabu list to avoid the possibility of recalculating the same solution, as it is very costly. For the fitness function, described in Algorithm 8, the number of fast simulations is set to 30, while the indices c_0, c_h, and c_1 are set to 1.05, 3 and 1.1, respectively. The tail area α for the standard normal distribution is set to 0.025, i.e., $z_{.025} = 1.96$.

We run the experiments to calculate the UOI with three and four digits precision. The combination of precision and the stopping criteria we use is presented in Table 4.3.

Table 4.3: Stopping criteria.

Fast/Long	**Significant digits**	**Relaxed**	**Normal**
Fast	3	0.005	0.003
Long	3	0.002	0.001
Fast	4	0.001	0.0005
Long	4	0.0002	0.0001

In all of the following experiments, ten independent runs with different random seeds were conducted for each scenario and instance, to acquire statistically significant results. For each scenario and instance we pick the best solution. For the experiments with three Decimal Places (DP) precision, the average time and the 95% confidence interval for the experiments was found to be equal to 348 sec

+/- 26 sec. The minimum and maximum time were found to be 185 sec and 694 sec, respectively.

For the experiments with four DP precision, the average time and the 95% confidence interval for the experiments was found to be equal to 1,350 sec +/- 372 sec. The minimum and maximum time were found to be 211 sec and 7,557 sec, respectively.

4.4 Results

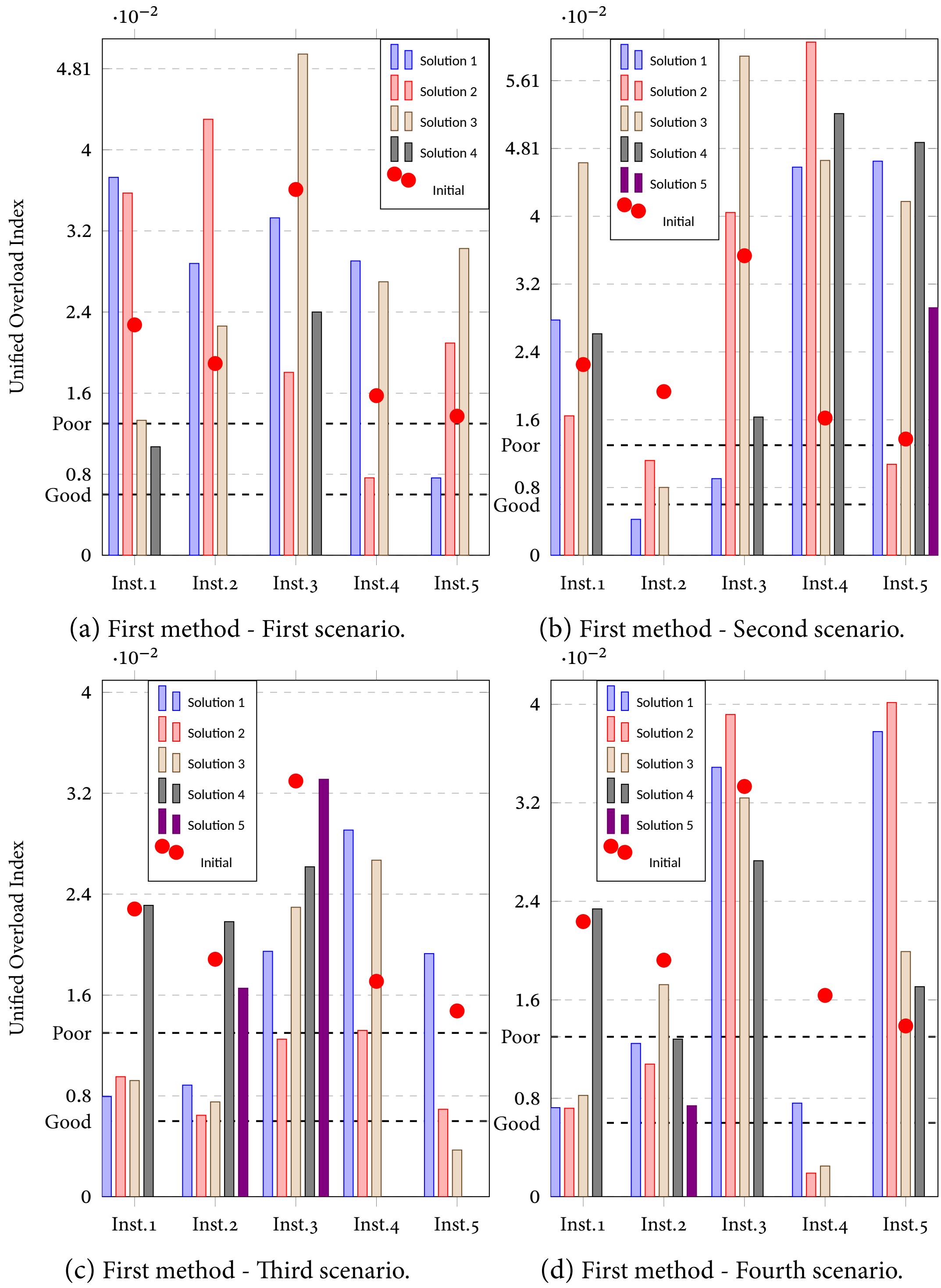
We can see from the results that the first method, as expected, gives poor results.

In Figures 4.3 we observe that about 44% of the cases, the calculated results are worse than the initial, *i.e.*, when an potential overload incident is occurred, ones. In only three instances, the UOI is under the "good" UOI level. The time to conclude the experiments also has a huge variance and, in the worst-case needed over four hours to finish. As our first method tries to find a solution without concerning the future electrical values, the quality of the solutions is, understandably, poor.

Our second method improves the results significantly. Even with 25% of overloaded customers, our method finds a proper plan to ensure a stable grid, with the minimum disturbances for the smart grid users and minimize the risk of overheating cables.

For the experiments that calculate the UOI with three digits precision, in the worst case, about 11.5 min are enough to find a solution, 3.5 min under the threshold of 15 min, which is the time interval between smart meter measurements, for the Creos Luxembourg SA case. In all scenarios and instances, the UOI is under the "good" UOI level, meaning that our method can be used as a robust reliability optimization tool for the smart grid companies, as can be seen in Figure 4.4, and in Figure 4.5.

We also conducted experiments with four digits precision to examine the risk's practical significance in the results. The difference between the results are given with the three decimal places precision, Figure 4.4 and the four decimal places precision, as it can be seen in Figure 4.5, is negligible. Moreover, the four decimal



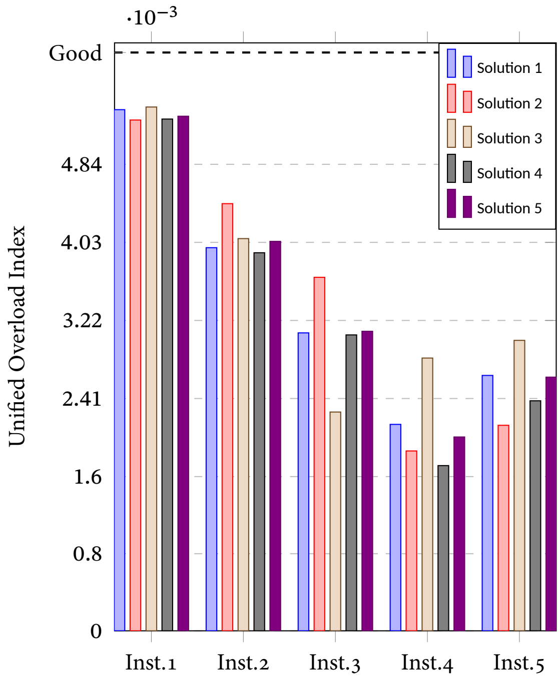
(a) First method - First scenario.

(b) First method - Second scenario.

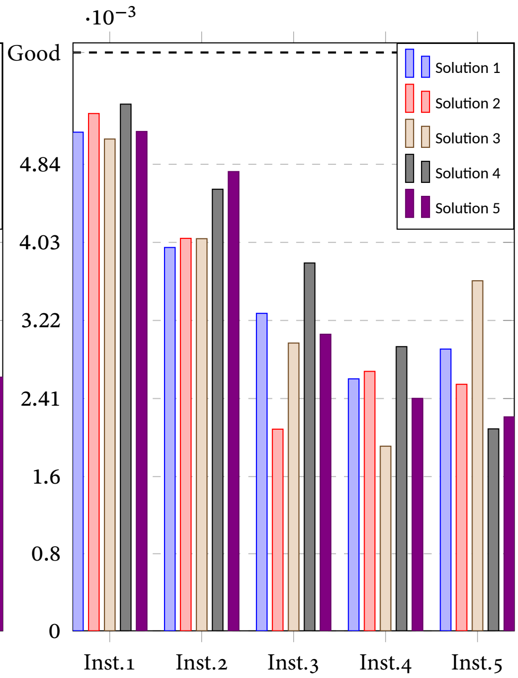
(c) First method - Third scenario.

(d) First method - Fourth scenario.

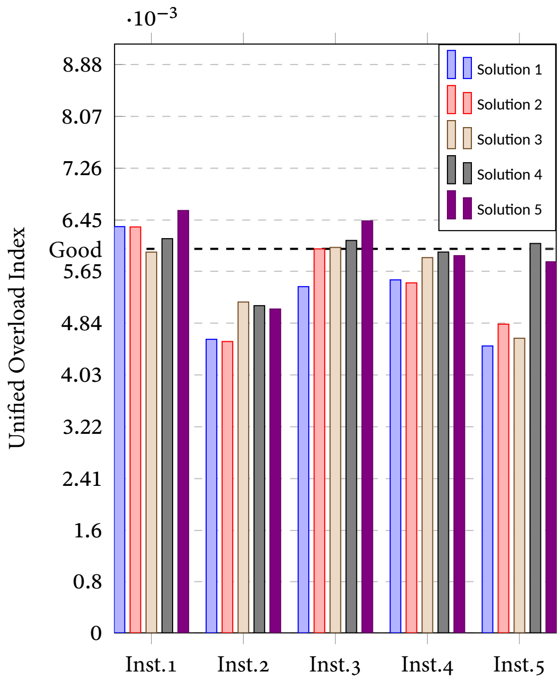
Figure 4.3: Simulation method experiments.



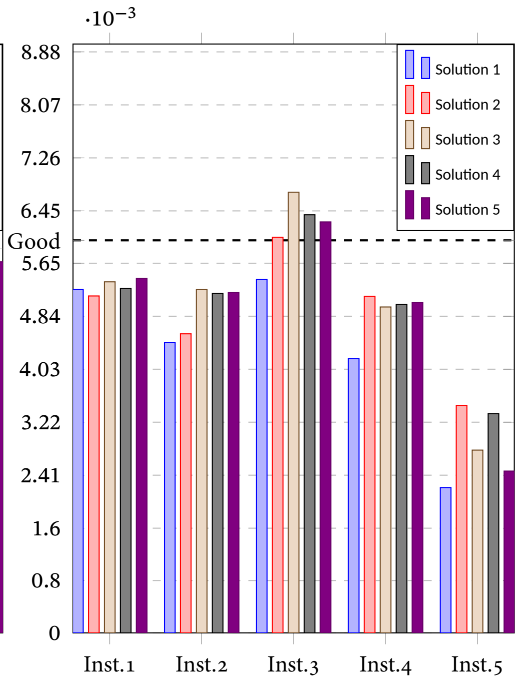
(a) 2$^{\text{nd}}$ method, 1$^{\text{st}}$ scenario, 3 DP.



(b) 2$^{\text{nd}}$ method, 2$^{\text{nd}}$ scenario, 3 DP.

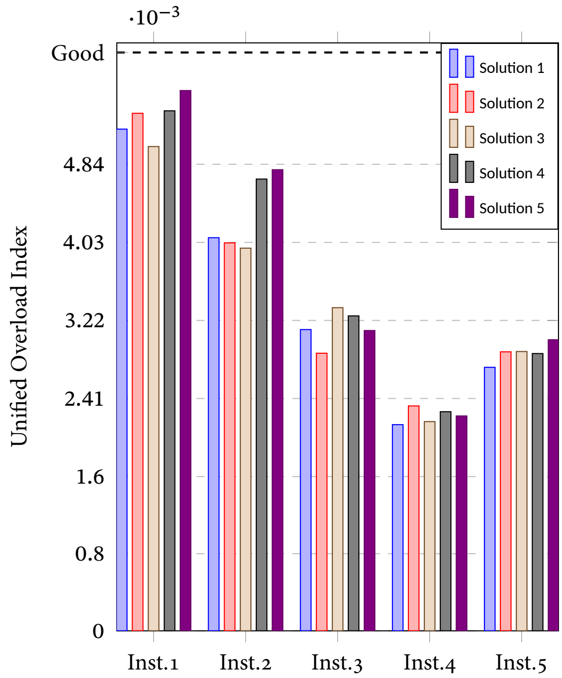


(c) 2$^{\text{nd}}$ method, 3$^{\text{rd}}$ scenario, 3 DP.

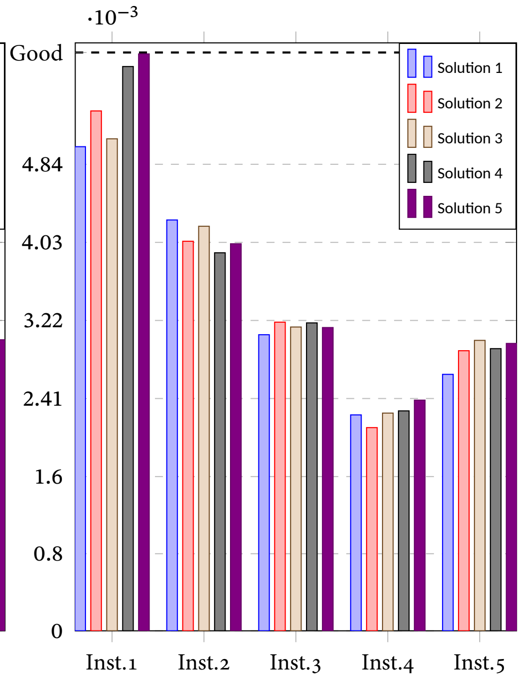


(d) 2$^{\text{nd}}$ method, 4$^{\text{th}}$ scenario, 3 DP.

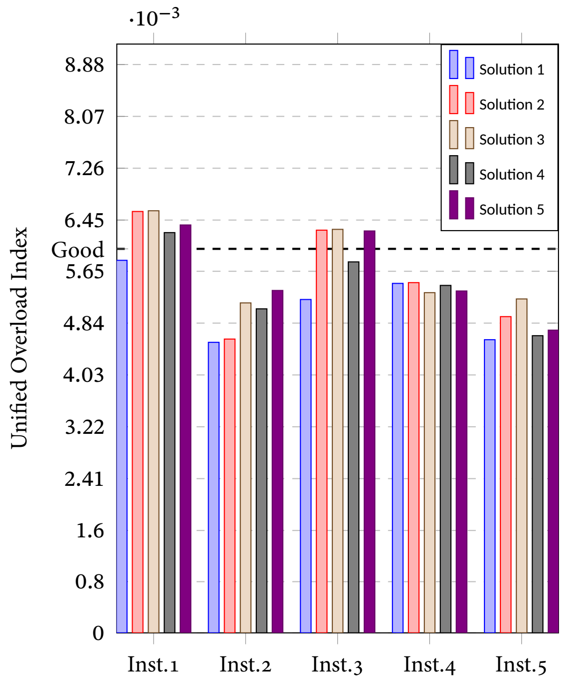
Figure 4.4: Simheuristic method experiments, 3 DP.



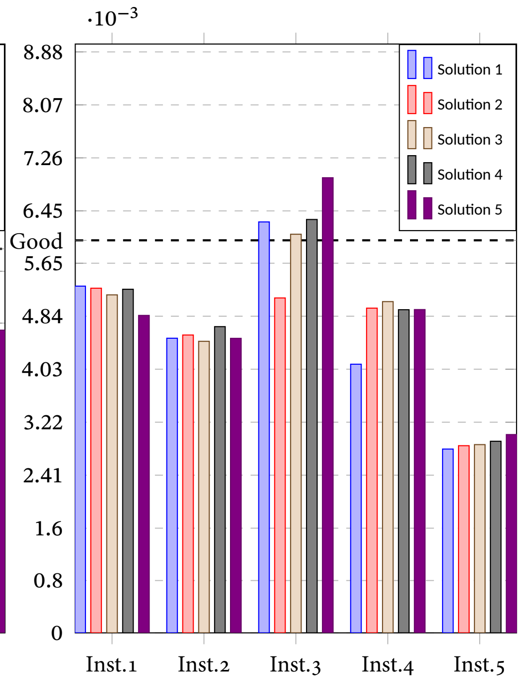
(a) 2nd method, 1st scenario, 4 DP.



(b) 2nd method, 2nd scenario, 4 DP.



(c) 2nd method, 3rd scenario, 4 DP.



(d) 2nd method, 4th scenario, 4 DP.

Figure 4.5: Simheuristic method experiments, 4 DP.

places precision experiments last, on average, about 22.5 min, over the smart meter 15 min measurement threshold. As our method needs to find fast such healing actions to ensure a smart grid's stability after a potential overload, the three digits precision for the UOI is enough for the smart grid operators, according to Creos Luxembourg SA.

4.5 Conclusions

In this chapter we defined and formulated the single-stage stochastic reliability optimization problem in smart grids and suggested two solution methods using MCS and simheuristics. It is shown that this approach can be included in the grid operator's decision-making process as it can successfully and rapidly help to ensure the stability of a smart grid after a potential overloading incident of about the size of a neighborhood in Luxembourg.

5 Smart Meter Communication Monitoring

> Dans les champs de l'observation le hasard ne favorise que les esprits préparés. [In the field of observation, chance favours only the prepared mind.]
>
> *(Louis Pasteur)*

Into this chapter, we suggest using ND-trees to learn for each smart meter a failure pattern over time, which then acts as a profile for the respective smart meter. This profile is used to decide, in real-time, if an alarm should be raised or if the reading error can be considered as "normal". This derivative contribution can help grid operators decide when reading failures can be considered uncritical and ensure that the electrical data are clear from any false reading failures.
This chapter is based on work that has been prepared in the following paper:

- *"Intelligent Smart Meter Communication Monitoring by Learning Failure Patterns using ND-Trees." Manuscript in preparation. Antoniadis N., Cordy M., Le Traon Y.*

Contents

5.1 Introduction

The so-called smart grid has begun with the need to manage the increasing complexity of today's electricity grid. It aims to follow the rising demand for energy, for example by integrating renewable energies and by providing new and innovative services, mainly driven by sensors and two-way communication between smart meters and electricity providers. The vast amount of data, *e.g.*, customers' consumption, which can be automatically measured in real-time, has the potential to significantly increase the reliability and efficiency of today's electricity grid. Besides enabling new functions like automated meter reading [44] and on-demand pricing [124], the regular measurements from smart meters are also used beyond these uses cases to monitor essential characteristics of the grid, as the quality of power supply [72]. While certain conditions, like heavy rain, wind, electrical interference, or worn communication lines, can naturally disturb the remote reading of smart meters, meter reading failures can indicate severe disturbances and potential problems in the grid [146], [78], [71], [102].

The challenge for grid operators is to differentiate between the two. The question is when it becomes necessary for a grid operator to send a technician to check if there is an actual problem or if a reading error can be considered as "normal", given a specific meter. For example, in Power-Line Communication (PLC) [49] and Global Positioning System (GPS) networks, meter reading failures can naturally occur due to weather conditions, high cable load [125], electrical appliances, medium wave broadcast stations [23], and nearby photovoltaic panels [112]. Therefore, some smart meters might be regularly not reachable for a significant amount of time, *e.g.*,up to several hours. Nonetheless, depending on the meter and external parameters (location of the meter, weather conditions, nearby photovoltaic panels etc.) [3] [92] this could be still considered as "normal". On the other hand, for other meters, only a few reading failures could indicate an actual problem in the grid and its communication infrastructure.

Significant effort has been made to minimise the noise effects in the communication topology. For example, in the context of PLC, protocols like PLC PRIME [8], PLC G3 [115], and PLC G1 [114] include techniques for handling noise. Although they reduce the number of communication errors, the general problem remains and hence the necessity for detecting the normal communication behaviour.

Therefore, an intelligent monitoring and alerting system has to only raise an alarm if the reading failures for certain smart meters can be considered as unusual. We are working together with our industrial partner, Creos Luxembourg[1], on a machine learning-based monitoring system that is able to learn the failure patterns for smart meters and that raises only alarms if the reading errors are outside of their expected failure patterns. More specifically, in this chapter, we present a novel approach using ND-trees [109] to encode the failure pattern of a smart meter over time. In this way, every smart meter is associated with its own ND-tree, reflecting the learned failure pattern of the smart meter. In other words, the ND-tree encodes the profile of meter failure readings of a smart meter. Whenever a new reading failure is detected, the ND-tree of the smart meter can be used to verify if the reading failure is expected or not. This is fast enough to be done in near real-time. Most importantly, this approach is able to significantly lower false positive and negative alarms.

To evaluate our method, we first define a dataset including results of reading attempts on smart meters. This dataset contains five years of data, including measurements of weather conditions and sun altitude, together with meter readings. Then, we use one ND-tree for each smart meter. We use two-thirds of the dataset to "train", *i.e.*,feed, the ND-trees and the remaining third of the dataset for testing.

The remainder of this chapter is as follows. In Section 5.2, we describe the necessary background for this work. First, we describe the most relevant elements of a typical smart grid communication infrastructure, based on the example of the one in Luxembourg. Secondly, we describe ND-trees, which we use in our proposed monitoring system to learn the failure patterns of smart meters. Then, in Section 5.3 we detail the implementation of our proposed monitoring system, which we evaluate in Section 5.4, before we conclude in Section 5.5.

[1] Creos is the main electricity grid operator in Luxembourg

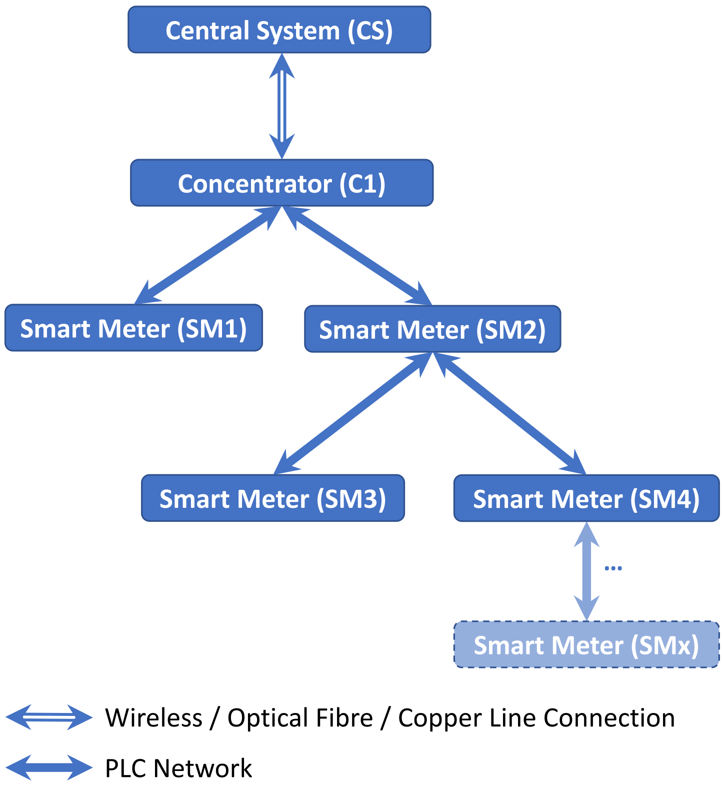


Figure 5.1: Representation of the smart grid communication topology.

5.2 Background

In this section, we first explain the main communication components of a typical smart grid. In the second part of this section, we introduce ND-trees.

5.2.1 A smart grid's communication topology

Like it is the case in many countries [72], the smart grid communication topology in Luxembourg is based on a PLC [49] network. The main reason for the widespread use of PLC is that the communication and information exchange between the smart grid elements can be handled using the same network that is used for transmitting the energy. Probably one of its most significant disadvantages is that data exchange is comparatively slow. The main elements that constitute a smart grid communication topology are shown in Figure 5.1.

Smart meters are responsible for measuring the electric consumption and the quality of the power supply at customers' houses. They are responsible for transmitting the collected data to a so-called *data concentrator*, or to another smart meter in case the data concentrator could not be reached. In such situation, this smart meter acts as a so-called *repeater* for another meter (*e.g.*, smart meter 2 and smart meter 4 in Figure 5.1).

Data concentrators are controlling all smart meters, one at a given time, that are directly or indirectly connected to it. Data concentrators are in charge of collecting and storing the consumption data from the smart meters. Data concentrators send this data periodically (several times per day or immediately) to a central system.

The ***central system*** stores and analyses all data received from the concentrators, communicating with them through an optical fibre, a copper line, or a wireless connection.

Due to space limitations, this description is limited to the necessary parts of a smart grids communication infrastructure, which directly concern the context of this chapter. More detailed information about the typical elements of a smart grid communication topology can be found in [72], [75], and [73].

5.2.2 ND-tree

To identify a real reading error—*i.e.*, outside of the expected behaviour—for a smart meter, information about its regular behaviour statistics, weather conditions, time and reading results, are needed. Some of this information, like the temperature during measurement, is ordered and continuous, while other information, like the reading error, is non-ordered and discrete (true, false). A widely used model [11] to represent these elements is a multi-dimensional vector. In this model, each element is then a point in a multi-dimensional data space. Such space that contains both continuous and discrete non-ordered dimensions is often called a Hybrid Data Space (HDS) [29]. A HDS can be considered as a union between the Continuous Data Space (CDS), where data values in each dimension are continuous, and the Non-ordered Discrete Data Space (NDDS), in which all elements along each dimension are discrete and have no natural ordering. An efficient method to search such data space is to search for similarities; in other words, to find the most similar vector to the one we search for.

An ND-tree is such a technique to search for similarities, initially designed for multi-dimensional NDDS. Qian *et al.* [109] have proposed it to discuss exploring non-ordered data spaces where the index comprises multi-dimensional information, *e.g.*, weather conditions, time and reading results. Even so it has been proposed for NDDS, it performs equally well for other methods, *e.g.*, for HDS, when the number of discrete dimensions is the same or greater than continuous dimensions [29].

Its tree structure, assuming that our dataset is organized as vectors in a four-dimensional HDS, is organized as follows: Every **non-leaf node** in ND-tree stores, as an array of entries, a pointer to its child node as well as the nearby values, Discrete Minimum Bounding Rectangle (DMBR) [109], of its child nodes. Each **leaf node** also stores an array of entries: a four-dimensional, or multi-dimensional in general, vector, as *key* containing both the discrete and continuous components, and a pointer to the actual data of this specific vector. Figure 5.2 shows an example of an ND-tree where its keys are vectors in a four-dimensional HDS with discrete domains {0,1}, {0,1,...,23}[2], and continuous domains [-12,36] and [0,100].[3] Each leaf node contains an array of the vector's coordinates in the HDS space. Non-leaf nodes, on each level, group the more "similar" vectors, based on their DMBR [109]; *e.g.*, the second, non-leaf, non-root, level signifies this grouping as the $[16, 17] \times [3, 7] \times [65, 70] \times \{0, 1\}$ entry is based on its child (*leaf*) nodes.

To sum up, an ND-tree can provide, efficiently, similarity searches in NDDS. It performs very well [108] when searching similarity ranges; in fact, the larger the dataset, the more the improvement in performance. Moreover, it scales well with the domain size as well as the data space's dimensions.

[2] for discrete domains the numbers in brackets denote the discrete values this dimension is allowed to take

[3] for continuous domains the numbers in brackets denote the lower and higher values this dimension is allowed to take

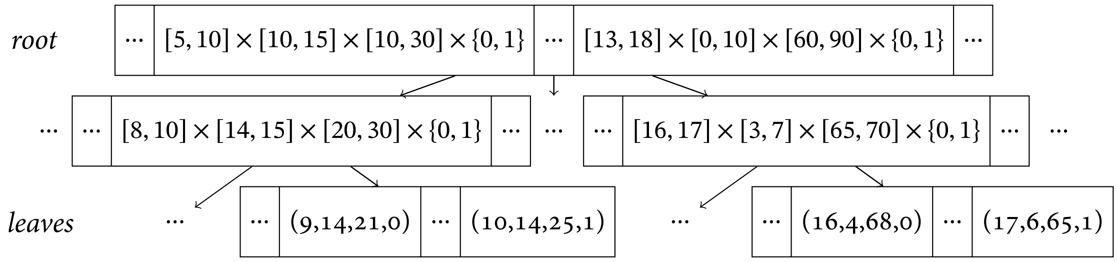


Figure 5.2: An example of the ND-tree

5.3 Contribution

In this section, we discuss our machine learning based monitoring and alerting system. This system continuously learns the "normal" pattern of the smart meter communication in order to alert the smart grid operator only in case a smart meter behaves differently from this pattern. We present how we create a multi-dimensional HDS consisting of smart meters' profiles and then make similarity searches in this space for unexpected behaviours. In other words, our purpose is to develop a technique that learns the "normal" communication profile of each smart meter and alerts the smart grid operator when a different behaviour is detected. More specifically, we use ND-trees to store and compute the individual profiles for each smart meter, as we believe that this technique is very suitable for efficient similarity searches in HDSs.

Each smart meter profile consists of the outcome of the communication attempts to it, success or failure. Moreover, it stores additional information that influences the transmission success, which are:

- hour of the day
- temperature
- cloud coverage

The similarity queries are performed in near real-time using the previous communication attempts stored in the ND-tree. In this way, it is possible to get an estimation at a particular time of the day, temperature and cloud coverage percentage, about the communication status of a given smart meter.

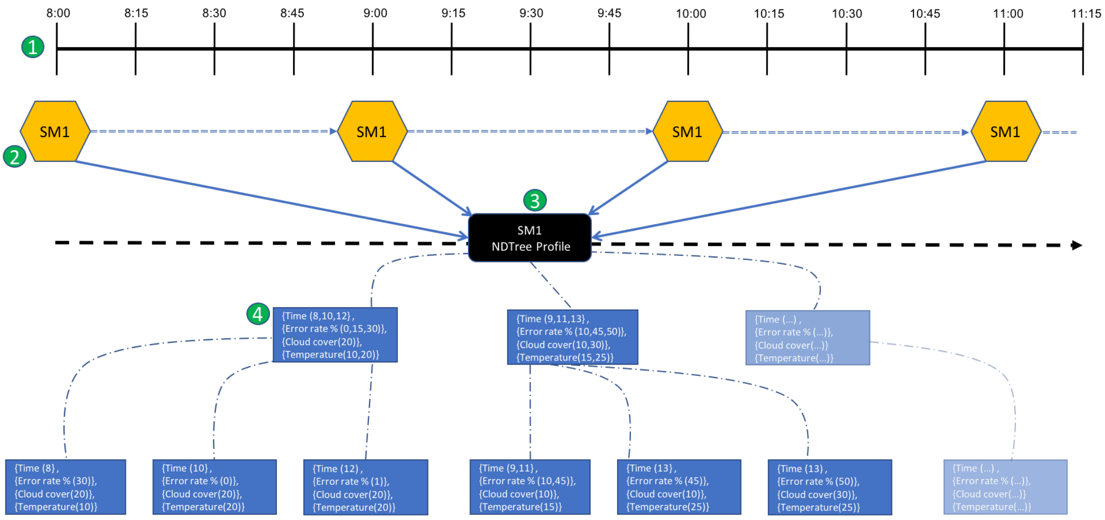


Figure 5.3: Working Structure

In the event that we have similar data, we would construct the following ND-tree for this smart meter. Figure 5.3 exemplifies the process of our approach. The time scale (1) represents the hour of the day when the reading process takes place. In this scenario, each smart meter is being read every 15 minutes, and the hourly success rate with the environmental values are stored as a vector in the smart meter's ND-tree (3). This new vector, based on the insertion method of the ND-tree, will be held to the most relevant leaf node. This operation starts from the root node and follows a path to the spotted leaf node. At each non-leaf node, it has to select which child node to follow. If the appointed leaf node overflows after this reconciliation, a splitting procedure is applied from leaf nodes to the root (*bottom-up*).

Each hexagon, marked as (2), represents the *SM1* smart meter, and each arrow represents the hourly aggregation of its profile values. Furthermore, each rectangle marked as (4) represents a non-leaf node of the ND-tree structure. The non-leaf nodes aggregate the multi-dimensional index of its subnodes (*leaf nodes*), while the leaf nodes contain the specific vector in our four-dimensional HDS. For example, in Figure 5.3, based on the leftmost leaf node, the vector contains the information that at 8 AM, 20% of cloud coverage, 10°C of temperature, and we had 30% of errors in reading attempts.

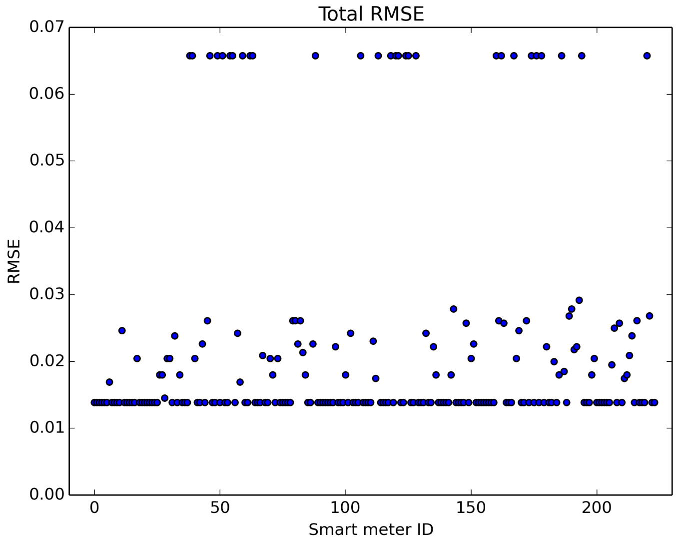


Figure 5.4: Total RMSE by smart meter

Moreover, a search on this ND-tree could verify if the reading failure is expected or not. A query, when the tree is constructed, acts as a thorough search into the ND-tree. In fact, the query algorithm starts from the root node and eliminates dissimilar, based on their DMBR, nodes until it finds the leaf nodes with the desired vectors. Given that, the retrieved information—a four-dimensional vector that represents each smart meter's profile—is used to inform us, if in the given conditions the smart meter is expected to be reachable or not. As an illustration, if we search for the expected error rate in case that time is 1 PM, there is 25% of cloud coverage and 20 °C of temperature, the result will be 50% due to the most "similar" leaf node—the rightmost blue one.

5.4 Evaluation

In this section, we discuss the evaluation of our proposed monitoring method. We first describe the evaluation setup and used dataset before discussing the results of the experiments.

5.4.1 Setup

Before we detail the evaluation setup, we start with a description of the kind of meter reading errors that we consider in the following experiments. To transform solar energy, which comes from the sun in the form of solar irradiance, Photovoltaic (PV) technology[4] is applied. The output of a PV solar panel is Direct Current (DC); therefore a solar inverter is essential to convert the variable DC into a utility frequency Alternating Current (AC) that can be fed into the electrical grid. However, solar inverters generate not only AC but also radio interference, due to their high-frequency switching devices [112]. This interference can also travel through AC power lines. In our experiments, it was assumed that the reading failures depend *only* on noisy photovoltaic panels, except the other factors hour of the day, temperature, and cloud coverage. With this in mind, we evaluate our approach, based on real data from the smart grid testbed deployment in Luxembourg. Particularly, for the cabling and location of the smart meters, real smart grid data is used, whereas topology data, concerning solar panels and filters, are generated as follows. We suppose that solar panels are installed into approximately 15% of total consumers, while 30% of them have noise filters. Therefore, we pick, uniformly, the corresponding number of consumers from the testbed data, in which solar panel and noise filter is installed. To build the dataset for our experiments, for each day and hour of the weather data and each smart meter, the reading failures, which are caused by filterless nearby photovoltaic panels, are generated. To include the weather conditions into the dataset, temperature and cloud coverage, we use public data taken from *https://rp5.ru*[5] starting from 12 July 2013 12 AM until 11 April 2018 11 PM, thus about 40,000 hours of meteorological data. We also use the *commons-suncalc*[6] Java library in order to calculate the sun altitude angle for each timepoint we had weather data. The final dataset for our experiments consists of the time stamp from weather data, the id of each smart meter, the temperature and the cloud coverage percentage at the specific time, and the result of each communication

[4] PV technology uses solar cells made of semiconductors to absorb the irradiance from the sun and convert it to electrical energy
[5] Raspisaniye Pogodi Ltd., St. Petersburg, Russia
[6] https://github.com/shred/commons-suncalc

Table 5.1: Confusion matrix (Full dataset)

		Communication errors test set (smart meters not reachable)		
		Condition positive	**Condition negative**	
Outcome	**Test positive**	True positive 175984	False positive 3493	**Precision 98.05%**
	Test negative	False negative 3689	True negative 2794242	**Recall 99.87%**
		Sensitivity 97.95%	**Specificity 99.88%**	

attempt. After the dataset was built, it was split between a training and a test set; two-thirds, from 12 July 2013 12 AM until 25 September 2016 11 PM, was used for training the ND-tree for each smart meter, and the rest one third was used for testing. The minimum and maximum values *[min,max]* for the ND-tree, on each dimension, was: for the hours [0, 23], for the temperature [-12, 36], for the cloud coverage [0, 100] and for the read errors [0, 1], where *1 denotes error*.

5.4.2 Experiments

The experiments were conducted on a standard MacBook Pro with a 2.6 GHz Intel Core i7 processor, macOS High Sierra 10.13.4 operating system, 16 GB 2133 MHz LPDDR3 memory using Java JDK 1.8.0-162. The training procedure for our experiments lasted about 2.5 hours, while the testing period was about 3 minutes. Figure. 5.4 depicts the Root Mean Squared Error (RMSE) for the whole profile vector (hour, temperature, cloud coverage, reading error). About 60% of the smart meters scored RMSE 0.014, 13% had RMSE 0.066 and the rest scored between 0.017 and 0.029. A contingency table of predictions, Table 5.1, against actual classes (also known as confusion matrix) represents the results of our binary classifier. As our method's goal is to reduce false positive and negative alarms, in Figure 5.5 the distribution, by hour, of the false values between the real ones, from the test dataset, and the ones our profiler resulted, is shown. This distribution indicates that the only period of the day we have, few, false values is during daylight. This is in line with our expectations, as the solar inverters

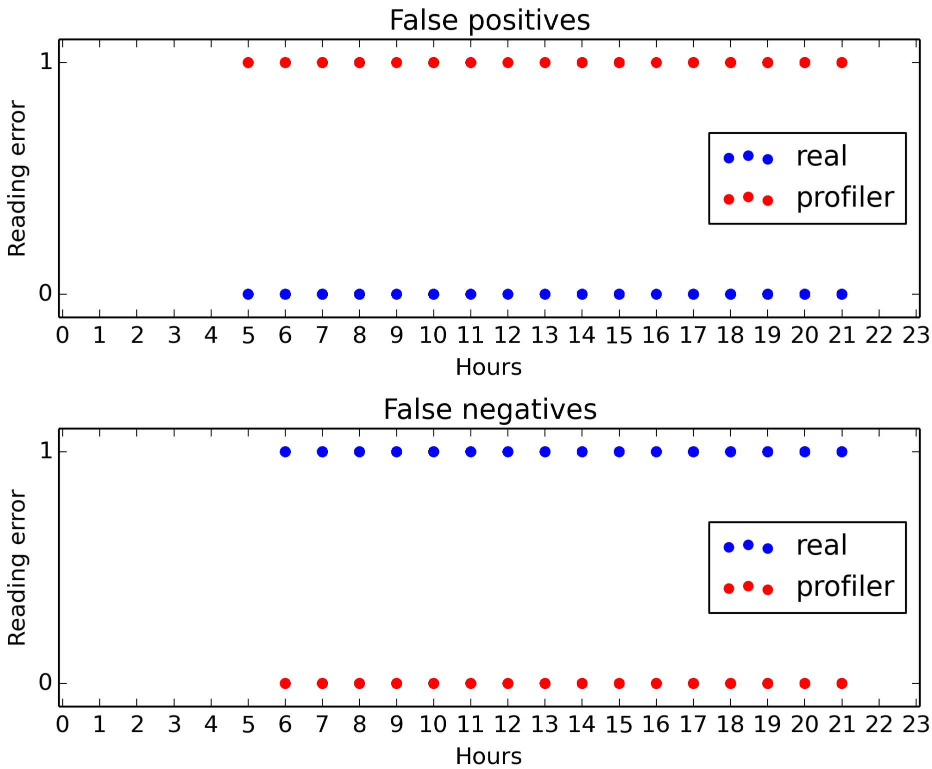


Figure 5.5: Hours of day false values occurred

Table 5.2: Confusion matrix (only daylight hours)

		Communication errors test set (smart meters not reachable)		
		Condition positive	**Condition negative**	
Outcome	**Test positive**	True positive 175984	False positive 3493	**Precision 98.05%**
	Test negative	False negative 3689	True negative 1925346	**Recall 99.81%**
		Sensitivity 97.95%	**Specificity 99.82%**	

generate radio interference only when electricity is produced, something impossible during nighttime. Therefore, in Table 5.2 the same test statistics, but only for daylight hours—between 5 and 21—are shown. To evaluate the robustness of our model, for the second table, we also calculated accuracy 99.66%, meaning the ratio of correctly predicted observations to the total observations.

5.4.3 Discussion

The above evaluation suggests that our proposed monitoring and alerting system can be used to identify unexpected reading failures in smart meters. The sensitivity reaches 97.95%, meaning that false negative alarms rarely occur, a crucial goal characteristic for the proposed system. Additionally, we evaluate the test performance through the Matthews Correlation Coefficient (MCC), as it is more informative than other confusion matrix measures [31], like accuracy, in binary classification problems. Our method scores $+0.9781$, where $+1$ represents a perfect classifier, 0 denotes random prediction, while -1 indicates total disagreement between prognosis and observation. Regarding the aforementioned confusion matrix, the proposed approach succeeds in detecting reading failures for certain smart meters, when the hour of the day and the weather conditions are given.

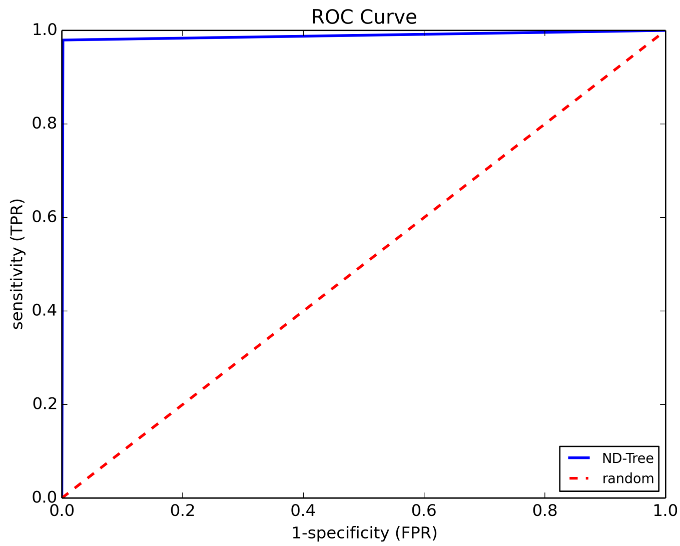


Figure 5.6: Receiver Operating Characteristic (ROC) curve of our classifier

5.5 Conclusion

The transition from the traditional power grid to an intelligent, self-healing and self-adaptive smart grid is currently in full swing, probably most visible through the large-scale deployment of smart meters presently undertaken in many countries. While certain conditions, like heavy rain, wind, electrical interference, or worn communication lines, can naturally disturb the remote reading of these smart meters, reading failures can indicate severe disturbances and potential problems in the grid. The challenge for grid operators is to decide when reading failures can be considered as uncritical and when it becomes necessary to send a technician to verify the problem. In this chapter, we presented a machine learning based approach—using ND-trees—which continuously learns the "normal" pattern of the smart meter communication. We presented how we create a multi-dimensional hybrid data space consisting of smart meters' profiles and then make similarity searches in this space to detect unexpected behaviour. We showed that this approach can significantly reduce false negatives, avoiding unobserved, that can be critical for the smart grid, communication errors. Equally important is the key reduction of false positives, limiting the number of sending technicians unnecessarily to investigate the reason why a smart meter is not reachable.

Part III

Epilogue

6 Conclusion and Future Work

> Everything gets better in the end. If it's not better, it's not quite the end.
>
> *(Paolo Coelho)*

This chapter summarizes the contributions of this dissertation and discusses potential directions for future work.

Contents

6.1 Conclusion

The demand for a "self-healing" grid requires novel tools to provide a resilient and reliable power grid to its users. As we have analyzed in Section 1.2, the overall aim of this thesis was to present a fully-edge study on reliability optimization for smart grids to provide, fast, the most appropriate countermeasures after a potential overloading disturbance. To achieve the above aim, we established a few objectives.

This study fulfilled the aim mentioned above by extending the applicability of simulation and mathematical optimization to the domain of power grids. In the following, a short summary is provided about each objective.

In Chapter 3 the overloading disturbances in a power grid were explored, and the counteractions to recover from an overloading incident were identified. The Overloading Prevention Problem (OPP) in smart grids as a Multi-objective Mixed Integer Quadratically Constraint Program (MO-MIQCP) was also described and formulated, and a solution method using a state-of-the-art exact solver was suggested. It is shown that this approach can be included in the grid operator's decision-making process as it can successfully and rapidly help to prevent challenging overloading incidents in a smart grid of about the size of a medium city in Luxembourg, minimizing the disconnections of the grid's users.

Then, in Chapter 4, the reliability assessment on power grids were examined and analyzed. To estimate the reliability for a planning horizon with unknown future electrical values, as a first method, Monte Carlo Simulation (MCS) was applied. The Stochastic OPP in smart grids was described and formulated as a Single-State Stochastic Program (SSSP), and a *simheuristic* approach was suggested to solve it. It is shown that this approach can be included in the grid operator's decision-making process as it can successfully and rapidly help to ensure the stability of a smart grid after a potential overloading incident of about the size of a neighborhood in Luxembourg, for the next 24 hours.

A software monitoring and alerting system to reduce false positive alarms for meter reading failures, based on live machine learning techniques, was proposed in Chapter 5. In this chapter, a machine learning based approach—using ND-trees—was presented, which continuously learns the "normal" pattern of the smart meter communication. We presented how we create a multi-dimensional hybrid data space consisting of smart meters' profiles and then make similarity searches in this space to detect unexpected behaviour and showed that this approach can significantly reduce false negatives, avoiding unobserved, that can be critical for the smart grid, communication errors. Equally important is the key reduction of false positives, limiting the number of sending technicians unnecessarily to investigate the reason why a smart meter is not reachable.

The above proposed solution methods were evaluated in a real-world smart grid topology, provided by our industrial partner, Creos Luxembourg SA.

6.2 Future Work

Initially, as examining new research directions, the approach presented on Chapter 3 can be parallelized to analyze every substation subgraph independently from other ones, as in [74]. As future work, it would be interesting to analyze the intermediate states to find the optimal order of fuses' change. During the analysis of these intermediate states, a "trade-off" metric should be calculated, as the difference between the maximum and the minimum load on the grid. This metric should offer an optimal trade-off between the number of actions to perform and the maximal overload that any cable or substation reaches during the execution of the actions. Furthermore, the application of a dynamic soft curtailment policy [20] to the grid's users would be a challenging idea. Another absorbing addition should be the appliance of a fairness policy to avoid curtailing the same users repetitively over time. Such considerations, raise the need for considering the future states of the grid and their inherent stochasticity, as the recovery response solution should guarantee stability over the next 24 hours. Inevitably, the aforementioned considerations complexify the problem, increasing the size of the problem and its solution space. As such, exact methods may not be suitable to address those new concerns. Thus, we also plan to exploit metaheuristic methods [130] to solve the overloading prevention problem.

As future work, we plan to extend our single-stage stochastic optimization problem into a multi-stage stochastic optimization problem, so that in every period, *i.e.*, 15 min, considering the expected electrical values, new actions are calculated and, in the end, an action plan for the next 24 hours is calculated with more precision. This multi-stage stochastic optimization problem inevitably complexify the presented problem, increasing the size and its solution space. Thus, other metaheuristics, as part of our simheuristic algorithm, should be investigated. Hybrid and parallel metaheuristics may be a suitable solution method.

For the monitoring contribution, we plan to implement various what-if scenarios to inspect under which conditions we could expect reachability errors. Another application would be to develop a simulator for a specific or a set of different scenarios to measure the smart grid robustness. Other techniques, like C-ND-tree [29], designed explicitly for HDS, can also be evaluated and compared with this proposed method.

Bibliography

1. S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury. "Simulation optimization: a review of algorithms and applications". *Annals of Operations Research* 240:1, 2016, pp. 351–380. ISSN: 1572-9338. DOI: 10.1007/s10479-015-2019-x.
2. M. Amin. "Modeling and Control of Electric Power Systems and Markets". *IEEE Control Systems* 20:4, 2000, pp. 20–24. ISSN: 1066-033X. DOI: 10.1109/MCS.2000.856176.
3. E. Ancillotti, R. Bruno, and M. Conti. "The role of the RPL routing protocol for smart grid communications". *IEEE Communications Magazine* 51:1, 2013, pp. 75–83.
4. G. J. Anders. *Probability concepts in electric power systems*. Wiley-Interscience, 1989. ISBN: 978-0471502296.
5. S. Andradóttir. "Chapter 20 An Overview of Simulation Optimization via Random Search". In: *Simulation*. Ed. by S. G. Henderson and B. L. Nelson. Vol. 13. Handbooks in Operations Research and Management Science. Elsevier, 2006, pp. 617–631. DOI: https://doi.org/10.1016/S0927-0507(06)13020-0.
6. N. Antoniadis, M. Cordy, A. Sifaleras, and Y. Le Traon. "Preventing Overloading Incidents on Smart Grids: A Multiobjective Combinatorial Optimization Approach". In: *Optimization and Learning*. Ed. by B. Dorronsoro, P. Ruiz, J. C. de la Torre, D. Urda, and E.-G. Talbi. Springer International Publishing, Cham, 2020, pp. 269–281. ISBN: 978-3-030-41913-4. DOI: 10.1007/978-3-030-41913-4_22.

7. J. April, F. Glover, J. Kelly, and M. Laguna. "Simulation/optimization using "real-world" applications". In: *Proceeding of the 2001 Winter Simulation Conference (Cat. No.01CH37304)*. Vol. 1. 2001, 134–138 vol.1. DOI: 10.1109/WSC.2001.977254.

8. A. Aruzuaga, I. Berganza, A. Sendin, M. Sharma, and B. Varadarajan. "PRIME interoperability tests and results from field". In: *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE. 2010, pp. 126–130.

9. S. Asmussen and P. W. Glynn. *Stochastic simulation: algorithms and analysis*. 1st ed. Stochastic Modelling and Applied Probability. Springer, 2007. ISBN: 9780387306797.

10. A. A. Babalola, R. Belkacemi, and S. Zarrabian. "Real-Time Cascading Failures Prevention for Multiple Contingencies in Smart Grids Through a Multi-Agent System". *IEEE Transactions on Smart Grid* 9:1, 2018, pp. 373–385.

11. R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*. Vol. 463. 1999.

12. J. Banks, J. S. Carson II, B. L. Nelson, and D. M. Nicol. *Discrete-Event System Simulation*. 5th internat. Pearson, 2013. ISBN: 9781292024370.

13. R. R. Barton and M. Meckesheimer. "Chapter 18 Metamodel-Based Simulation Optimization". In: *Simulation*. Ed. by S. G. Henderson and B. L. Nelson. Vol. 13. Handbooks in Operations Research and Management Science. Elsevier, 2006, pp. 535–574. DOI: https://doi.org/10.1016/S0927-0507(06)13018-2.

14. R. Benmansour, A. Sifaleras, and N. Mladenović, eds. *Variable Neighborhood Search. 7th International Conference, ICVNS 2019, Rabat, Morocco, October 3-5, 2019, Revised Selected Papers*. Vol. 12010. LNCS. Springer, Cham, 2020, pp. 1–195.

15. R. Berthier, W. H. Sanders, and H. Khurana. "Intrusion Detection for Advanced Metering Infrastructures: Requirements and Architectural Directions". In: *2010 First IEEE International Conference on Smart Grid Communications*. 2010, pp. 350–355. DOI: 10.1109/SMARTGRID.2010.5622068.

16. R. Billinton and R. N. Allan. *Reliability assessment of large electric power systems*. Springer Science & Business Media, 2012.

17. R. Billinton and R. N. Allan. *Reliability evaluation of engineering systems*. Springer, 1992.

18. R. Billinton and R. N. Allan. *Reliability evaluation of power systems; 2nd ed.* Springer, Boston, 1996.

19. R. Billinton and W. Li. *Reliability assessment of electric power systems using Monte Carlo methods*. Springer Science & Business Media, 1994.

20. M. H. Bollen. "The Smart Grid: Adapting the Power System to New Challenges". *Synthesis Lectures on Power Electronics* 2:1, 2011, pp. 1–180.

21. P. Bonami, A. Lodi, A. Tramontani, and S. Wiese. "On mathematical programming with indicator constraints". *Mathematical programming* 151:1, 2015, pp. 191–223.

22. J. Branke, K. Deb, K. Miettinen, and R. Słowiński. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Vol. 5252. Springer-Verlag, 2008. ISBN: 978-3-540-88907-6.

23. P. A. Brown. "Some key factors influencing data transmission rates in the power line environment when utilizing carrier frequencies above 1 MHz". In: *Proceedings of International Symposium on Power-line Communications and its Applications*. 1998, pp. 67–75.

24. R. E. Brown and A. P. Hanson. "Impact of two-stage service restoration on distribution reliability". *IEEE Transactions on Power Systems* 16:4, 2001, pp. 624–629.

25. R. E. Brown. *Electric Power Distribution Reliability*. CRC Press, 2017. DOI: 10.1201/9780849375682.

26. G. Cabrera, Á. Juan, D. Lázaro, J. M. Marquès, and I. Proskurnia. “A simulation-optimization approach to deploy Internet services in large-scale systems with user-provided resources”. *SIMULATION* 90:6, 2014, pp. 644–659. DOI: 10.1177/0037549714531350.

27. V. Calderaro, C. N. Hadjicostis, A. Piccolo, and P. Siano. “Failure identification in smart grids based on petri net modeling”. *IEEE Transactions on Industrial Electronics* 58:10, 2011, pp. 4613–4623.

28. G. Celli, E. Ghiani, F. Pilo, and G. G. Soma. “Reliability assessment in smart distribution networks”. *Electric Power Systems Research* 104, 2013, pp. 164–175. ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2013.07.001.

29. C. Chen, S. Pramanik, Q. Zhu, W. Alok, and G. Qian. “The C-ND Tree: A Multidimensional Index for Hybrid Continuous and Non-ordered Discrete Data Spaces”. In: *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*. EDBT ’09. ACM, Saint Petersburg, Russia, 2009, pp. 462–471. ISBN: 978-1-60558-422-5. DOI: 10.1145/1516360.1516414.

30. M. Chica, A. A. Juan Pérez, O. Cordon, and D. Kelton. “Why simheuristics? Benefits, limitations, and best practices when combining metaheuristics with simulation”. *Benefits, Limitations, and Best Practices When Combining Metaheuristics with Simulation (January 1, 2017)*, 2017.

31. D. Chicco. “Ten quick tips for machine learning in computational biology”. *BioData Mining* 10:1, 2017, p. 35. ISSN: 1756-0381. DOI: 10.1186/s13040-017-0155-3.

32. A. Chowdhury and D. Koval. *Power Distribution System Reliability: Practical Methods and Applications*. Wiley-IEEE Press, 2009. ISBN: 0470292288.

33. R. Corrêa, G. Cardoso, O. C. de Araújo, and L. Mariotto. “Online coordination of directional overcurrent relays using binary integer programming”. *Electric Power Systems Research* 127, 2015, pp. 118–125. ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2015.05.017.

34. Creos Luxembourg S.A. *Le réseau de transport d' électricité de Creos Luxembourg S.A. (French) [Creos Luxembourg's electricity transmission network]*. 2021.

35. G. A. Croes. "A Method for Solving Traveling-Salesman Problems". *Operations Research* 6:6, 1958, pp. 791–812. DOI: 10.1287/opre.6.6.791. eprint: https://doi.org/10.1287/opre.6.6.791.

36. E. Dehghanpour, H. Kazemi Karegar, R. Kheirollahi, and T. Soleymani. "Optimal Coordination of Directional Overcurrent Relays in Microgrids by Using Cuckoo-Linear Optimization Algorithm and Fault Current Limiter". *IEEE Transactions on Smart Grid* 9:2, 2018, pp. 1365–1375. ISSN: 1949-3053. DOI: 10.1109/TSG.2016.2587725.

37. R. Deng, Z. Yang, M. Chow, and J. Chen. "A Survey on Demand Response in Smart Grids: Mathematical Models and Approaches". *IEEE Transactions on Industrial Informatics* 11:3, 2015, pp. 570–582. DOI: 10.1109/TII.2015.2414719.

38. E. Dialynas. *Analysi Axiopistias Technologikon Systimaton. (Greek) [Reliability Analysis of Technological Systems]*. Symeon, Athens, Greece, 1989. ISBN: 9789607888419.

39. E. Dialynas. *Axiopistia Systimaton Ilektrikis Energeias. (Greek) [Reliability of Electrical Power Systems]*. National Technical University of Athens, Athens, Greece, 1992.

40. R. Durstenfeld. "Algorithm 235: Random Permutation". *Communications of the ACM* 7:7, 1964, pp. 420–.

41. T. E. Dy Liacco. "The Adaptive Reliability Control System". *IEEE Transactions on Power Apparatus and Systems* PAS-86:5, 1967, pp. 517–531. DOI: 10.1109/TPAS.1967.291728.

42. M. A. Elgenedy, A. M. Massoud, and S. Ahmed. "Smart grid self-healing: Functions, applications, and developments". In: *1st SGRE*. IEEE, 2015, pp. 1–6.

43. H. Falaghi and M. .-. Haghifam. "Distributed Generation Impacts on Electric Distribution Systems Reliability: Sensitivity Analysis". In: *EUROCON 2005 - The International Conference on "Computer as a Tool"*. Vol. 2. 2005, pp. 1465–1468.

44. H. Farhangi. "The path of the smart grid". *IEEE power and energy magazine* 8:1, 2010.

45. G. Figueira and B. Almada-Lobo. "Hybrid simulation–optimization methods: A taxonomy and discussion". *Simulation Modelling Practice and Theory* 46, 2014. Simulation-Optimization of Complex Systems: Methods and Applications, pp. 118–134. ISSN: 1569-190X. DOI: https://doi.org/10.1016/j.simpat.2014.03.007.

46. C. Fikar, A. A. Juan, E. Martinez, and P. Hirsch. "A discrete-event driven metaheuristic for dynamic home service routing with synchronised trip sharing". *European Journal of Industrial Engineering* 10:3, 2016, pp. 323–340.

47. R. A. Fisher and F. Yates. *Statistical tables for biological, agricultural and medical research*. 3rd, rev. and enl. Oliver and Boyd, London, 1949.

48. M. C. Fu. "Feature Article: Optimization for simulation: Theory vs. Practice". *INFORMS Journal on Computing* 14:3, 2002, pp. 192–215. DOI: 10.1287/ijoc.14.3.192.113. eprint: https://doi.org/10.1287/ijoc.14.3.192.113.

49. S. Galli, A. Scaglione, and Z. Wang. "For the grid and through the grid: The role of power line communications in the smart grid". *Proceedings of the IEEE* 99:6, 2011, pp. 998–1027.

50. J. M. Gers. *Distribution System Analysis and Automation*. Power and Energy. Institution of Engineering and Technology, 2013.

51. A. Ghassemi, P. Goudarzi, M. R. Mirsarraf, and T. A. Gulliver. "A Stochastic Approach to Energy Cost Minimization in Smart-Grid-Enabled Data Center Network". *Journal of Computer Networks and Communications* 2019, 2019, p. 4390917. ISSN: 2090-7141. DOI: 10.1155/2019/4390917.

52. F. Glover, J. P. Kelly, and M. Laguna. "New Advances and Applications of Combining Simulation and Optimization". In: *Proceedings of the 28th Conference on Winter Simulation*. WSC '96. IEEE Computer Society, Coronado, California, USA, 1996, pp. 144–152. ISBN: 0780333837. DOI: 10.1145/256562.256595.

53. G. Golub and C. Reinsch. "Singular value decomposition and least squares solutions". *Numerische Mathematik* 14:5, 1970, pp. 403–420.

54. S. González-Martín, D. Riera, A. Juan, M. Elizondo, and P. Fonseca i Casas. "Sim-RandSHARP: A hybrid algorithm for solving the Arc Routing Problem with Stochastic Demands". In: 2012, pp. 1–11. ISBN: 978-1-4673-4779-2. DOI: 10.1109/WSC.2012.6465034.

55. M. Goyal, A. Ghosh, and F. Shahnia. "Overload prevention in an autonomous microgrid using battery storage units". In: *IEEE PES General Meeting | Conference Exposition*. 2014, pp. 1–5.

56. A. Gruler, C. Fikar, A. A. Juan, P. Hirsch, and C. Contreras-Bolton. "Supporting multi-depot and stochastic waste collection management in clustered urban areas via simulation–optimization". *Journal of Simulation* 11:1, 2017, pp. 11–19. ISSN: 1747-7786. DOI: 10.1057/s41273-016-0002-4.

57. A. Gruler, J. Panadero, J. de Armas, J. A. M. Pérez, and A. A. Juan. "A variable neighborhood search simheuristic for the multiperiod inventory routing problem with stochastic demands". *International Transactions in Operational Research* 27:1, 2020, pp. 314–335.

58. A. Gruler, J. Panadero, J. de Armas, J. A. Moreno Pérez, and A. A. Juan. "Combining variable neighborhood search with simulation for the inventory routing problem with stochastic demands and stock-outs". *Computers & Industrial Engineering* 123, 2018, pp. 278–288. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2018.06.036.

59. A. Gruler, C. L. Quintero-AraÃºjo, L. Calvet, and A. A. Juan. "Waste collection under uncertainty: a simheuristic based on variable neighbourhood search". *European Journal of Industrial Engineering* 11:2, 2017, pp. 228–255.

60. L. Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2020.

61. L. Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2021.

62. N. Hadjsaïd, J.-C. Sabonnadière, and J.-P. Angelier. *The Electrical Distribution Network: From Heritage to Innovation*. John Wiley & Sons, Ltd, 2013. ISBN: 9781118601280. DOI: 10.1002/9781118601280.

63. J. M. Hammersley and D. Handscomb. *Monte Carlo Methods*. Chapman and Hall, 1964.

64. J. Han and M. Piette. "Solutions for Summer Electric Power Shortages: Demand Response and its Applications in Air Conditioning and Refrigerating Systems". *Refrigeration, Air Conditioning, & Electric Power Machinery* 29:1, 2008, pp. 1–4.

65. P. Hansen and N. Mladenović. "An Introduction to Variable Neighborhood Search". In: *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Ed. by S. Voß, S. Martello, I. H. Osman, and C. Roucairol. Springer US, Boston, MA, 1999, pp. 433–458. ISBN: 978-1-4615-5775-3. DOI: 10.1007/978-1-4615-5775-3_30.

66. P. Hansen and N. Mladenović. "Developments of Variable Neighborhood Search". In: *Essays and Surveys in Metaheuristics*. Springer US, Boston, MA, 2002, pp. 415–439. ISBN: 978-1-4615-1507-4. DOI: 10.1007/978-1-4615-1507-4_19.

67. P. Hansen and N. Mladenović. "Variable neighborhood search: Principles and applications". *European Journal of Operational Research* 130:3, 2001, pp. 449–467. ISSN: 0377-2217. DOI: https://doi.org/10.1016/S0377-2217(00)00100-4.

68. P. Hansen, N. Mladenović, J. Brimberg, and J. A. M. Pérez. "Variable Neighborhood Search". In: *Handbook of Metaheuristics*. Ed. by M. Gendreau and J.-Y. Potvin. Springer International Publishing, Cham, 2019, pp. 57–97. ISBN: 978-3-319-91086-4. DOI: 10.1007/978-3-319-91086-4_3.

69. P. Hansen, N. Mladenović, R. Todosijević, and S. Hanafi. "Variable neighborhood search: basics and variants". *EURO Journal on Computational Optimization* 5:3, 2017, pp. 423–454.

70. P. E. S. Harald Held Gabriela Martinez. “Stochastic programming approach for energy management in electric microgrids”. *Numerical Algebra, Control & Optimization* 4:3, 2014, p. 241. ISSN: 2155-3289. DOI: 10.3934/naco.2014.4.241.

71. J. Hare, X. Shi, S. Gupta, and A. Bazzi. “Fault diagnostics in smart microgrids: A survey”. *Renewable and Sustainable Energy Reviews* 60, 2016, pp. 1114–1124.

72. T. Hartmann, F. Fouquet, J. Klein, Y. L. Traon, A. Pelov, L. Toutain, and T. Ropitault. “Generating realistic Smart Grid communication topologies based on real-data”. In: *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. 2014, pp. 428–433. DOI: 10.1109/SmartGridComm.2014.7007684.

73. T. Hartmann, A. Moawad, F. Fouquet, Y. Reckinger, J. Klein, and Y. Le Traon. “Near real-time electric load approximation in low voltage cables of smart grids with models@ run. time”. In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. ACM. 2016, pp. 2119–2126.

74. T. Hartmann, A. Moawad, F. Fouquet, Y. Reckinger, J. Klein, and Y. Le Traon. “Near real-time electric load approximation in low voltage cables of smart grids with models@run.time”. In: *Proc. of SAC 2016*. ACM Press, 2016, pp. 2119–2126.

75. T. Hartmann, A. Moawad, F. Fouquet, Y. Reckinger, T. Mouelhi, J. Klein, and Y. Le Traon. “Suspicious electric consumption detection based on multi-profiling using live machine learning”. In: *Smart Grid Communications (SmartGridComm), 2015 IEEE International Conference on*. IEEE. 2015, pp. 891–896.

76. M. He and J. Zhang. “Fault detection and localization in smart grid: A probabilistic dependence graph approach”. In: *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE. 2010, pp. 43–48.

77. Hongbin Li and Qing Zhao. "A cut/tie set method for reliability evaluation of control systems". In: *Proceedings of the 2005, American Control Conference, 2005*. 2005, 1048–1053 vol. 2.

78. J. Huang, H. Wang, and Y. Qian. "Smart grid communications in challenging environments". In: *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*. IEEE. 2012, pp. 552–557.

79. IEEE. "IEEE Guide for Electric Power Distribution Reliability Indices - Redline". *IEEE Std 1366-2012 (Revision of IEEE Std 1366-2003) - Redline*, 2012.

80. J. Sichi, J. Kinable, D. Michail, B. Naveh, and Contributors. *JGraphT - Graph Algorithms and Data Structures in Java (Version 1.3.0)*. http://www.jgrapht.org. 2018.

81. H. Jiang, J. J. Zhang, W. Gao, and Z. Wu. "Fault detection, identification, and location in smart grid based on data-driven computational methods". *IEEE Transactions on Smart Grid* 5:6, 2014, pp. 2947–2956.

82. A. Juan, J. Faulin, S. Grasman, D. Riera, J. Marull, and C. Mendez. "Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands". *Transportation Research Part C: Emerging Technologies* 19:5, 2011. Freight Transportation and Logistics (selected papers from ODYSSEUS 2009 - the 4th International Workshop on Freight Transportation and Logistics), pp. 751–765. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2010.09.007.

83. A. A. Juan, W. David Kelton, C. S. M. Currie, and J. Faulin. "SIMHEURISTICS APPLICATIONS: DEALING WITH UNCERTAINTY IN LOGISTICS, TRANSPORTATION, AND OTHER SUPPLY CHAIN AREAS". In: *2018 Winter Simulation Conference (WSC)*. 2018, pp. 3048–3059.

84. A. A. Juan, J. Panadero, L. Reyes-Rubiano, J. Faulin, R. de la Torre, and I. Latorre. "Simulation-Based Optimization in Transportation and Logistics: Comparing Sample Average Approximation with Simheuristics". In: *2019 Winter Simulation Conference (WSC)*. 2019, pp. 1906–1917.

85. A. A. Juan, B. B. Barrios, E. Vallada, D. Riera, and J. Jorba. "A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times". *Simulation Modelling Practice and Theory* 46, 2014. Simulation-Optimization of Complex Systems: Methods and Applications, pp. 101–117. ISSN: 1569-190X. DOI: https://doi.org/10.1016/j.simpat.2014.02.005.

86. A. A. Juan, J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. "A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems". *Operations Research Perspectives* 2, 2015, pp. 62–72. ISSN: 2214-7160. DOI: https://doi.org/10.1016/j.orp.2015.03.001.

87. A. A. Juan, S. E. Grasman, J. Caceres-Cruz, and T. Bektaş. "A simheuristic algorithm for the Single-Period Stochastic Inventory-Routing Problem with stock-outs". *Simulation Modelling Practice and Theory* 46, 2014. Simulation-Optimization of Complex Systems: Methods and Applications, pp. 40–52. ISSN: 1569-190X. DOI: https://doi.org/10.1016/j.simpat.2013.11.008.

88. S. Kher, V. Nutt, D. Dasgupta, H. Ali, and P. Mixon. "A prediction model for anomalies in smart grid with sensor network". In: *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*. ACM. 2013, p. 60.

89. D. Kolbe, Q. Zhu, and S. Pramanik. "Efficient k-nearest neighbor searching in nonordered discrete data spaces". *ACM Transactions on Information Systems (TOIS)* 28:2, 2010, p. 7.

90. P. Kundur, N. Balu, and M. Lauby. *Power System Stability and Control*. EPRI power system engineering series. McGraw-Hill Education, 1994. ISBN: 978-0-07-035958-1.

91. J. Kytöjoki, T. Nuortio, O. Bräysy, and M. Gendreau. "An efficient variable neighborhood search heuristic for very large scale vehicle routing problems". *Computers & Operations Research* 34:9, 2007, pp. 2743–2757. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2005.10.010.

92. L. Lampe, A. M. Tonello, and T. G. Swart. *Power Line Communications: Principles, Standards and Applications from multimedia to smart grid*. John Wiley & Sons, 2016.

93. H. Lin, J. M. Guerrero, C. Jia, Z.-h. Tan, J. C. Vasquez, and C. Liu. "Adaptive overcurrent protection for microgrids in extensive distribution systems". In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. 2016, pp. 4042–4047. DOI: 10.1109/IECON.2016.7793091.

94. Z. Lü, F. Glover, and J.-K. Hao. "Neighborhood combination for unconstrained binary quadratic programming". *MIC* 2009, 2010.

95. G. Mavromatidis, K. Orehounig, and J. Carmeliet. "Design of distributed energy systems under uncertainty: A two-stage stochastic programming approach". *Applied Energy* 222, 2018, pp. 932–950. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.04.019.

96. D. Mazza, A. Pagès-Bernaus, D. Tarchi, A. A. Juan, and G. E. Corazza. "Supporting Mobile Cloud Computing in Smart Cities via Randomized Algorithms". *IEEE Systems Journal* 12:2, 2018, pp. 1598–1609.

97. N. Mladenović and P. Hansen. "Variable neighborhood search". *Computers & Operations Research* 24:11, 1997, pp. 1097–1100.

98. S. A. Nasar. *Schaum's outline of theory and problems of electric power systems*. McGraw-Hill, 1990. ISBN: 0070459177.

99. T. Niknam, R. Azizipanah-Abarghooee, and M. R. Narimani. "An efficient scenario-based stochastic programming framework for multi-objective optimal micro-grid operation". *Applied Energy* 99, 2012, pp. 455–470. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2012.04.017.

100. S. Ólafsson. "Chapter 21 Metaheuristics". In: *Simulation*. Ed. by S. G. Henderson and B. L. Nelson. Vol. 13. Handbooks in Operations Research and Management Science. Elsevier, 2006, pp. 633–654. DOI: https://doi.org/10.1016/S0927-0507(06)13021-2.

101. J. Panadero, J. Doering, R. Kizys, A. A. Juan, and A. Fito. "A variable neighborhood search simheuristic for project portfolio selection under uncertainty". *Journal of Heuristics* 26:3, 2020, pp. 353–375. ISSN: 1572-9397. DOI: 10.1007/s10732-018-9367-z.

102. M. Panteli and P. Mancarella. "Influence of extreme weather and climate change on the resilience of power systems: Impacts and possible mitigation strategies". *Electric Power Systems Research* 127, 2015, pp. 259–270. DOI: http://dx.doi.org/10.1016/j.epsr.2015.06.012.

103. E. Pashajavid, F. Shahnia, and A. Ghosh. "Overload management of autonomous microgrids". In: *11th IEEE PEDS*. 2015, pp. 73–78.

104. R. Pasupathy and Y. Song. "An Adaptive Sequential Sample Average Approximation Framework for Solving Two-stage Stochastic Programs". *Under Review*, 2019. (Last time accessed on Sunday 29 November). eprint: http://www.optimization-online.org/DB_HTML/2019/02/7068.html.

105. K. Paton. "An Algorithm for Finding a Fundamental Set of Cycles of a Graph". *Communications of the ACM* 12:9, 1969, pp. 514–518.

106. Peng Wang and R. Billinton. "Reliability cost/worth assessment of distribution systems incorporating time-varying weather conditions and restoration resources". *IEEE Transactions on Power Delivery* 17:1, 2002, pp. 260–265.

107. M. V. F. Pereira and N. J. Balu. "Composite generation/transmission reliability evaluation". *Proceedings of the IEEE* 80:4, 1992, pp. 470–491.

108. G. Qian, Q. Zhu, Q. Xue, and S. Pramanik. "Dynamic Indexing for Multidimensional Non-ordered Discrete Data Spaces Using a Data-partitioning Approach". *ACM Trans. Database Syst.* 31:2, 2006, pp. 439–484. ISSN: 0362-5915. DOI: 10.1145/1138394.1138395.

109. G. Qian, Q. Zhu, Q. Xue, and S. Pramanik. "The ND-tree: A Dynamic Indexing Technique for Multidimensional Non-ordered Discrete Data Spaces". In: *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29*. VLDB '03. VLDB Endowment, Berlin, Germany, 2003, pp. 620–631. ISBN: 0-12-722442-4.

110. C. L. Quintero-Araujo, D. Guimarans, and A. A. Juan. "A simheuristic algorithm for the capacitated location routing problem with stochastic demands". *to appear in Journal of Simulation*, 2019.

111. D. Raba, A. Estrada-Moreno, J. Panadero, and A. A. Juan. "A reactive simheuristic using online data for a real-life inventory routing problem with stochastic demands". *Intern. Transactions in Operational Research* 27:6, 2020, pp. 2785–2816.

112. N. A. Rahim and S. Mekhilef. "Implementation of three-phase grid connected inverter for photovoltaic solar power generation system". In: *Proceedings. International Conference on Power System Technology*. Vol. 1. 2002, 570–573 vol.1. DOI: 10.1109/ICPST.2002.1053607.

113. P. C. Ramaswamy and G. Deconinck. "Relevance of voltage control, grid reconfiguration and adaptive protection in smart grids and genetic algorithm as an optimization tool in achieving their control objectives". In: *ICNSC*. 2011, pp. 26–31.

114. S. Ramseler, M. Arzberger, and A. Hauser. "MV and LV powerline communications: New proposed IEC standards". In: *Transmission and Distribution Conference, 1999 IEEE*. Vol. 1. IEEE. 1999, pp. 235–239.

115. K. Razazian, M. Umari, A. Kamalizad, V. Loginov, and M. Navid. "G3-PLC specification for powerline communication: Overview, system simulation and field trial results". In: *Power Line Communications and Its Applications (ISPLC), 2010 IEEE International Symposium on*. IEEE. 2010, pp. 313–318.

116. S. S. Reddy, V. Sandeep, and C.-M. Jung. "Review of stochastic optimization methods for smart grid". *Frontiers in Energy* 11:2, 2017, pp. 197–209. ISSN: 2095-1698. DOI: 10.1007/s11708-017-0457-7.

117. L. S. Reyes-Rubiano, J. Faulin, L. Calvet, and A. A. Juan. "A simheuristic approach for freight transportation in smart cities". In: *2017 Winter Simulation Conference (WSC)*. 2017, pp. 3346–3357.

118. C. C. Ribeiro, V. Maniezzo, T. Stützle, C. Blum, A. A. Juan, H. Ramalhinho, N. Mladenović, A. Sifaleras, K. Sörensen, and M. J. Souza. “Preface to the Special Issue on Matheuristics and Metaheuristics”. *International Transactions in Operational Research* 27:1, 2020, pp. 5–8.

119. S. M. Ross. *A Course in Simulation*. Maxwell Macmilllan, 1991.

120. S. Ross. “Chapter 9 - Variance Reduction Techniques”. In: *Simulation (Fifth Edition)*. Ed. by S. Ross. Fifth Edition. Academic Press, 2013, pp. 153–231. ISBN: 978-0-12-415825-2. DOI: https://doi.org/10.1016/B978-0-12-415825-2.00009-7.

121. S. M. Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. 6th ed. Academic Press, 2020. ISBN: 9780128177464.

122. A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. O’Hare. “Real-time recognition and profiling of appliances through a single electricity sensor”. In: *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*. IEEE. 2010, pp. 1–9.

123. M. Saffari, M. S. Misaghian, M. Kia, A. Heidari, D. Zhang, P. Dehghanian, and J. Aghaei. “Stochastic robust optimization for smart grid considering various arbitrage opportunities”. *Electric Power Systems Research* 174, 2019, p. 105847. ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2019.04.025.

124. P. Samadi, H. Mohsenian-Rad, V. W. Wong, and R. Schober. “Real-time pricing for demand response based on stochastic approximation”. *IEEE Transactions on Smart Grid* 5:2, 2014, pp. 789–798.

125. L. Selander, T. Mortensen, and G. Lindell. “Load profile and communication channel characteristics of the low voltage grid”. *Proc. DistribuTECH DA/DSM Europe* 98, 1998.

126. F. Shahnia, S. Bourbour, and A. Ghosh. “Coupling Neighboring Microgrids for Overload Management Based on Dynamic Multicriteria Decision-Making”. *IEEE Transactions on Smart Grid* 8:2, 2017, pp. 969–983.

127. J. G. Shanthikumar and R. G. Sargent. "A Unifying View of Hybrid Simulation/Analytic Models and Modeling". *Operations Research* 31:6, 1983, pp. 1030–1052. DOI: 10.1287/opre.31.6.1030. eprint: https://doi.org/10.1287/opre.31.6.1030.

128. A. Shapiro and A. Philpott. "A tutorial on stochastic programming". *Manuscript. Available at www2.isye.gatech.edu/ashapiro/publications.html* 17, 2007.

129. A. Sifaleras, K. Paparrizos, and V. F. Demyanov. "Advances in discrete optimization". *Optimization* 62:8, 2013, pp. 1003–1006.

130. A. Sifaleras, S. Salhi, and J. Brimberg, eds. *Variable Neighborhood Search - ICVNS 2018*. Vol. 11328. Lecture Notes in Computer Science. Springer, Cham, 2019.

131. H. P. Simão, H. B. Jeong, B. Defourny, W. B. Powell, A. Boulanger, A. Gagneja, L. Wu, and R. N. Anderson. "A Robust Solution to the Load Curtailment Problem". *IEEE Transactions on Smart Grid* 4:4, 2013, pp. 2209–2219.

132. C. Singh, P. Jirutitijaroen, and J. Mitra. *Electric Power Grid Reliability Evaluation: Models and Methods*. John Wiley & Sons, Ltd, 2018. ISBN: 978-1-119-53677-2.

133. J. Soares, M. A. Fotouhi Ghazvini, N. Borges, and Z. Vale. "A stochastic model for energy resources management considering demand response in smart grids". *Electric Power Systems Research* 143, 2017, pp. 599–610. ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2016.10.056.

134. K. Sun, Y. Hou, W. Sun, and J. Qi. *Power System Control Under Cascading Failures: Understanding, Mitigation, and System Restoration*. John Wiley & Sons, Ltd, 2018. ISBN: 9781119282075.

135. J. R. Swisher, P. D. Hyden, S. H. Jacobson, and L. W. Schruben. "Simulation Optimization: A Survey of Simulation Optimization Techniques and Procedures". In: *Proceedings of the 32nd Conference on Winter Simulation*. WSC '00. Society for Computer Simulation International, Orlando, Florida, 2000, pp. 119–128. ISBN: 0780365828.

136. R. Tarjan. "Depth-First Search and Linear Graph Algorithms". *SIAM Journal on Computing* 1:2, 1972, pp. 146–160.

137. E. Tekin and I. Sabuncuoglu. "Simulation optimization: A comprehensive review on theory and applications". *IIE Transactions* 36:11, 2004, pp. 1067–1081. DOI: 10.1080/07408170490500654. eprint: https://doi.org/10.1080/07408170490500654.

138. M. Todinov. *Risk-Based Reliability Analysis and Generic Principles for Risk Reduction*. Elsevier, Oxford, 2007. ISBN: 978-0-08-044728-5. DOI: 10.1016/b978-0-08-044728-5.x5000-8.

139. A. Valdes, R. Macwan, and M. Backes. "Anomaly Detection in Electrical Substation Circuits via Unsupervised Machine Learning". In: *Information Reuse and Integration (IRI), 2016 IEEE 17th International Conference on*. IEEE. 2016, pp. 500–505.

140. G. K. Venayagamoorthy. "Dynamic, Stochastic, Computational, and Scalable Technologies for Smart Grids". *IEEE Computational Intelligence Magazine* 6:3, 2011, pp. 22–35. DOI: 10.1109/MCI.2011.941588.

141. P. A. Villarinho, J. Panadero, L. S. Pessoa, A. A. Juan, and F. L. C. Oliveira. "A simheuristic algorithm for the stochastic permutation flow-shop problem with delivery dates and cumulative payoffs". *International Transactions in Operational Research* 28:2, 2021, pp. 716–737.

142. N. Vovos and G. Giannakopoulos. *Eisagogi sta Systimata Ilektrikis Energeias. (Greek), [Introduction to Electricity Systems]*. Ziti, Thessaloniki, Greece, 2008. ISBN: 978-960-456-105-6.

143. S. W. Wallace and S.-E. Fleten. "Stochastic Programming Models in Energy". In: *Stochastic Programming*. Vol. 10. Handbooks in Operations Research and Management Science. Elsevier, 2003, pp. 637–677. DOI: https://doi.org/10.1016/S0927-0507(03)10010-2.

144. L.-F. Wang and L.-Y. Shi. "Simulation Optimization: A Review on Theory and Applications". *Acta Automatica Sinica* 39:11, 2013, pp. 1957–1968. ISSN: 1874-1029. DOI: https://doi.org/10.1016/S1874-1029(13)60081-6.

145. S. Wang, H. Gangammanavar, S. D. Ekşioğlu, and S. J. Mason. "Stochastic Optimization for Energy Management in Power Systems With Multiple Microgrids". *IEEE Transactions on Smart Grid* 10:1, 2019, pp. 1068–1079. DOI: 10.1109/TSG.2017.2759159.

146. D. M. Ward. "The effect of weather on grid systems and the reliability of electricity supply". *Climatic change* 121:1, 2013, pp. 103–113.

147. I. Xyngi and M. Popov. "Smart protection in Dutch Medium Voltage distributed generation systems". In: *2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*. 2010, pp. 1–8. DOI: 10.1109/ISGTEUROPE.2010.5638982.

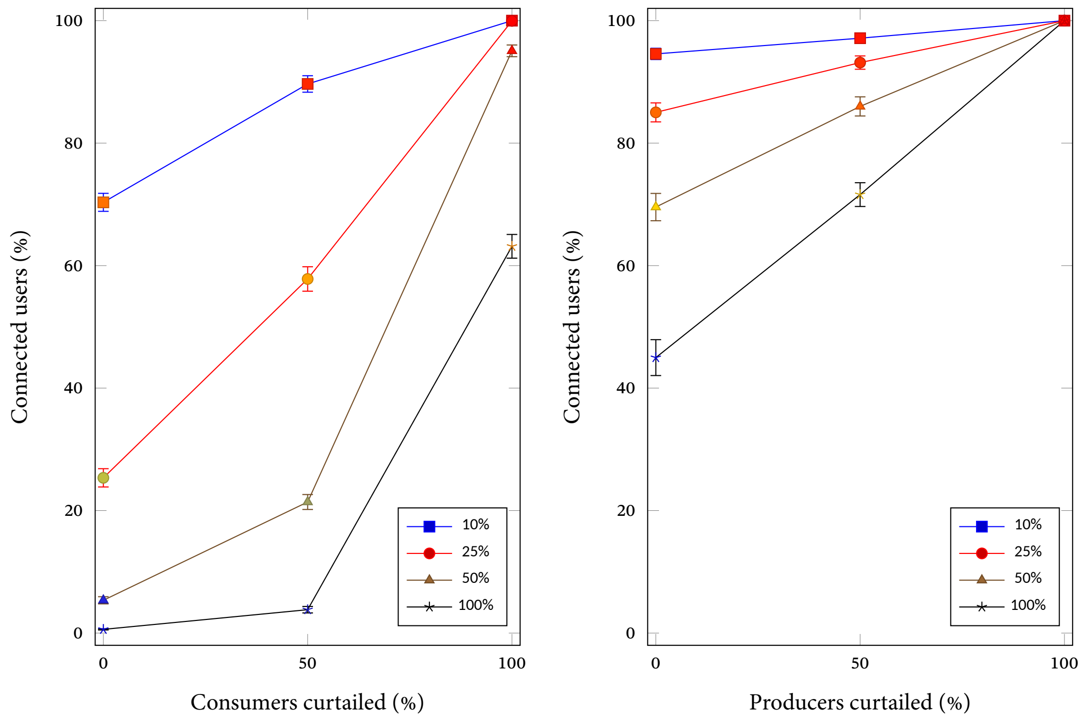
148. G. Zhang, Z. Li, B. Zhang, and W. A. Halang. "Understanding the cascading failures in Indian power grids with complex networks theory". *Physica A: Statistical Mechanics and its Applications* 392:15, 2013, pp. 3273–3280. ISSN: 0378-4371.

Appendices

A Detailed Results of the Deterministic OPP

A.1 Connected Users

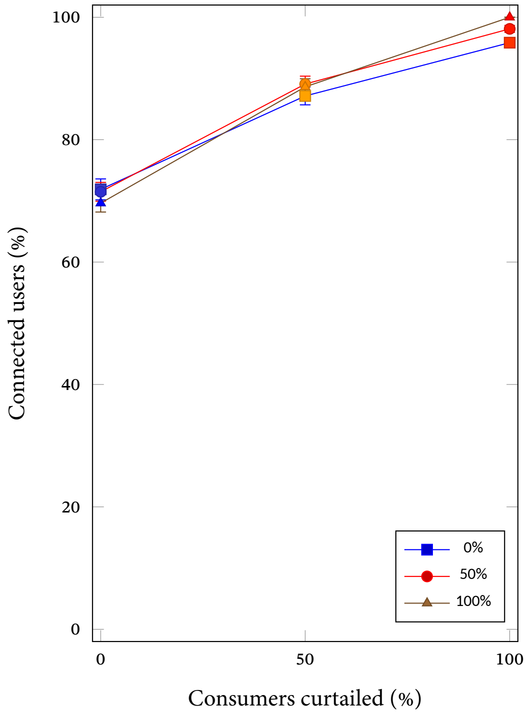
The percentage of the connected users is presented; the higher, the better.



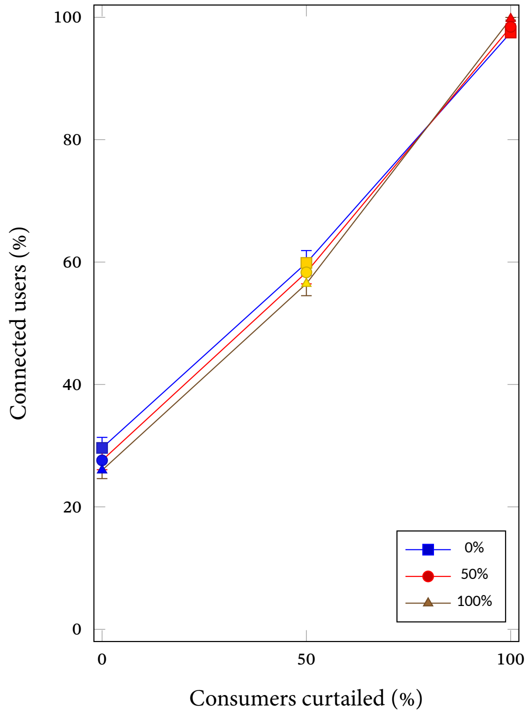
(a) None of the producers overloaded. Different percentage of consumers overload.

(b) None of the consumers overloaded. Different percentage of producers overload.

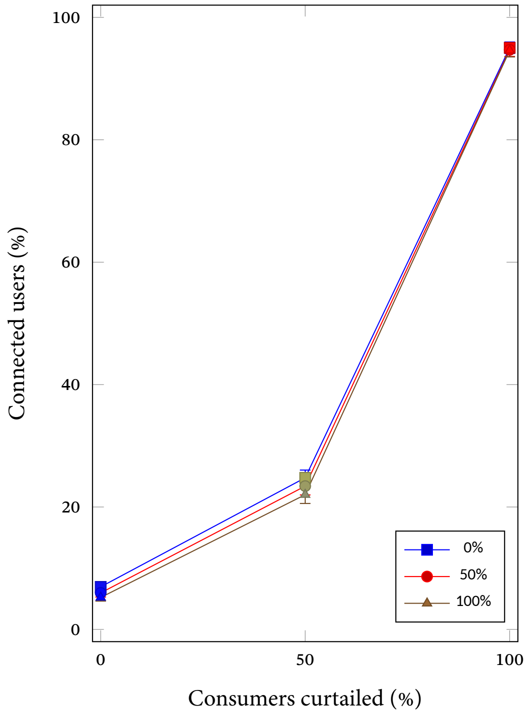
Figure A.1: Connected users (%)



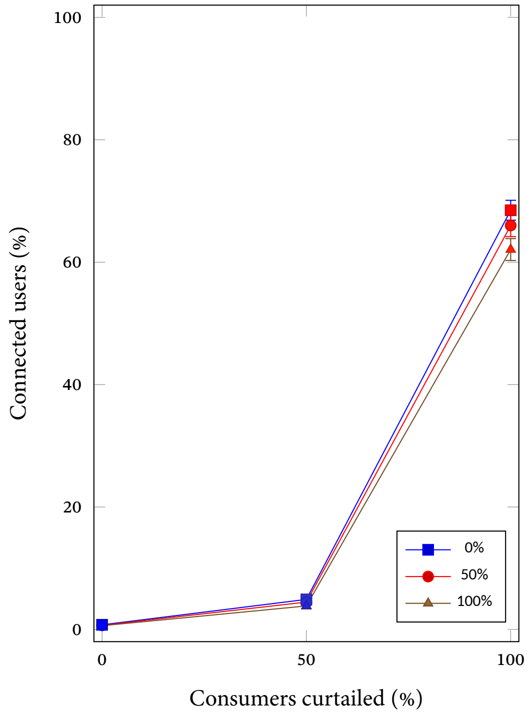
(a) 10% of consumers overload. Different percentage of curtailed producers.



(b) 25% of consumers overload. Different percentage of curtailed producers.

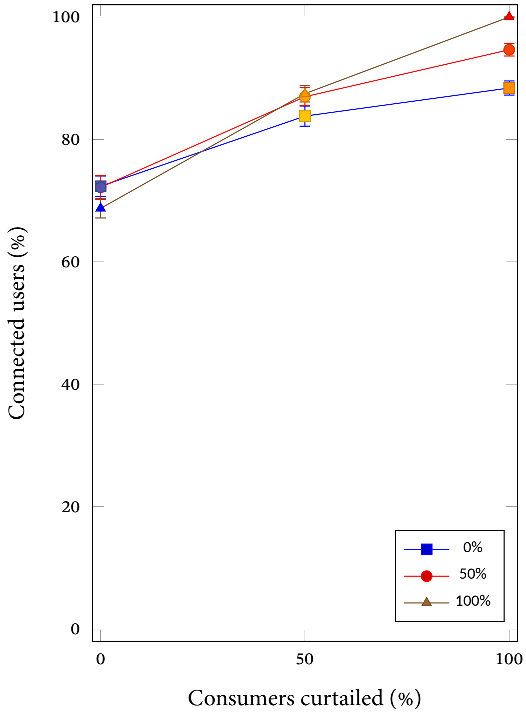


(c) 50% of consumers overload. Different percentage of curtailed producers.

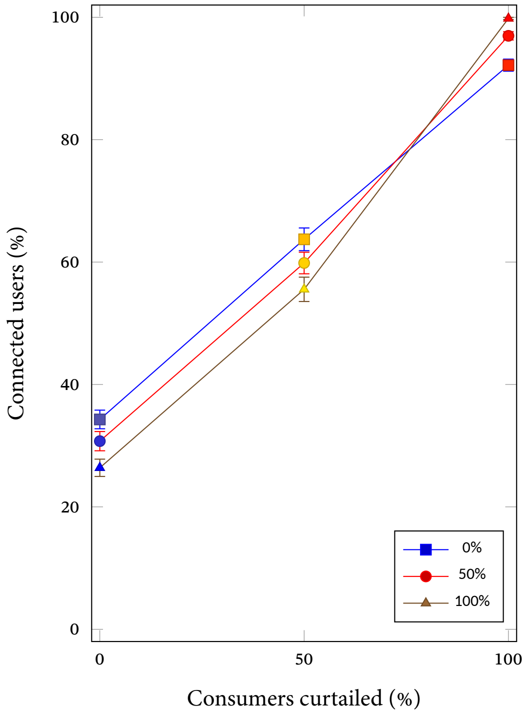


(d) 100% of consumers overload. Different percentage of curtailed producers.

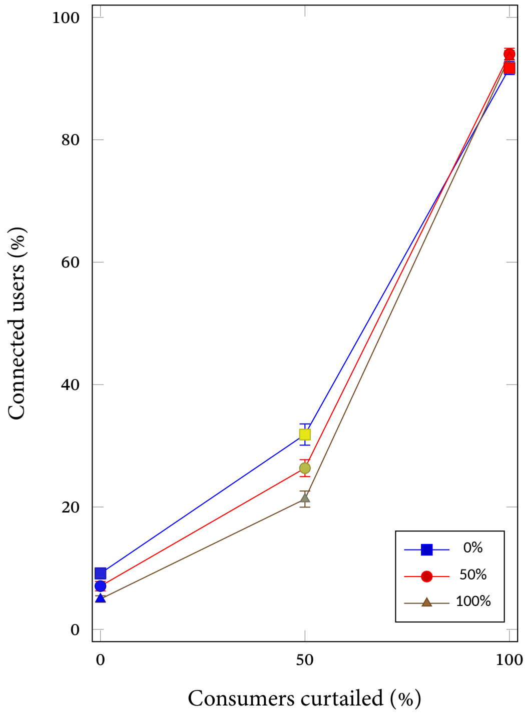
Figure A.2: Connected users (%) (10% of producers overload)



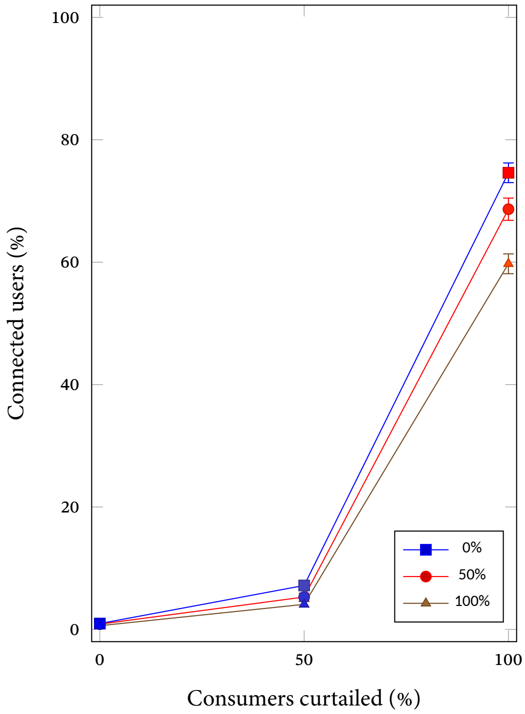
(a) 10% of consumers overload. Different percentage of curtailed producers.



(b) 25% of consumers overload. Different percentage of curtailed producers.

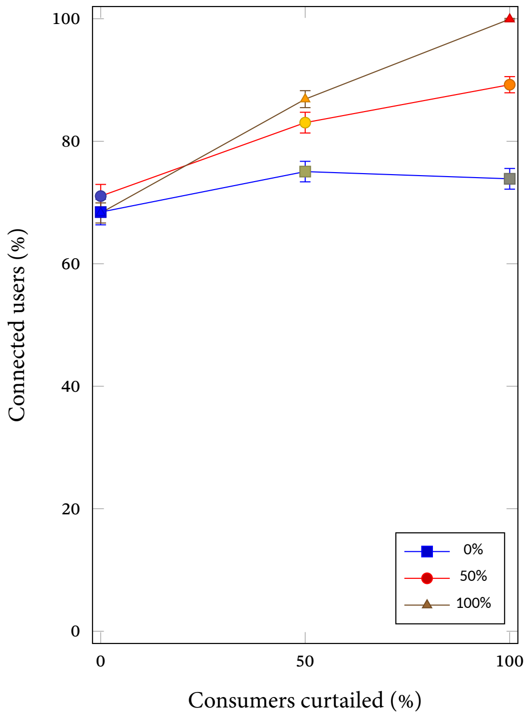


(c) 50% of consumers overload. Different percentage of curtailed producers.

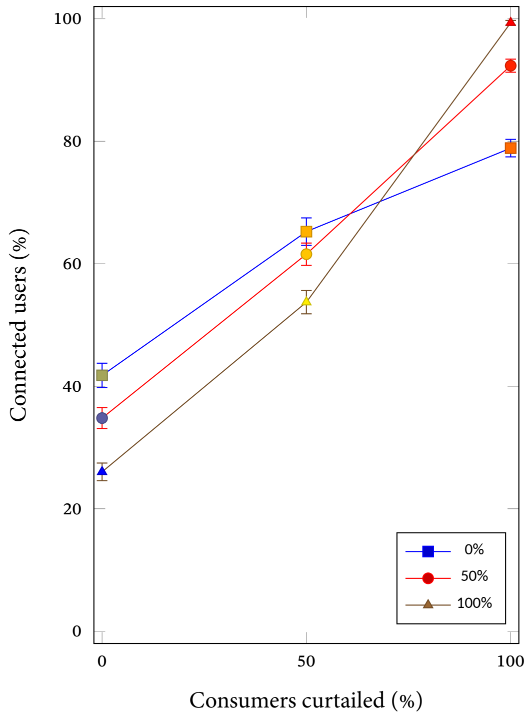


(d) 100% of consumers overload. Different percentage of curtailed producers.

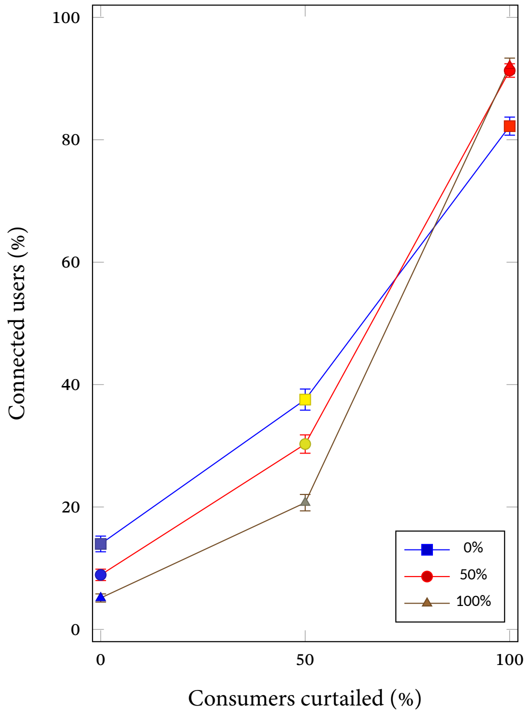
Figure A.3: Connected users (%) (25% of producers overload)



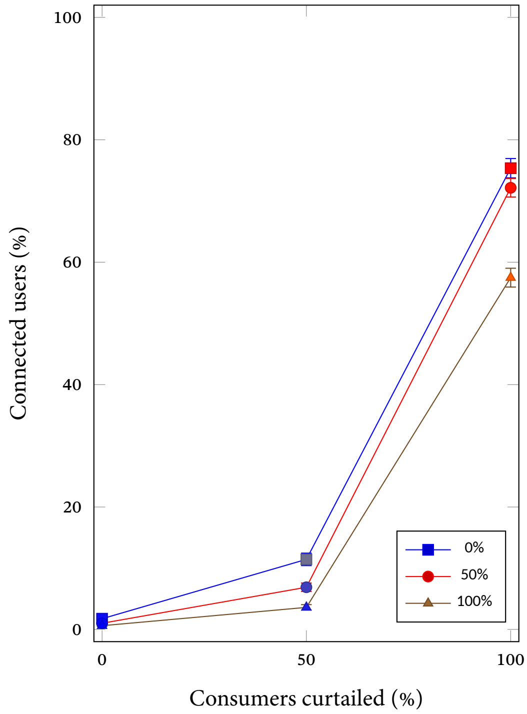
(a) 10% of consumers overload. Different percentage of curtailed producers.



(b) 25% of consumers overload. Different percentage of curtailed producers.

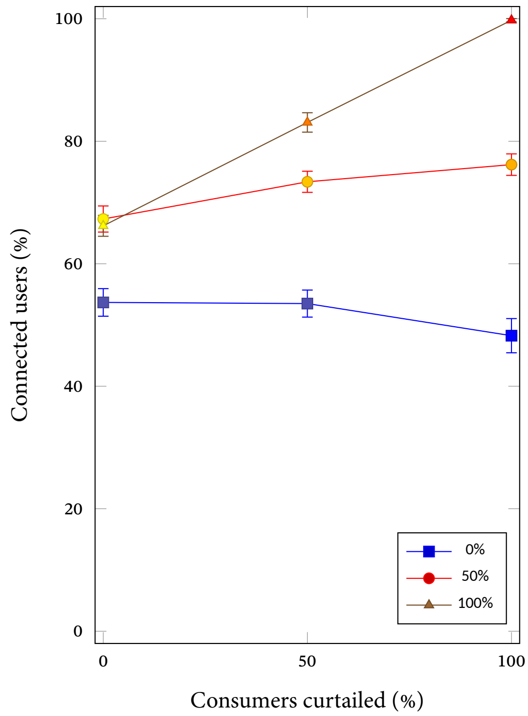


(c) 50% of consumers overload. Different percentage of curtailed producers.

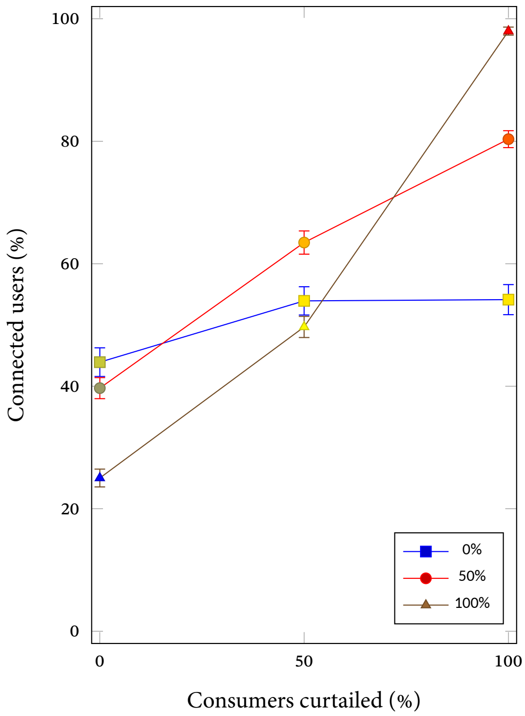


(d) 100% of consumers overload. Different percentage of curtailed producers.

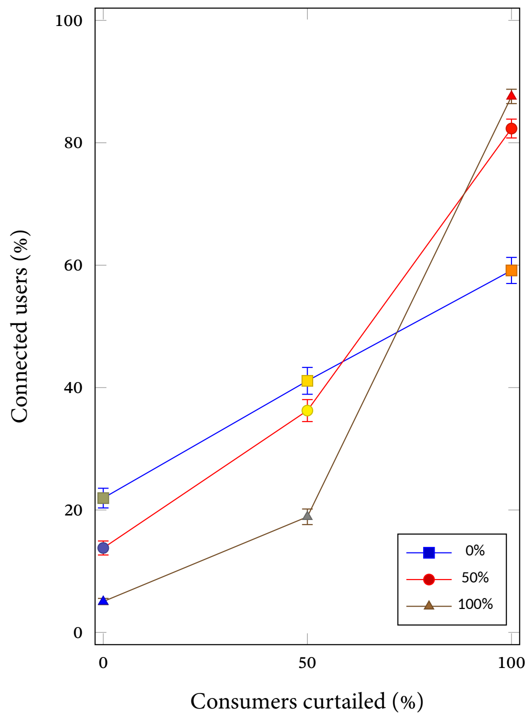
Figure A.4: Connected users (%) (50% of producers overload)



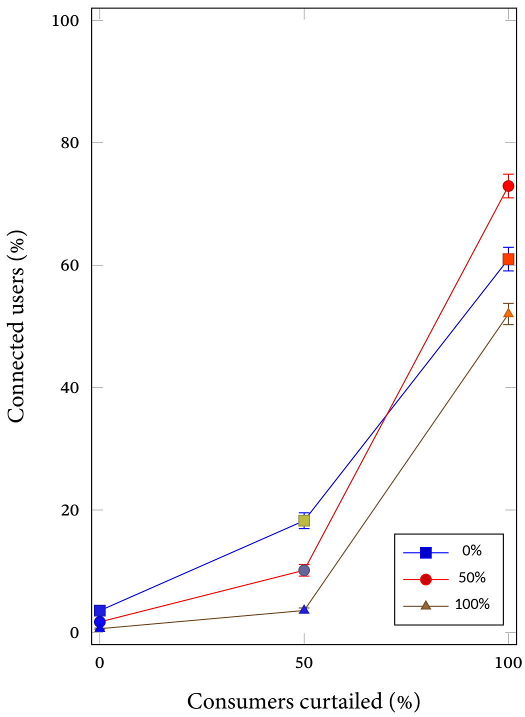
(a) 10% of consumers overload. Different percentage of curtailed producers.



(b) 25% of consumers overload. Different percentage of curtailed producers.



(c) 50% of consumers overload. Different percentage of curtailed producers.

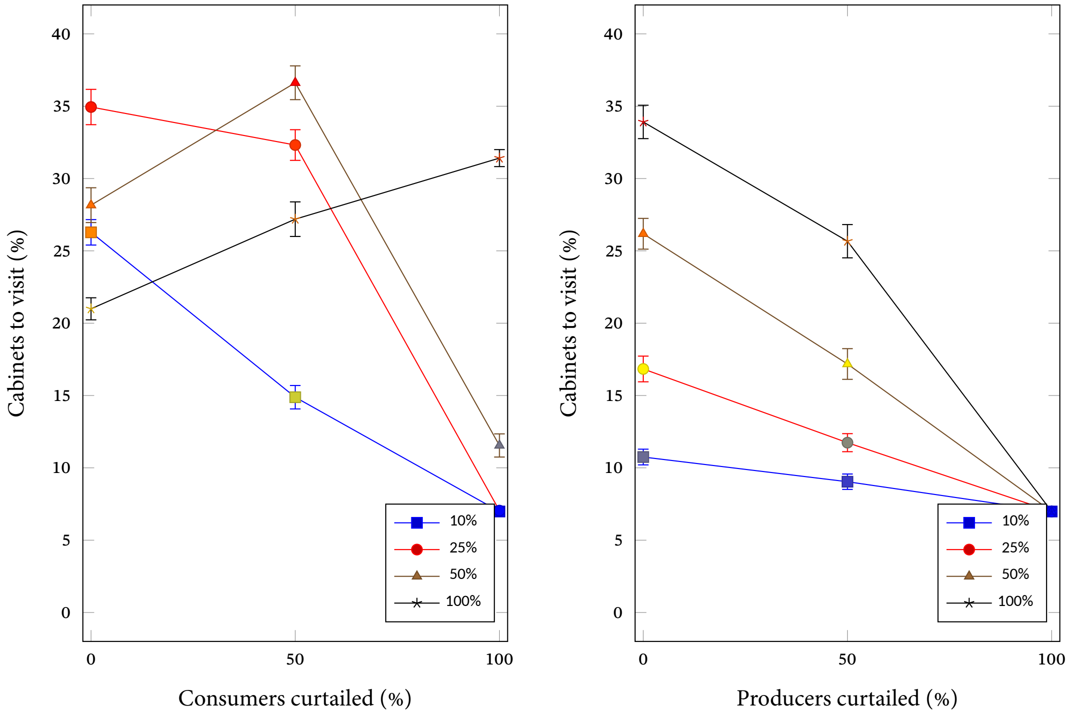


(d) 100% of consumers overload. Different percentage of curtailed producers.

Figure A.5: Connected users (%) (100% of producers overload)

A.2 Cabinets to visit

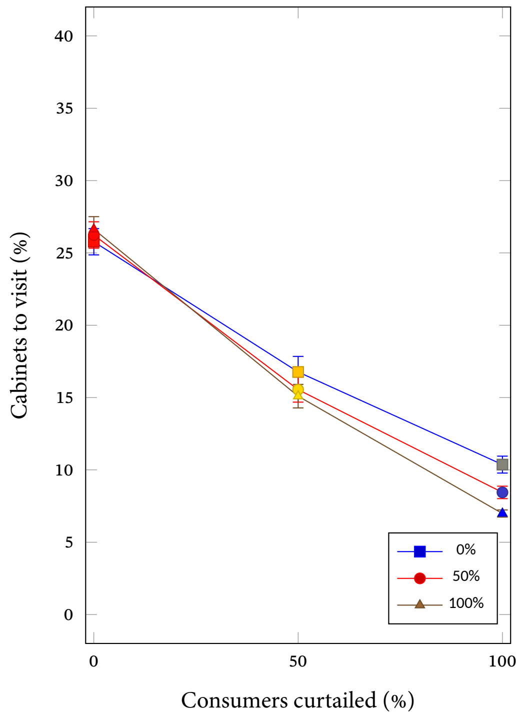
The percentage of the cabinets to visit is presented; the lower, the better.



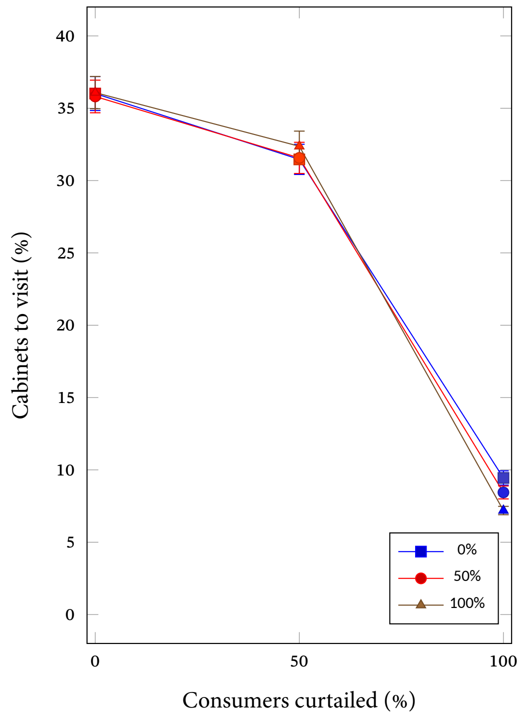
(a) None of the producers overloaded. Different percentage of consumers overload.

(b) None of the consumers overloaded. Different percentage of producers overload.

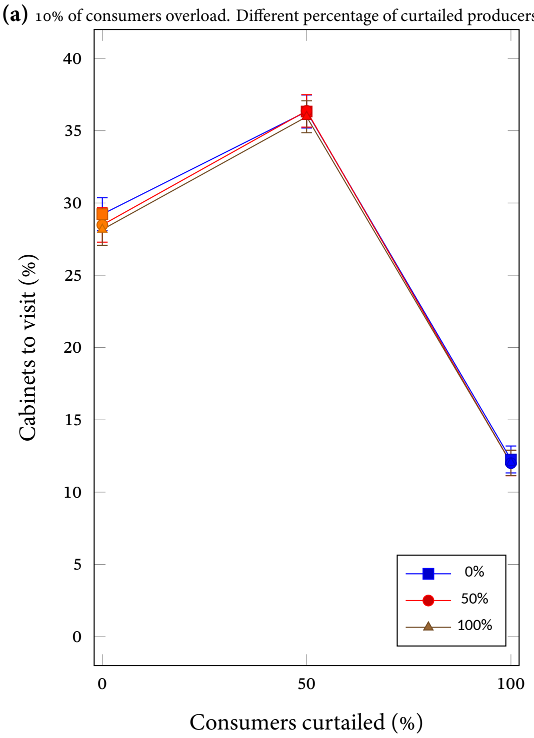
Figure A.6: Cabinets to visit (%)



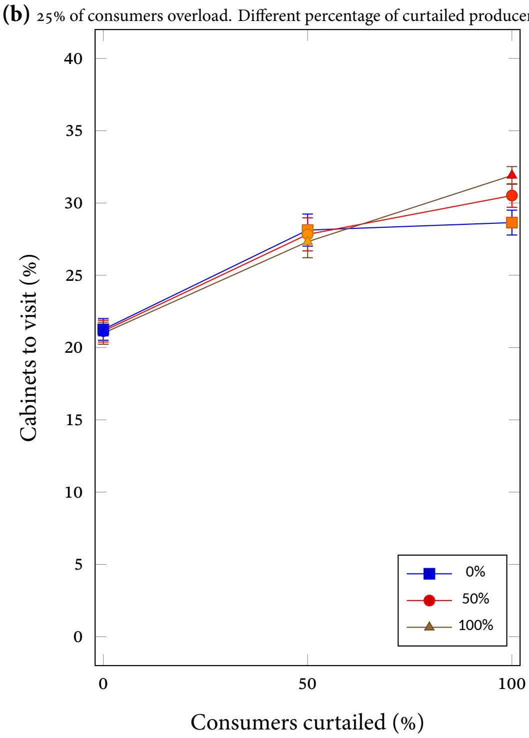
(a) 10% of consumers overload. Different percentage of curtailed producers.



(b) 25% of consumers overload. Different percentage of curtailed producers.

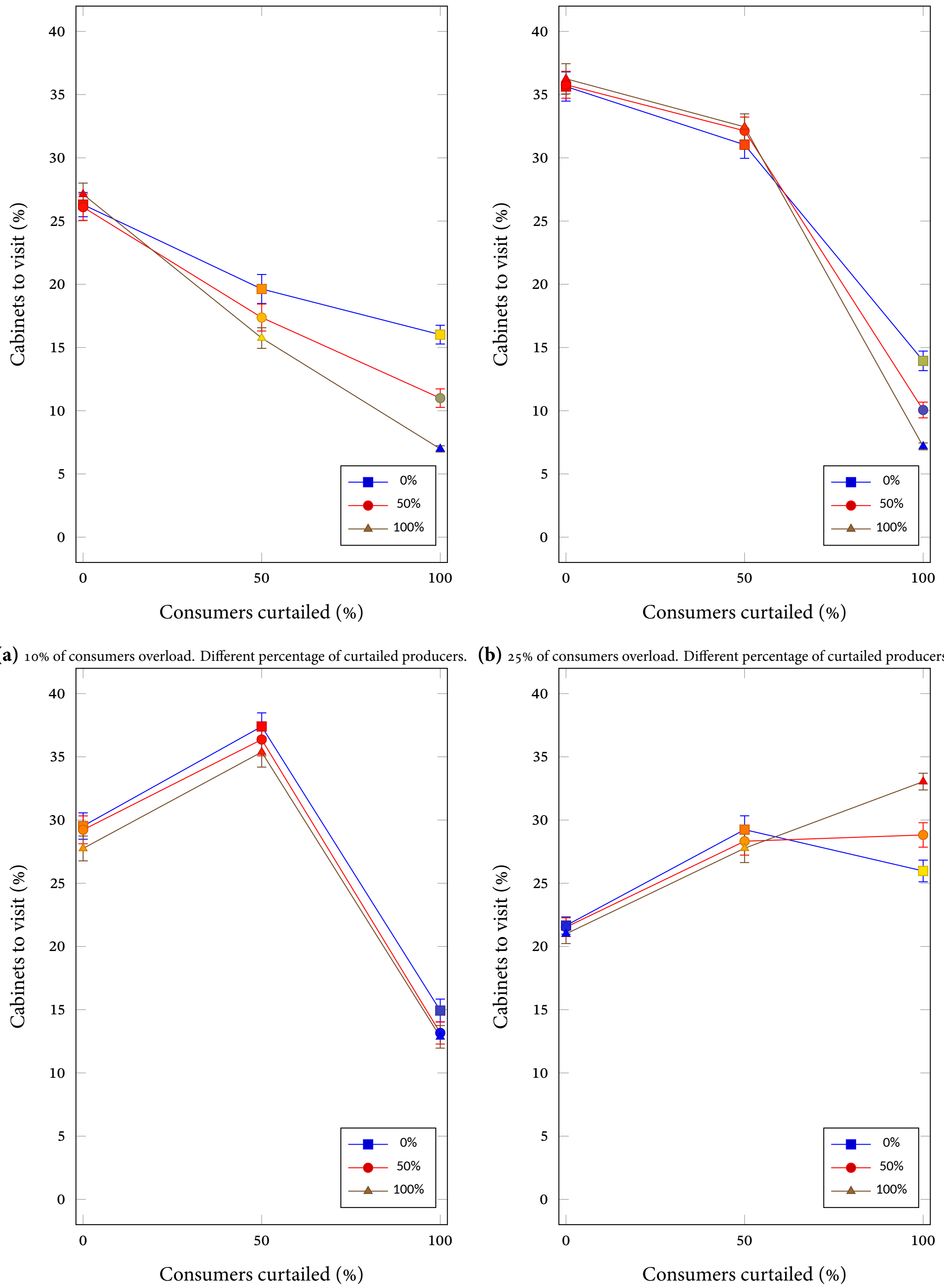


(c) 50% of consumers overload. Different percentage of curtailed producers.



(d) 100% of consumers overload. Different percentage of curtailed producers.

Figure A.7: Cabinets to visit (%) (10% of producers overload)



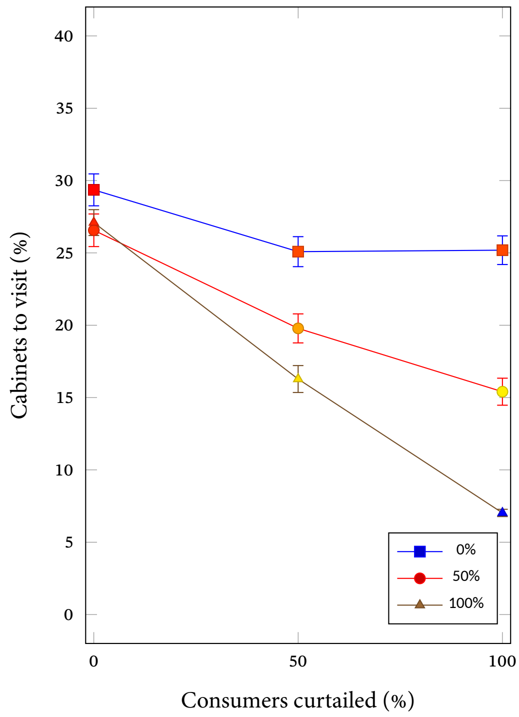
(a) 10% of consumers overload. Different percentage of curtailed producers.

(b) 25% of consumers overload. Different percentage of curtailed producers.

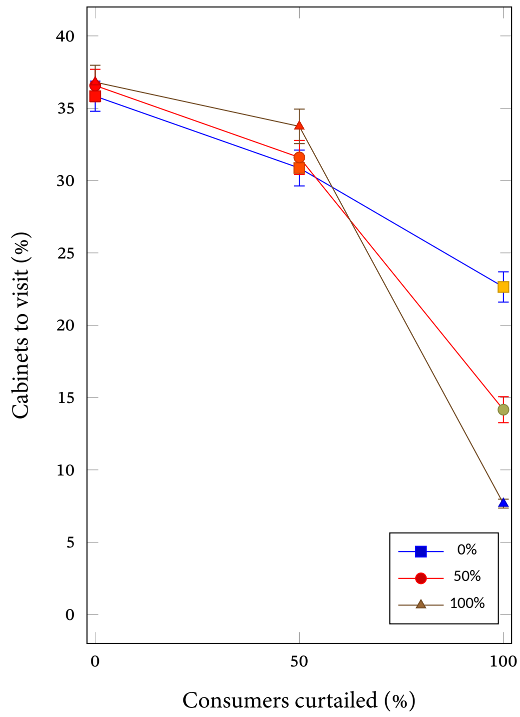
(c) 50% of consumers overload. Different percentage of curtailed producers.

(d) 100% of consumers overload. Different percentage of curtailed producers.

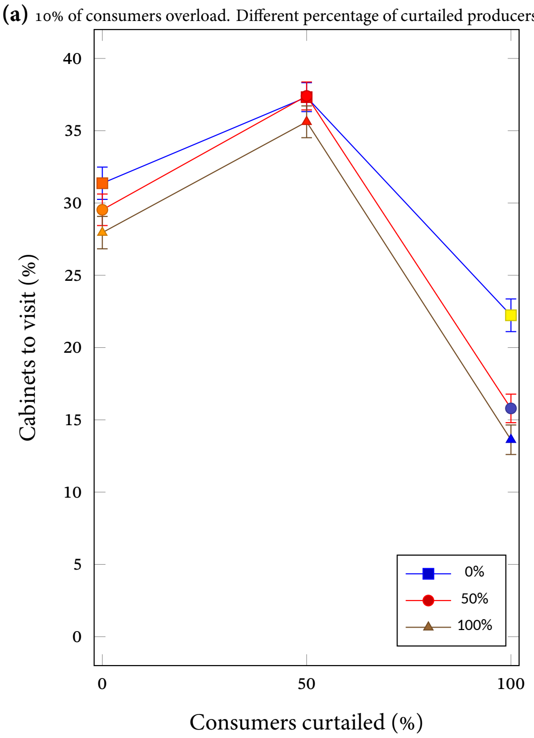
Figure A.8: Cabinets to visit (%) (25% of producers overload)



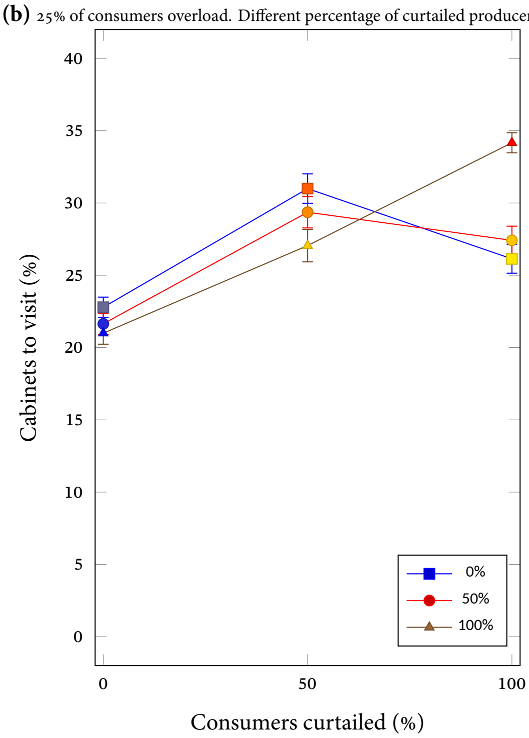
(a) 10% of consumers overload. Different percentage of curtailed producers.



(b) 25% of consumers overload. Different percentage of curtailed producers.

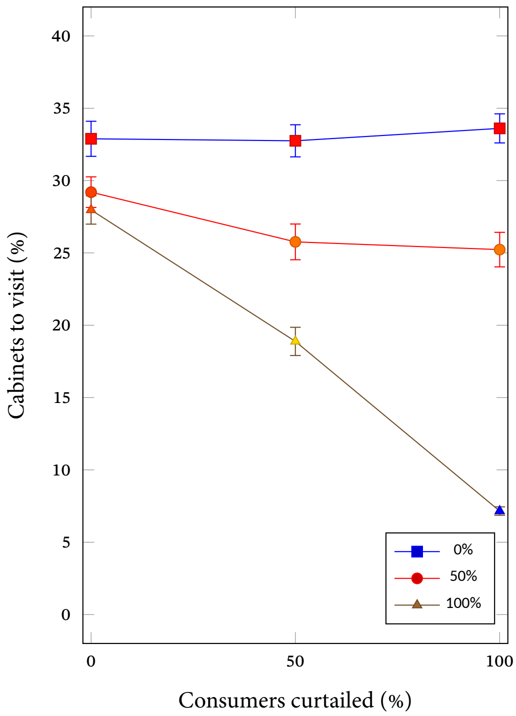


(c) 50% of consumers overload. Different percentage of curtailed producers.

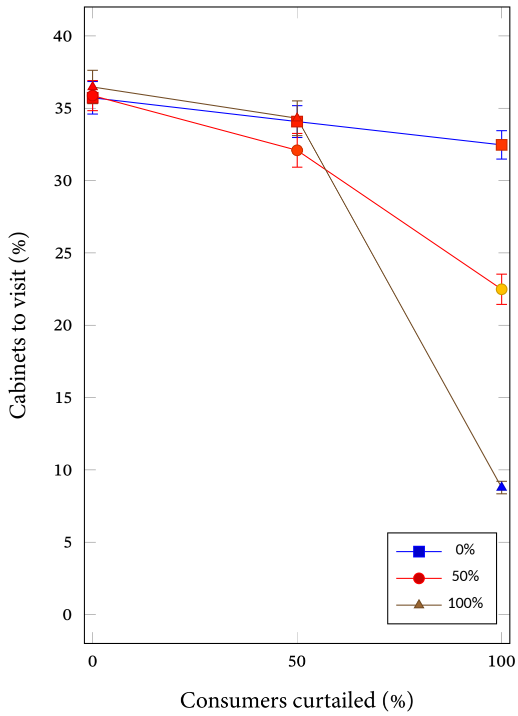


(d) 100% of consumers overload. Different percentage of curtailed producers.

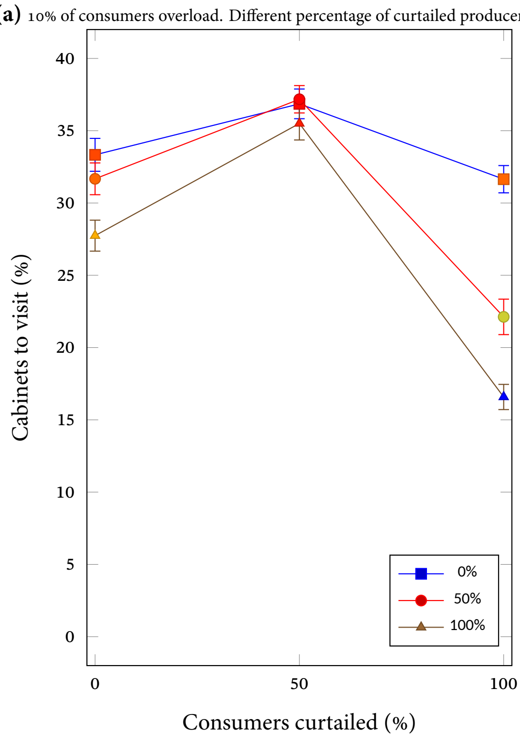
Figure A.9: Cabinets to visit (%) (50% of producers overload)



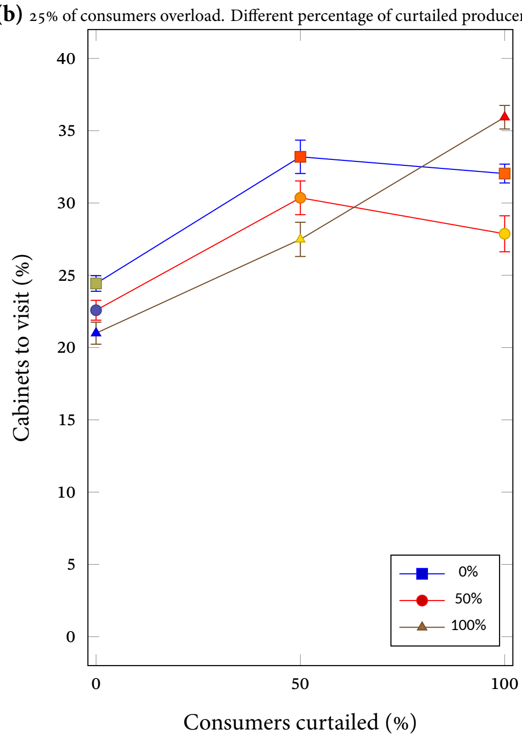
(a) 10% of consumers overload. Different percentage of curtailed producers.



(b) 25% of consumers overload. Different percentage of curtailed producers.



(c) 50% of consumers overload. Different percentage of curtailed producers.

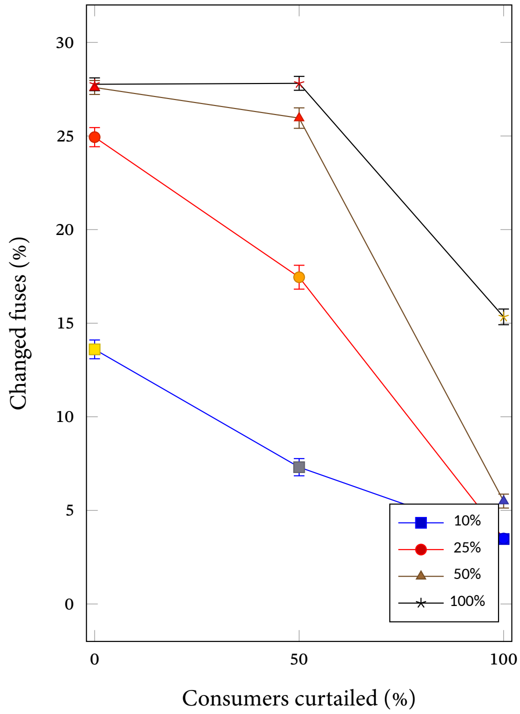


(d) 100% of consumers overload. Different percentage of curtailed producers.

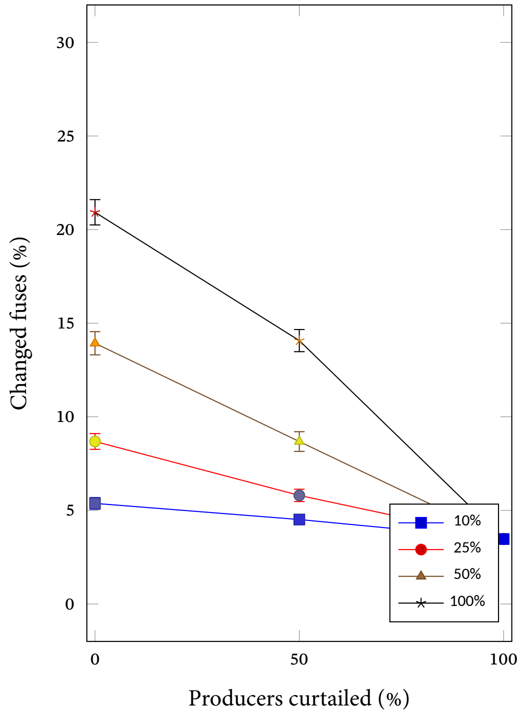
Figure A.10: Cabinets to visit (%) (100% of producers overload)

A.3 Changed fuses

The percentage of the fuses to be changed is presented; the lower, the better.

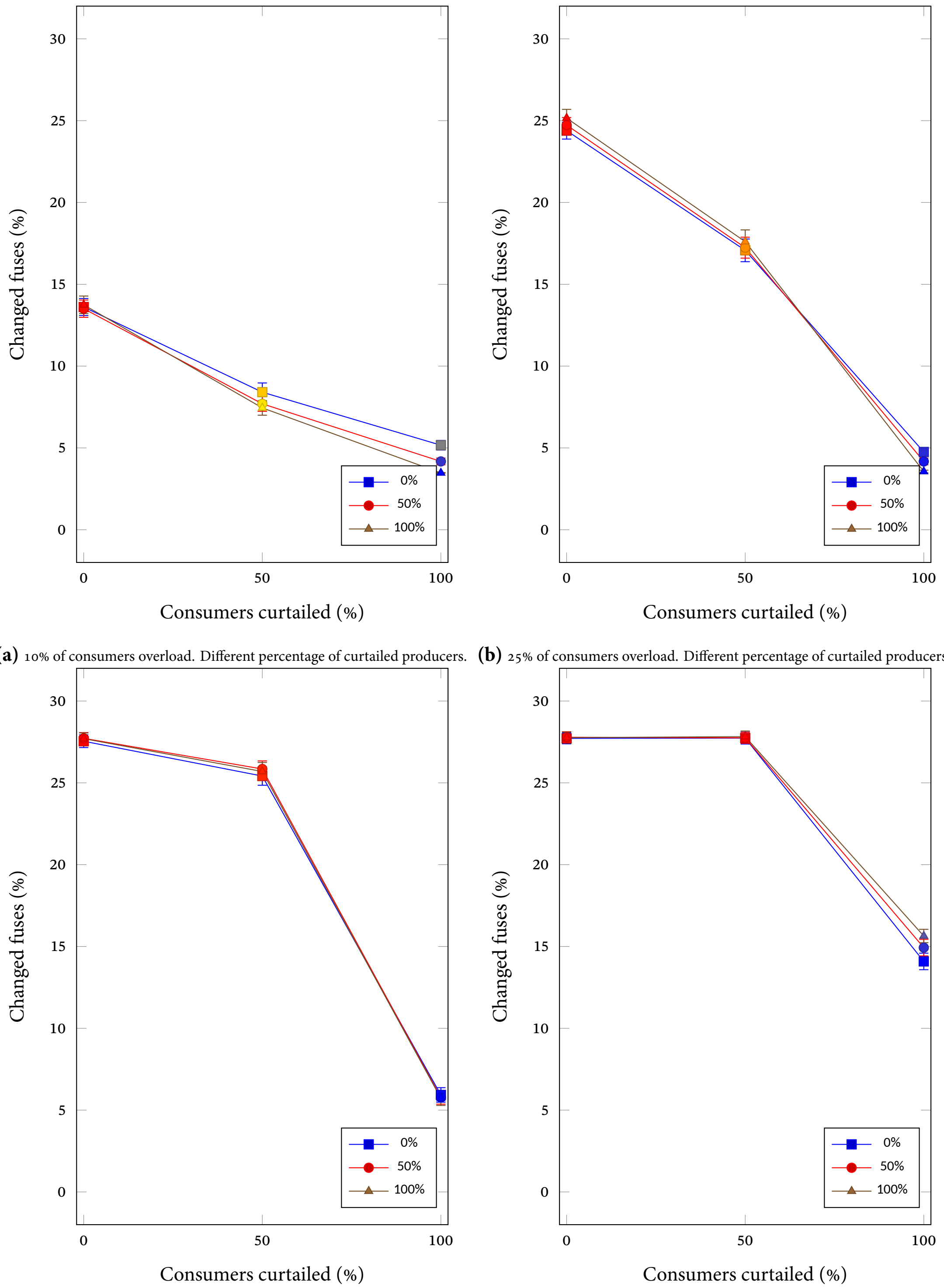


(a) None of the producers overloaded. Different percentage of consumers overload.



(b) None of the consumers overloaded. Different percentage of producers overload.

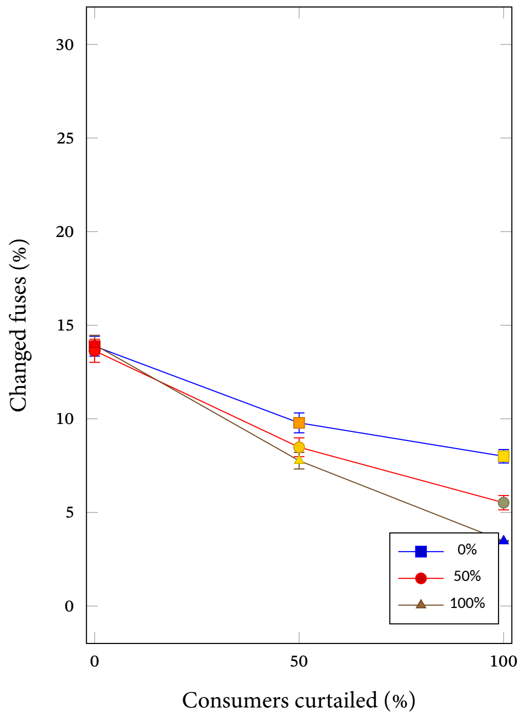
Figure A.11: Changed fuses (%)



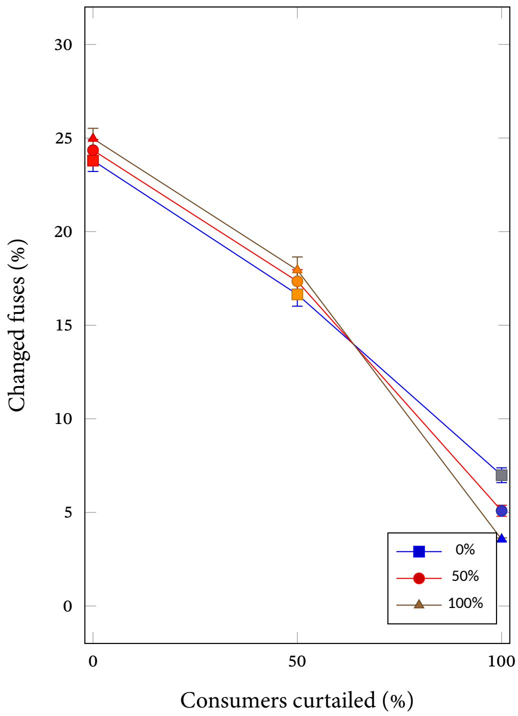
(a) 10% of consumers overload. Different percentage of curtailed producers. **(b)** 25% of consumers overload. Different percentage of curtailed producers.

(c) 50% of consumers overload. Different percentage of curtailed producers. **(d)** 100% of consumers overload. Different percentage of curtailed producers.

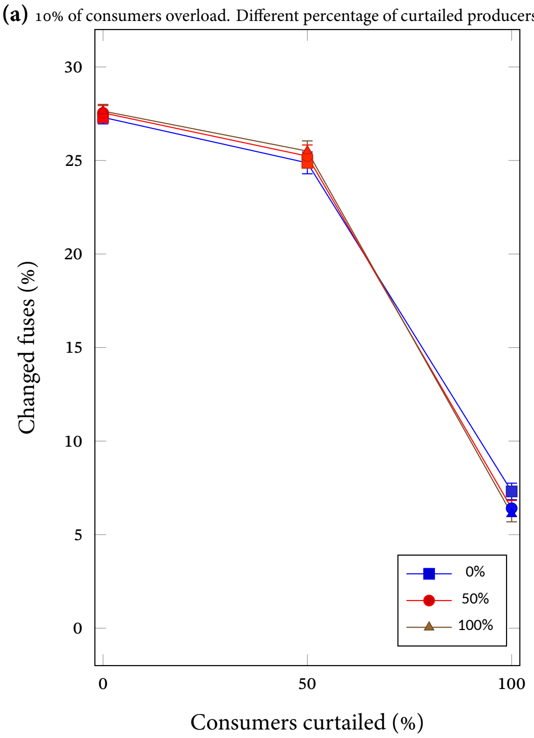
Figure A.12: Changed fuses (%) (10% of producers overload)



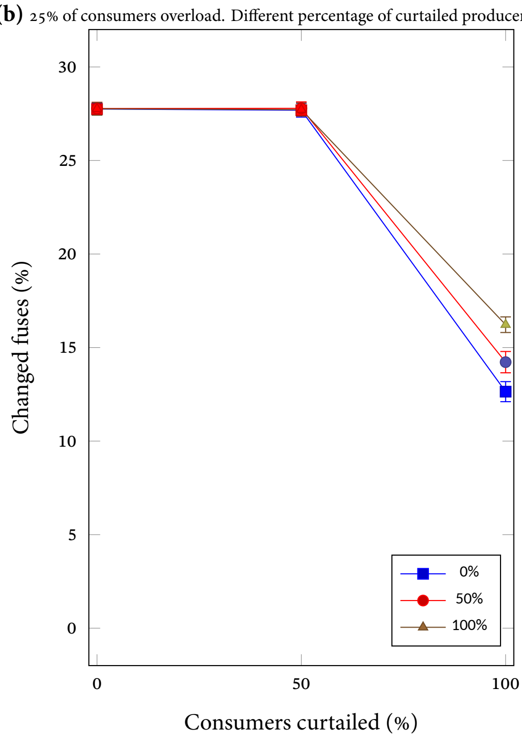
(a) 10% of consumers overload. Different percentage of curtailed producers.



(b) 25% of consumers overload. Different percentage of curtailed producers.

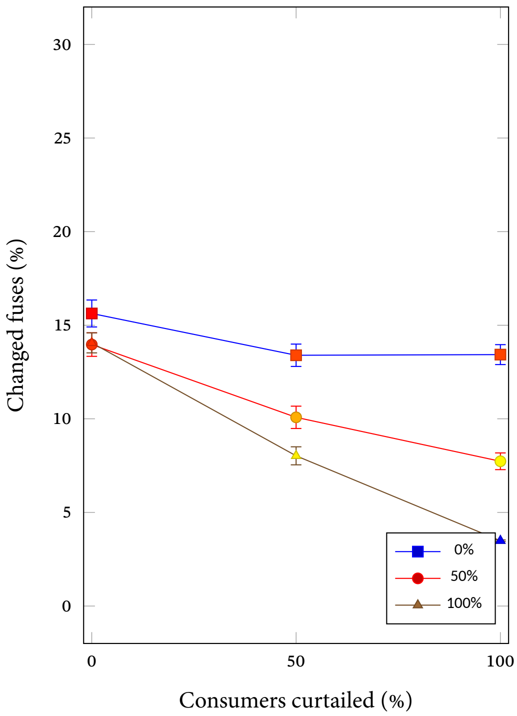


(c) 50% of consumers overload. Different percentage of curtailed producers.

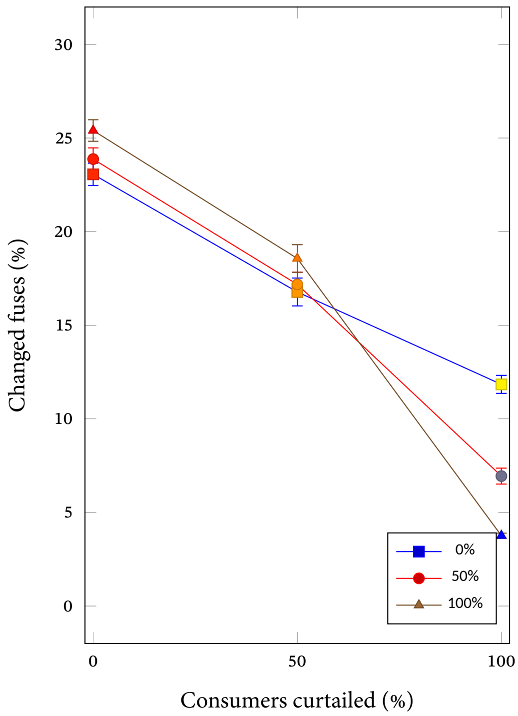


(d) 100% of consumers overload. Different percentage of curtailed producers.

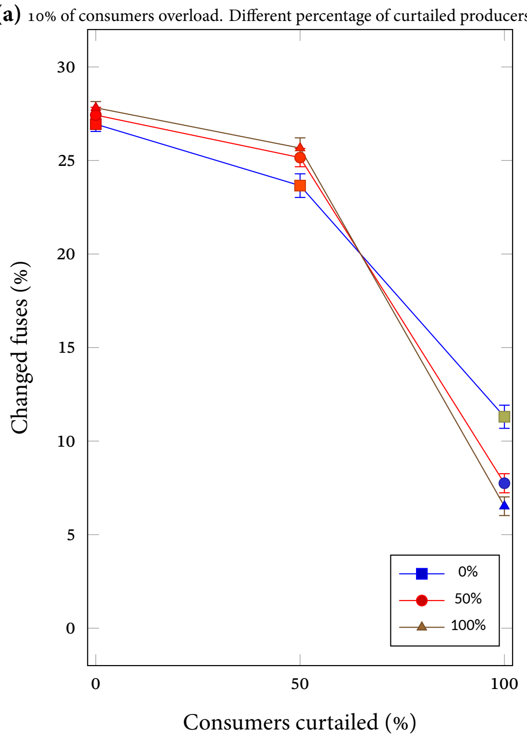
Figure A.13: Changed fuses (%) (25% of producers overload)



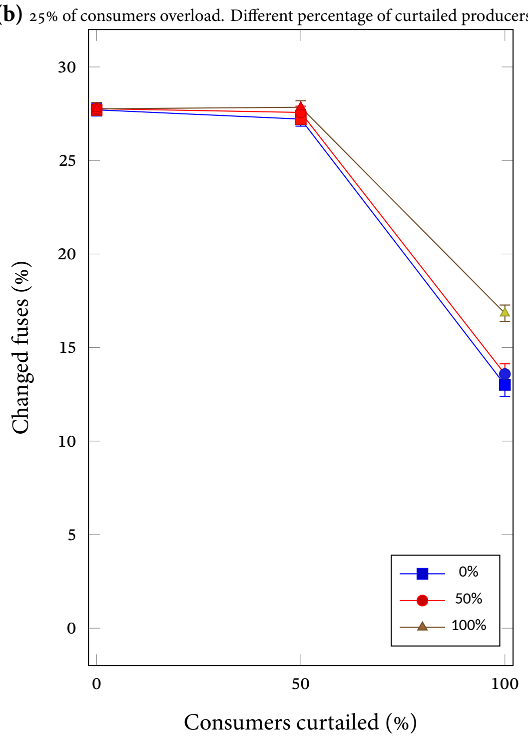
(a) 10% of consumers overload. Different percentage of curtailed producers.



(b) 25% of consumers overload. Different percentage of curtailed producers.

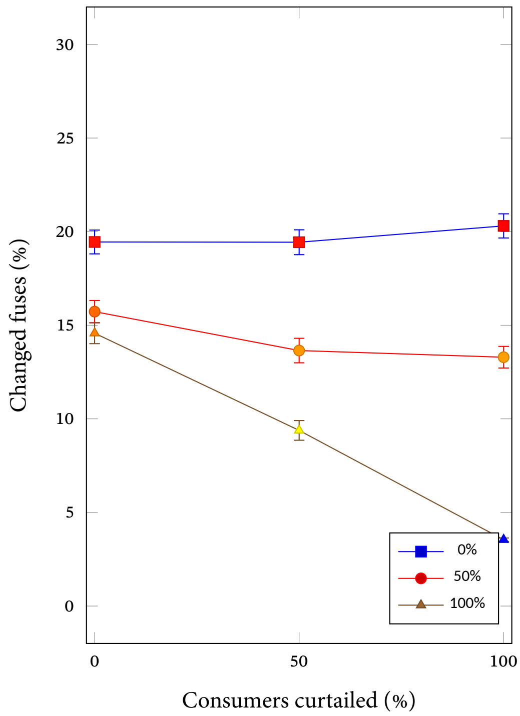


(c) 50% of consumers overload. Different percentage of curtailed producers.

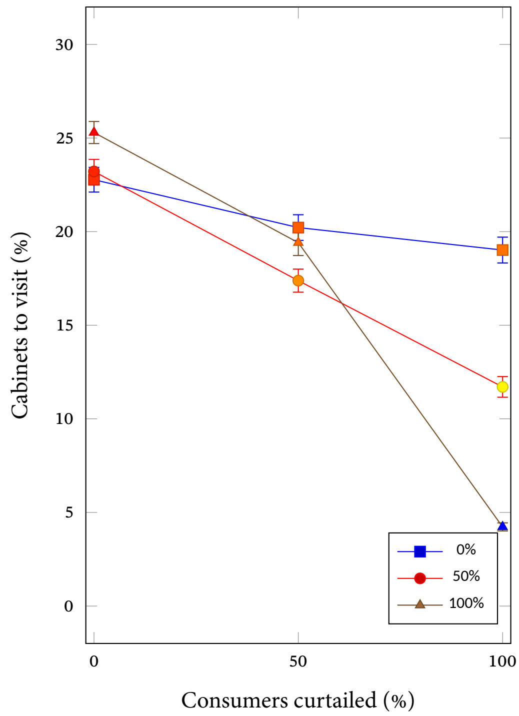


(d) 100% of consumers overload. Different percentage of curtailed producers.

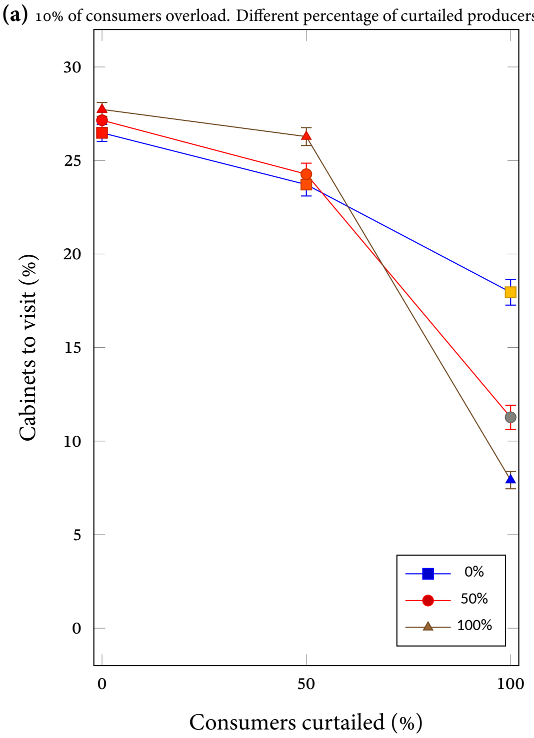
Figure A.14: Changed fuses (%) (50% of producers overload)



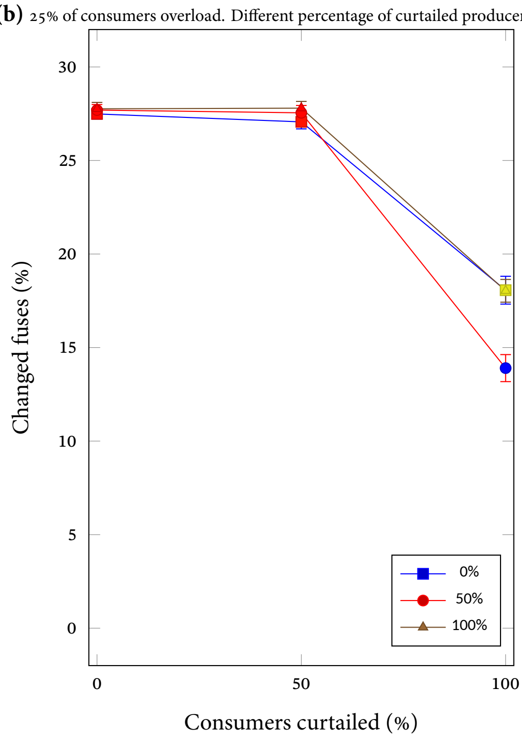
(a) 10% of consumers overload. Different percentage of curtailed producers.



(b) 25% of consumers overload. Different percentage of curtailed producers.



(c) 50% of consumers overload. Different percentage of curtailed producers.



(d) 100% of consumers overload. Different percentage of curtailed producers.

Figure A.15: Changed fuses (%) (100% of producers overload)