UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

EVELISE DE GODOY ANTUNES

ISLANDED MICROGRID OPTIMAL SIZING USING GRADIENT-BASED ALGORITHM

Porto Alegre
2022

EVELISE DE GODOY ANTUNES

ISLANDED MICROGRID OPTIMAL SIZING USING GRADIENT-BASED ALGORITHM

Thesis presented to Programa de Pós-Graduação em Engenharia Elétrica of Universidade Federal do Rio Grande do Sul in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.
Area: Energy

ADVISOR: Prof. Dr. Roberto Chouhy Leborgne

Porto Alegre
2022

EVELISE DE GODOY ANTUNES

ISLANDED MICROGRID OPTIMAL SIZING USING GRADIENT-BASED ALGORITHM

This thesis was considered adequate for obtaining the degree of Master in Electrical Engineering and approved in its final form by the Advisor and the Examination Committee.

Advisor: ____________________________________
Prof. Dr. Roberto Chouhy Leborgne, UFRGS
Doctor by Chalmers University of Technology – Gothenburg, Sweden

Examination Committee:

Prof. Dr. Flávio Antonio Becon Lemos, UFRGS
Doctor by the Federal University of Santa Catarina – Florianópolis, Brazil

Prof. Dr. Mariana Resener, SFU
Doctor by the Federal University of Rio Grande do Sul – Porto Alegre, Brazil

Prof. Dr. Paulo Ricardo da Silva Pereira, UNISINOS
Doctor by the Federal University of Santa Maria – Santa Maria, Brazil

Coordinator of PPGEE: ____________________________
Prof. Dr. Sérgio Luis Haffner

Porto Alegre, July 2022.

ABSTRACT

In isolated sites where there is not a connection with the main electric grid, the use of diesel generating set as electric energy source is considerable, because of its reliability and its climate independence. With the intensification of environmental concern, the use of renewables energy sources has increase. However, these islanded microgrids with high penetration of renewables need to be well sized to allow the load fulfillment with high level of reliability. That is why a suitable optimization algorithm need to be used to obtain the optimal microgrid size for each project. Therefore, this work presents the use of a deterministic gradient-based optimization algorithm to solve the microgrid optimal sizing, which is not often used for this type of problem. The aim of the optimization is to obtain the microgrid design with the lowest net present cost (NPC), that respect a defined maximum shedding fraction (SF). One possible way to obtain the values of NPC and SF for each microgrid configuration and components sizes is to simulate a whole year of its operation, by means of a microgrid simulator. The calculation of the gradients used in the optimization algorithm is done with a Automatic Differentiation package which has a high accuracy and computational performance. The convergence and computation time results are compared with the ones from Particle Swarm Optimization (PSO), a metaheuristic optimization algorithm that is commonly used to solve the optimal microgrid sizing problem.

Keywords: Microgrids, optimal sizing, gradient-based optimization, automatic differentiation (AD), particle swarm optimization (PSO).

RESUMO

Em locais isolados onde não há conexão com a rede elétrica de distribuição, o uso do gerador a diesel como fonte de energia elétrica é considerável, devido à sua confiabilidade e independência climática. Com a intensificação da preocupação ambiental, o uso de fontes de energia renováveis tem aumentado. Entretanto, estas micro redes isoladas com alta penetração de energia renovável precisam ser bem dimensionadas para permitir o atendimento da carga com alto nível de confiabilidade. É por isso que um algoritmo de otimização adequado precisa ser usado para obter o tamanho ideal da microrrede para cada projeto. Portanto, este trabalho apresenta o uso de um algoritmo de otimização determinístico baseado em gradiente para resolver o dimensionamento ótimo da microrrede, que não é frequentemente utilizado para este tipo de problema. O objetivo da otimização é obter o projeto da microrrede com o menor custo líquido atual (NPC), que respeite um valor máximo de não atendimento da carga (SF - do inglês *shedding fraction*). Uma maneira possível de obter os valores de NPC e SF para cada configuração de microrrede e especificação de componentes é simular um ano inteiro de sua operação, por meio de um simulador de microrrede. O cálculo dos gradientes usados no algoritmo de otimização é feito com um pacote de Diferenciação Automática que tem uma alta precisão e desempenho computacional. Os resultados da convergência e do tempo de computação são comparados com os resultados do algoritmo de otimização de Enxame de Partículas (PSO - do inglês *Particle Swarm Optimization*), um algoritmo de otimização meta-heurístico que é comumente utilizado para resolver o problema de dimensionamento ótimo da microrrede.

Palavras-chave: microrredes, dimensionamento ótimo, algoritmos de otimização com gradiente, diferenciação automática (AD), enxame de partícula (PSO).

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AD	Automatic Differentiation
BESS	Battery Storage System
BT	Battery
CHP	Combined Heat and Power
COE	Levelized Cost of Energy
CPU	Central Processing Unit
DER-CAM	Distributed Energy Resources Customer Adoption Model
DPSP	Deficiency of Power Supply Probability
EPE	Empresa de Pesquisa Energética
FA	Firefly Algorithm
GAD	Gradient-based+AD
GB+AD	Gradient-based+AD
GOA	Grasshopper Optimization Algorithm
GS	Diesel generating set
HOMER	Hybrid Optimization of Multiple Energy Resources
ICE	Internal Combustion Engine
IPOPT	Interior Point Optimizer
IQR	Interquartile range
JIT	Just-in-time
LPSP	Loss of Power Supply Probability
MAE	Mean Absolute Error
MG	Microgrid
MILP	Mixed Integer Linear Programming
MRE	Mean Relative Error
MT	Micro-turbine
NLP	Nonlinear Programming

NPC	Net present cost
OM	Operation and maintenance
PPGEE	Programa de Pós-Graduação em Engenharia Elétrica
PSO	Particle Swarm Optimization "
PSO-GWO	Particle Swarm Optimization-Grey Wolf Optimizer"
PV	Photovoltaic system
PVGIS	Photovoltaic Geographical Information System
Q1	First quartile
Q2	Second quartile
Q3	Third quartile
QP	Quadratic problem
RAM	Random Access Memory
RF	Renewable Fraction
RLE	Relative Error
RMSE	Root Mean Square Error
SARAH	Surface Solar Radiation Data Set - Heliosat
SD	Standard Deviation
SF	Shedding fraction
SIN	Sistema Interligado Nacional
SLSQP	Sequential Least Squares Programming
SQP	Sequential Quadratic Programming
SWDS	Seawater Desalination System
WT	Wind turbine

LIST OF SYMBOLS

Sets

$\mathscr{C}$	Set of components $c \in \mathscr{C} = \{PV, WT, BT, GS\}$
$\mathscr{Y}_c$	Set of years when the replacements of the component c happen

Decision variables

P_{PV}^{rtd}	Photovoltaic system rated power [kW]
P_{WT}^{rtd}	Wind turbine rated power [kW]
E_{BT}^{rtd}	Battery rated capacity [kWh]
P_{GS}^{rtd}	Diesel generating set rated power [kW]

Intermediate variables

N_c^{rep}	Number of replacements of component c
ℓ_c	Lifetime of component c [yr.]
ℓ_c^{rem}	Remaining life of component c [yr.]
f_{d_i}	Discount factor f_{d_i} for year i
σ_{f_d}	Summation of f_{d_i} for all project years
$f_{c,i}^{rep}$	Replacement factors for component c in $\mathscr{Y}_c$
$P_{PV}(t)$	Photovoltaic system power output [kW]
$U_{hub}(t)$	Wind speed at the hub height of the wind turbine [m/s]
$P_{WT}(t)$	Wind turbine power output [kW]
ℓ_{BT}^{thrpt}	Battery throughput lifetime [yr.]
Q_{BT}^{thrpt}	Total energy that cycles yearly through the battery [kWh]
$P_{BT}^{discharge}(t)$	Battery discharge power limit [kW]
$P_{BT}^{charge}(t)$	Battery charge power limit [kW]
$P_{BT}^{e,max}(t)$	Battery discharge power limit associated with the energy limit [kW]
$P_{BT}^{e,min}(t)$	Battery charge power limit associated with the energy limit [kW]
$E_{BT}(t)$	Battery energy available at time t [kWh]

$P_{BT}(t)$	Battery power in generator convention [kW]
ℓ_{BT}	Battery minimum lifetime [yr.]
$F_{GS}(t)$	Diesel generating set fuel consumption [l]
$P_{GS}(t)$	Diesel generating set electrical power output [kW]
$F_{GS}^{tot,year}$	Yearly diesel generating set total fuel consumption [l]
$h_{GS}(t)$	Diesel generating set operation hour [h]
$h_{GS}^{tot,year}$	Yearly diesel generating set total operation hours in a year [h]
$h_{GS}^{tot,proj}$	Diesel generating set total operation hours during project lifetime [h]
$h_{GS}^{rlx}(t)$	Diesel generating set relaxed operation hour [h]
$P_{net}(t)$	Microgrid net power [kW]
$P_{load}(t)$	Load demand [kW]
$P_{curt}(t)$	Curtailed renewable power [kW]
$P_{shed}(t)$	Shedding load demand [kW]
P_{shed}^{max}	Maximum shedding load demand [kW]
T_{shed}^{max}	Maximum consecutive duration of shedding load demand [h]
$P_{load}^{net}(t)$	Load net power after nondispatchable renewable production [kW]
$P_{ren}(t)$	Sum of power produced by nondispatchable renewable sources [kW]
E_{serv}	Load energy served in one year [kWh]
E_{serv}^{proj}	Total energy served during the project lifetime [kWh]
$C_{c}^{inv,tot}$	Total present investment cost for component c [\$]
$C_{c}^{OM,tot}$	Unitary present operation and maintenance cost for component c [\$/(kWh or kW or kW·h)]
$C_{GS}^{fuel,tot}$	Total present diesel generating set fuel cost [\$]
$C_{c}^{rep,tot}$	Total present replacement cost for component c [\$]
S_{c}	Proportional unitary salvage value for component c [\$/(kWh or kW)]
S_{c}^{tot}	Total present salvage value for component c [\$]
C_{c}^{tot}	Total present cost for component c [\$]
NPC	Total microgrid net present cost [\$]
COE	Levelized cost of energy [\$]
RF	Microgrid renewable fraction [$\%$]
SF	Microgrid shedding fraction [$\%$]

Parameters

T	Number of time steps
Δt	Time step [h]
d	Project discount rate
ℓ_{proj}	Project lifetime [yr.]
ℓ_c	Lifetime of component $c \in \{PV, WT, GS\}$ [yr.]
f_{PV}	Photovoltaic system derating factor
$I_T(t)$	Global solar radiation (beam plus diffuse) incident on the surface of the PV array [kW/m^2]
I_S	Standard amount of radiation used to rate de capacity of the PV array, whose value is 1 [kW/m^2]
$P_{PV}^{1k}(t)$	Power output of a 1 kW PV panel including temperature effects and system loss [pu]
$U_{anem}(t)$	Wind speed measured at the anemometer height [m/s]
z_{hub}	Hub height of the wind turbine [m]
z_{anem}	Anemometer height [m]
z_0	Surface roughness length [m]
α_{WT}	Power law exponent
U_{cut-in}	Wind turbine cut-in wind speed [m/s]
$U_{cut-out}$	Wind turbine cut-out wind speed [m/s]
U_{rtd}	Wind turbine rated wind speed [m/s]
ℓ_{BT}^{years}	Battery lifetime [yr.]
N_{cycles}	Maximum number of battery cycles
E_{BT}^{max}	Battery maximum available energy [kWh]
E_{BT}^{min}	Battery minimum available energy [kWh]
P_{BT}^{max}	Battery discharge power limit associated with the rate limit [kW]
P_{BT}^{min}	Battery charge power limit associated with the rate limit [kW]
α_{BT}	Battery linear loss factor
F_0	Diesel generating set fuel curve intercept coefficient [$l/h/$kW]
F_1	Diesel generating set fuel curve slope [$l/h/$kW]
ε	Diesel generating set relaxation factor
C_c^{inv}	Unitary present investment cost for component c [\$/(kWh or kW)]
C_c^{OM}	Unitary present operation and maintenance cost for component c [\$/(kWh or kW)]
C_{GS}^{fuel}	Unitary diesel generating set fuel cost [\$/l]
C_c^{rep}	Unitary present replacement cost for component c [\$/(kWh or kW)]

CONTENTS

1 INTRODUCTION

Power systems from all over the world are experiencing changes in their configurations. This is mainly caused by the upgrowth climate change awareness by power system actors, including the population. The Brazilian electricity mix is predominantly composed by renewable sources, with a participation of 84.8% (EPE, 2021). This means that the majority load consumption, the part connected to the brazilian National Grid System (SIN, from portuguese Sistema Interligado Nacional), is supplied by renewables sources. Moreover, with the Norm 482/2012 (from portuguese Resolução Normativa 482/2012), followed by its revision Norm 687/15 (from portuguese Resolução Normativa 687/15), which facilitates the installation and the compensation of the residual energy produce by prosumers, the number of distributed photovoltaic systems increased, representing 92.7% of the micro and mini generation (EPE, 2021).

Nevertheless the increase of centralized or distributed renewable power plants, more than 90% of Brazilian islanded systems' electricity is generated by diesel generating sets (gensets or GS, for short) (EPE, 2022), a very pollutant energy source. Therefore, there is a visible divergence between the composition of the Brazilian electricity mix and the Brazilian isolated systems mix. Natural gas and renewable plants were retained in recent auctions, thus, a reduction to 61% of participation is expected in 2023. However, greenhouse gases emissions from diesel gensets will still represent almost 80% of Brazilian islanded systems total emissions (EPE, 2022), indicating that a more expressive reduction is necessary to reduce its environmental impacts.

This is not only a reality of Brazilian islanded systems scenary. Due to the high costs, high price variation and high pollutant emissions of diesel, microgrids with high penetration of renewable energy sources have shown to be a viable option for the electric power supply in remote regions. Suman; Guerrero; Roy (2021) examined the optimal sizing of hybrid islanded microgrids for rural areas, showing that it is possible to have a high penetration of renewable energy in this kind of network. Another application of a islanded microgrid was presented in (Zhao *et al.*, 2014), where a optimal sizing was performed and used as a base to implement a real microgrid in Dongfushan Island in China.

For microgrids with high renewable penetration work properly, i.e., meeting the demand with high reliability, it is necessary to perform their planning in a thorough and detailed way. According to (Gamarra; Guerrero, 2015) the planning steps include selection of energy generation and storage technologies, their sizing, allocation in the microgrid, energy resource operation planning, sensitivity analysis, and pricing strategies.

This work focuses in the sizing and energy resource operation planning steps, which are generally formulated as optimization problems to obtain the best results. The optimization algorithms commonly used to solve these problems are the metaheuristic ones, because of the nonlinearities and discontinuities of the problems. In some studies, a linearization of the models are performed to use exact optimization methods. Each of them have advantages and disadvantages, which will be clarified in Chapter 2.

In this work the steps of sizing and operation will be attached in a unique optimization problem. The algorithm chosen to perform the optimization is a gradient-based one, due to its fast convergence characteristic. This approach is not usually applied to islanded microgrid sizing, because it needs a continuous and differentiable mathematical formulation of the problem to converge. However, using the Automatic Differentiation (AD) technique its possible to obtain the gradients even if the models are not perfectly smooth. Therefore, a convergence and timing analysis are performed to assess if the compromise between these two is suitable to use this optimization approach to obtain optimal microgrid components sizes.

1.1 Objectives

The main objective of this work is to develop and implement a methodology to apply a gradient-based optimization algorithm to the islanded microgrid optimal sizing problem. That is possible with the development of a microgrid simulator (MG Simulator) and the use of an Automatic Differentiation (AD) package. This main objective can be decomposed into the following ones:

- develop a microgrid simulator, capable of receive the climate and technical microgrid data as input, and give as output the power production and balance at each time instant, the yearly or lifetime energetic and economic indicators;

- create a framework to interface the MG Simulator, a gradient-based optimization algorithm, and a Automatic Differentiation package;

- analyze the convergence and computation time of the proposed methodology.

1.2 Outline

This work is organized in 5 chapters. The Chapter 1 presents the Introduction of the subject studied, addressing the motivation and objectives of this research.

Chapter 2 presents the theoretical foundation of the concepts used in the development of the methodology. The base of Automatic Differentiation, the optimization algorithms used in this work, the energetic modeling of the microgrid components and its form of dispatch, and the indicators used are explained. In this chapter, the state of the art is also addressed and discussed.

The proposed methodology to design an optimal islanded microgrid is described in Chapter 3. The development of the MG simulator is fully explained. The coupling method between the MG simulator, AD package and gradient-based optimization algorithm is established, as well as the coupling between the MG simulator and the Particle Swarm Optimization algorithm. It will also be explained how the convergence is analyzed to compare the two algorithms.

In Chapter 4, the methodology is applied to four case studies and all the parameters values and tools used are listed. The results for each microgrid configuration are also presented and discussed.

Chapter 5 summaries the conclusions of the presented work and suggests advances that could be done to this research.

2 BACKGROUND

This chapter presents the concepts from the literature that were used in the development of this work.

2.1 Automatic differentiation

Automatic Differentiation (AD), also known as algorithmic differentiation, is a form to evaluate derivatives. It is primarily based on the chain rule (Baydin *et al.*, 2018; Gebremedhin; Walther, 2020), i.e., a formula to evaluate the derivatives of composed functions as in (1). The methods can also be extended to evaluate higher orders of derivatives, gradients and jacobian matrices.

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx} \tag{1}$$

where

$$y = f(u)$$
$$u = g(x)$$

In theory, AD should evaluate the derivatives of any function that can be written as a program code (Gilbert; Le Vey; Masse, 1991). This include functions that can't be represented by explicit mathematical formulas, composed of programming control flow functions, e.g., if-then-else,while-do,for-do, that calls another subroutines, that have iterative and recursive process, and any other structure of computer programming (Gilbert; Le Vey; Masse, 1991). The capacity of AD to works in all of this cases depends of the AD program implementation.

The working principle of AD programs is to transform the function code, decomposing it as a combination of elementary operations (binary arithmetic operations, unary sign switch, and functions as exponential, logarithm, and trigonometric functions) for which the derivatives are known (Baydin *et al.*, 2018). For functions without programming control flow functions, it is possible to represent the combination of the elementary operations as a code list and a computational graph (Gilbert; Le Vey; Masse, 1991).

The computational graph is a direct acyclic graph, where the vertices represent the input, intermediate and output variables and the edges describe the dependency relationship between the vertices (Gebremedhin; Walther, 2020). The elementary operations can be annotated in the vertices and in the edges the local partial derivatives, i.e., partial derivative of the target vertex with respect to the source vertex, can be showed as weights (Gebremedhin; Walther, 2020). The interested readers can found more concepts of graphs in (Godsil; Royle, 2001).

A function $f : \Re^n \to \Re^m$ can be decomposed into $n + m + l$ values v_i, where l is the number of elementary operations needed. The elementary operations are represented by an elemental function φ. Thus, each value $v_i \in \Re$, $i > 0$, is obtained with

$$v_i = \varphi_i(v_j)_{j \prec i}, \; \varphi_i : \Re^{n_i} \to \Re, \tag{2}$$

where n_i is the numbers of arguments v_j, $j < i$, required to evaluate φ_i (Gebremedhin; Walther, 2020). Therefore, a general evaluation procedure for a function $f : \Re^n \to \Re^m$, using the notation of (Griewank; Walther, 2008), is

1. $v_{i-n} = x_i, \; i = 1, \ldots, n$, i.e., the attribution of values for the n input variables;
2. $v_i = \varphi_i(v_j)_{j \prec i}, \; i = 1, \ldots, l$, i.e., the calculation of the l intermediate variables;
3. $y_{m-i} = v_{l-i}, \; i = m - 1, \ldots, 0$, i.e., the attribution of values for the m output variables.

There are two basic modes that an AD program can use to calculate the derivatives: forward mode and backward mode. Some AD programs implement a hybrid of these two to take advantage of each one. The function

$$y = f(x_1, x_2) = \ln(x_1) + x_1 \cdot x_2 - \sin(x_2) \tag{3}$$

is used as an example to explain the basic modes (Baydin *et al.*, 2018). Figure 1 shows the computational graph of (3) with the elemental functions annotated in the vertices, and the decomposition as a code list is presented in Table 1 with the function value calculation for the point $\boldsymbol{x} = [2, 5]^T$.

2.1.1 Forward mode

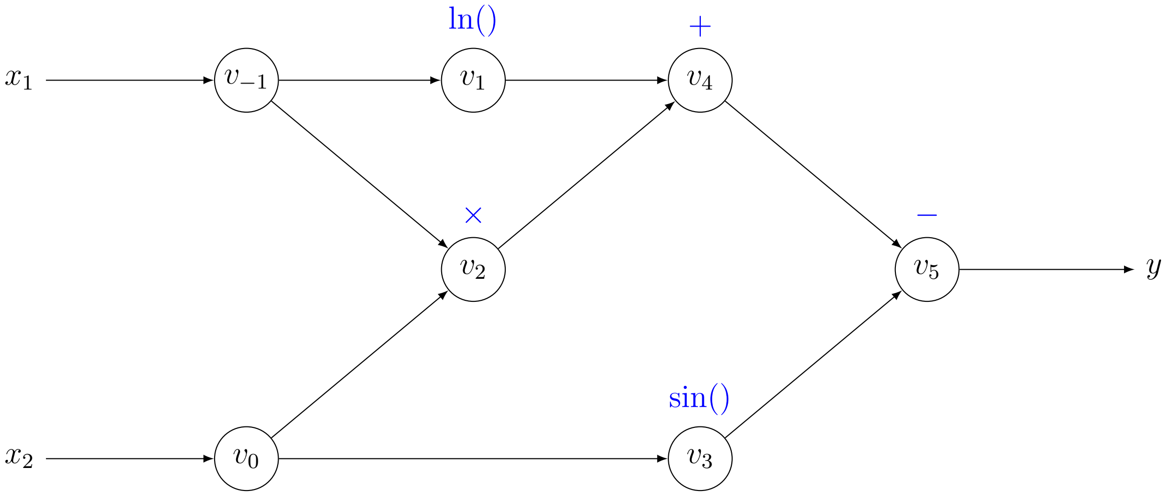
The forward mode consists of computing the partial derivatives $\dot{v}_i$ at the same time as the function value v_i. Using the definition of elemental functions (2) and the chain rule (1), we have

$$\dot{\varphi}_i : \Re^{2n_i} \to \Re, \; \dot{v}_i = \nabla\varphi_i(u_i)\dot{v}_i \equiv \dot{\varphi}_i(u_i, \dot{u}_i),$$

that can also be written as

$$\dot{v}_i = \sum_{j \prec i} \frac{\partial \varphi_i(u_i)}{\partial v_j} \cdot \dot{v}_j, \tag{4}$$

Figure 1 – Computational graph of Eq. (3).



Source: adapted from (Baydin *et al.*, 2018).

Table 1 – Step-by-step calculation of Eq. (3) as a code list.

Forward Primal Trace	
General	$\boldsymbol{x} = [2, 5]^T$
$v_{-1} = x_1$	2
$v_0 = x_2$	5
$v_1 = \varphi_1(v_{-1}) = \ln(v_{-1})$	$\ln(2)$
$v_2 = \varphi_2(v_{-1}, v_0) = v_{-1} \cdot v_0$	$2 \cdot 5$
$v_3 = \varphi_3(v_0) = \sin(v_0)$	$\sin(5)$
$v_4 = \varphi_4(v_1, v_2) = v_1 + v_2$	$0.693 + 10$
$v_5 = \varphi_5(v_3, v_4) = v_4 - v_3$	$10.693 + 0.959$
$y = v_5$	11.652

Source: adapted from (Baydin *et al.*, 2018).

where $u_i = (v_j)_{j \prec i}$ and $\dot{u}_i = (\dot{v}_j)_{j \prec i}$ (Gebremedhin; Walther, 2020).

Considering a function $f : \Re^n \to \Re$, for each i iteration of the forward process, the input variable, $\boldsymbol{x} = [x_1, \ldots, x_n]^T$, is the i-th vector of the canonical base of $\Re^n$, i.e., $\boldsymbol{x} = \boldsymbol{e}_i = [0, \ldots, 0, 1, 0, \ldots, 0]^T$ and the partial derivative $\frac{\partial f}{\partial x_i}$ is obtained. Thus, the gradient $\nabla f = [\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}]^T$ is obtained with n iterations of the forward process (Gilbert; Le Vey; Masse, 1991; Baydin *et al.*, 2018).

This process can be extended for a function $f : \Re^n \to \Re^m$, for which a Jacobian matrix is computed. Because of the forward mode characteristic, all partial derivatives with respect to a same input variable are evaluated in the same iteration. In that manner, the number of iterations continues to be n, because one column of the matrix

$$J_f(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial y_1}{\partial x_1} & \dfrac{\partial y_1}{\partial x_2} & \cdots & \dfrac{\partial y_1}{\partial x_n} \\ \dfrac{\partial y_2}{\partial x_1} & \dfrac{\partial y_2}{\partial x_2} & \cdots & \dfrac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial y_m}{\partial x_1} & \dfrac{\partial y_m}{\partial x_2} & \cdots & \dfrac{\partial y_m}{\partial x_n} \end{bmatrix} \tag{5}$$

is obtained for each iteration. This leads to a general preference to use the forward mode for $f : \Re^n \to \Re^m$ where $n \gg m$ (Baydin *et al.*, 2018). Some implementations of forward mode are more efficient for functions where $n > m$ but n is not much greater than m, e.g., ForwardDiff.jl compared to ReverseDiff.jl (Revels; Pearson, 2016).

For the example function (3), applying the equation (4) to the elemental functions (first column of Table 1) results in the partial derivatives presented in Table 2. Here, each row of Table 2 is executed at the same time. The partial derivatives $\dot{v}_i$ are also annotated as weights in the computational graph presented in Fig. 2, where the direction of the calculation is also shown.

Table 2 – Step-by-step calculation of Eq. (3) gradient with forward automatic differentiation.

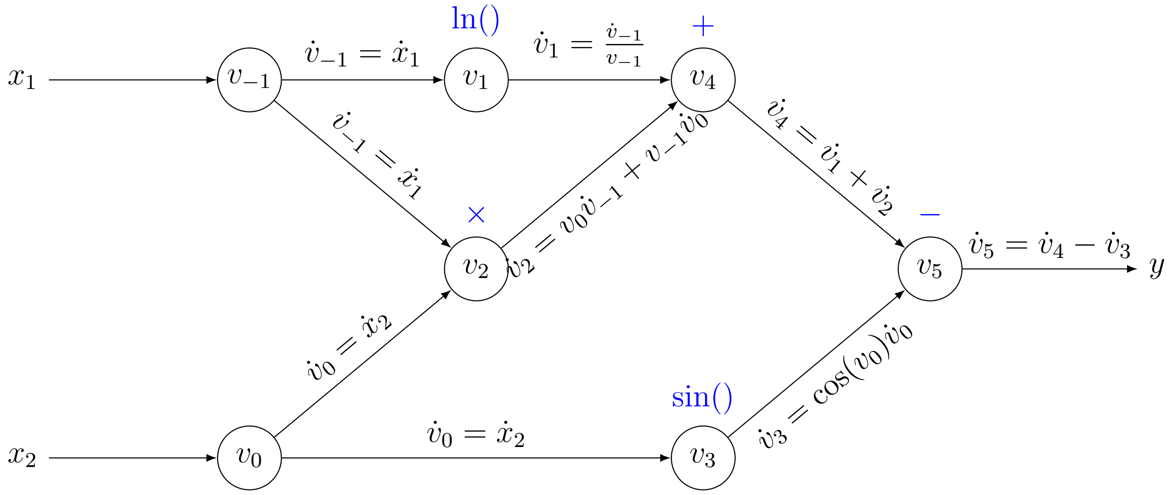
Forward Tangent (Derivative) Trace		
General	$\dot{\boldsymbol{x}} = [1, 0]^T$	$\dot{\boldsymbol{x}} = [0, 1]^T$
$\dot{v}_{-1} = \dot{x}_1$	1	0
$\dot{v}_0 = \dot{x}_2$	0	1
$\dot{v}_1 = \frac{\partial \varphi_1}{\partial v_{-1}} \cdot \dot{v}_{-1} = \frac{\dot{v}_{-1}}{v_{-1}}$	$\frac{1}{2}$	$\frac{0}{2}$
$\dot{v}_2 = \frac{\partial \varphi_2}{\partial v_{-1}} \cdot \dot{v}_{-1} + \frac{\partial \varphi_2}{\partial v_0} \cdot \dot{v}_0 = v_0 \cdot \dot{v}_{-1} + v_{-1} \cdot \dot{v}_0$	$5 \cdot 1 + 2 \cdot 0$	$5 \cdot 0 + 2 \cdot 1$
$\dot{v}_3 = \frac{\partial \varphi_3}{\partial v_0} \cdot \dot{v}_0 = \cos(v_0) \cdot \dot{v}_0$	$\cos(5) \cdot 0$	$\cos(5) \cdot 1$
$\dot{v}_4 = \frac{\partial \varphi_4}{\partial v_1} \cdot \dot{v}_1 + \frac{\partial \varphi_4}{\partial v_2} \cdot \dot{v}_2 = \dot{v}_1 + \dot{v}_2$	$0.5 + 5$	$0 + 2$
$\dot{v}_5 = \frac{\partial \varphi_5}{\partial v_4} \cdot \dot{v}_4 - \frac{\partial \varphi_5}{\partial v_3} \cdot \dot{v}_3 = \dot{v}_4 - \dot{v}_3$	$5.5 - 0$	$2 - 0.284$
$\dot{y} = \dot{v}_5$	$\frac{\partial y}{\partial x_1} = 5.5$	$\frac{\partial y}{\partial x_2} = 1.716$

Source: adapted from (Baydin *et al.*, 2018).

2.1.2 Reverse mode

The reverse mode is a two-phase process. In the first phase the function value is evaluated, through the evaluation of the intermediate variables, as in the forward mode (Baydin *et al.*, 2018). The difference is that the partial derivatives can be only evaluated after the complete execution of this first phase. That happens because the partial derivatives

Figure 2 – Computational graph of Eq. (3) with the partial derivatives from forward AD annotated.


x_1
x_2
v_{-1}
v_0
v_1
v_2
v_3
v_4
v_5
y
ln()
$\times$
sin()
$+$
$-$
$\dot{v}_{-1} = \dot{x}_1$
$\dot{v}_{-1} = \dot{x}_1$
$\dot{v}_1 = \frac{\dot{v}_{-1}}{v_{-1}}$
$\dot{v}_0 = \dot{x}_2$
$\dot{v}_0 = \dot{x}_2$
$\dot{v}_2 = v_0\dot{v}_{-1} + v_{-1}\dot{v}_0$
$\dot{v}_3 = \cos(v_0)\dot{v}_0$
$\dot{v}_4 = \dot{v}_1 + \dot{v}_2$
$\dot{v}_5 = \dot{v}_4 - \dot{v}_3$

Source: adapted from (Baydin *et al.*, 2018).

are computed backwards, using adjoint functions $\bar{v}_i$, from the output variables y_j to the input variables x_i (Gebremedhin; Walther, 2020).

The adjoint functions can be seen as a sensitivity of a considered output y_j with respect to changes in v_i, i.e., $\bar{v}_i = \frac{\partial y_j}{\partial v_i}$ (Baydin *et al.*, 2018). Using the concept of elemental functions, the formula of adjoint functions can be expressed as

$$\bar{u}_i {+}{=}\ \bar{v}_i \cdot \nabla\varphi_i(u_i) \tag{6}$$

for each elemental function $\varphi_i : \Re^{n_i} \rightarrow \Re$, where $\bar{u}_i \equiv (\bar{v}_j)_{j \prec i} \in \Re^{n_i}$ and $+=$ represents a updating addition operator that assigns the result $\bar{u}_i$ of the operation $\bar{v}_i \cdot \nabla\varphi_i(u_i)$ back into its left operand, e.g, for a second operation of $\bar{u}_i$, the Equation (6) will be $\bar{u}_i = \bar{u}_i + \bar{v}_i \cdot \nabla\varphi_i(u_i)$ (Gebremedhin; Walther, 2020).

As opposed to forward mode, the reverse mode can evaluate the gradient ∇f for a function $f : \Re^n \rightarrow \Re$ with only one iteration of the reverse process. The process is initialized with $\bar{v}_l = \bar{y} = \frac{\partial y}{\partial y} = 1$ and the partial derivatives $\bar{x}_1 = \frac{\partial y}{\partial x_1}, \bar{x}_2 = \frac{\partial y}{\partial x_2}, \ldots, \bar{x}_n = \frac{\partial y}{\partial x_n}$, are obtained at the end of the iteration. For a function $f : \Re^n \rightarrow \Re^m$, each iteration will calculate one row of the Jacobian matrix (5), requiring m iterations to obtain the full Jacobian. Thus, the reverse mode of AD is usually preferred when $m \ll n$. This advantage comes with a increased storage cost, proportional with the l intermediate variables, because all the values v_i need to be stored for the subsequent use. (Baydin *et al.*, 2018)

Using the reverse mode of AD to solve the gradient for the example function (3) results in the steps presented in Table 3. Following the two-phase process, first all the rows of the *forward primal trace* are executed. In the second phase, the rows of *reverse adjoint (derivative) trace* are executed from the last one to the first one. In this example it is possible to observe the evaluation of the full gradient $\nabla f(x_1, x_2)$ in only one iteration of the reverse mode. Figure 3 presents the computational graph with the adjoints annotated as weights in the edges.

Table 3 – Step-by-step calculation of Eq. (3) gradient with reverse automatic differentiation.

Reverse Adjoint (Derivative) Trace	
General	$\bar{y} = 1$
$\bar{x}_1 = \bar{v}_{-1}$	5.5
$\bar{x}_2 = \bar{v}_0$	1.716
$\bar{v}^b_{-1} = \bar{v}^a_{-1} + \bar{v}_1 \cdot \frac{\partial \varphi_1}{\partial v_{-1}} = \bar{v}^a_{-1} + \frac{\bar{v}_1}{v_{-1}}$	5.5
$\bar{v}^b_0 = \bar{v}^a_0 + \bar{v}_2 \cdot \frac{\partial \varphi_2}{\partial v_0} = \bar{v}^a_0 + \bar{v}_2 \cdot v_{-1}$	1.716
$\bar{v}^a_{-1} = \bar{v}_2 \cdot \frac{\partial \varphi_2}{\partial v_{-1}} = \bar{v}_2 \cdot v_0$	5
$\bar{v}^a_0 = \bar{v}_3 \cdot \frac{\partial \varphi_3}{\partial v_0} = \bar{v}_3 \cdot \cos(v_0)$	-0.284
$\bar{v}_2 = \bar{v}_4 \cdot \frac{\partial \varphi_4}{\partial v_2} = \bar{v}_4 \cdot 1$	1
$\bar{v}_1 = \bar{v}_4 \cdot \frac{\partial \varphi_4}{\partial v_1} = \bar{v}_4 \cdot 1$	1
$\bar{v}_3 = \bar{v}_5 \cdot \frac{\partial \varphi_5}{\partial v_3} = \bar{v}_5 \cdot (-1)$	-1
$\bar{v}_4 = \bar{v}_5 \cdot \frac{\partial \varphi_5}{\partial v_4} = \bar{v}_5 \cdot 1$	1
$\bar{v}_5 = \bar{y}$	1

Source: adapted from (Baydin *et al.*, 2018).

2.2 Optimization algorithms

Optimization is a mathematical study area whose aim is to obtain a minimal or maximal value for a model. The optimization problems can be classified as discrete, continuous, linear, nonlinear, unconstrained, constrained, deterministic and stochastic (Nocedal; Wright, 2006). The mathematical formulation of constrained optimization problems can be written as (Luenberger; Ye, 2016)

$$\min \quad f(\boldsymbol{x}) \tag{7a}$$

$$s.t. \quad h_i(\boldsymbol{x}) = 0,\ i = 1, 2, \ldots, m_1 \tag{7b}$$

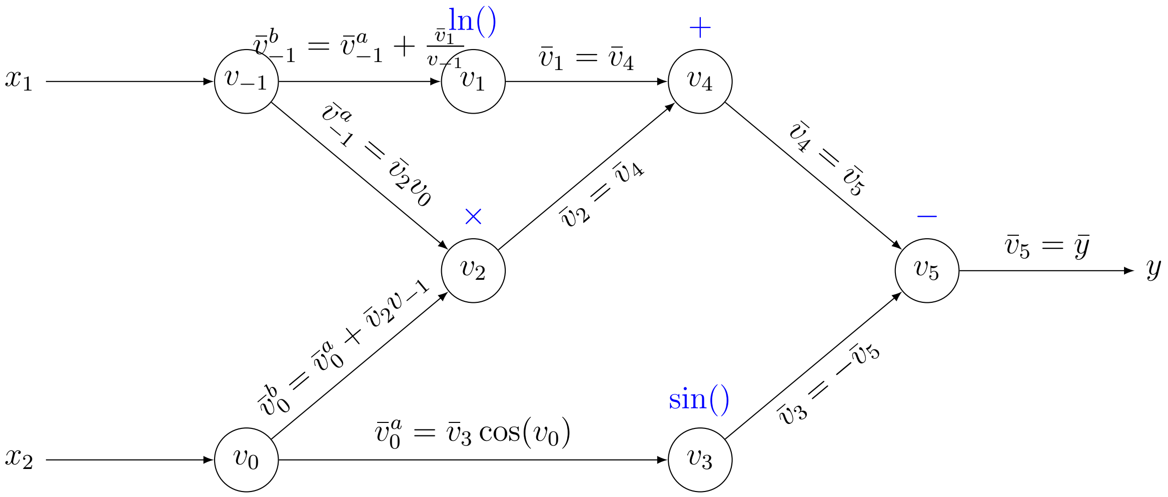
$$g_j(\boldsymbol{x}) \leq 0,\ j = 1, 2, \ldots, m_2 \tag{7c}$$

$$\boldsymbol{x} \in \Omega$$

where $\boldsymbol{x} \in \Re^n$ is the vector of n optimization variables, $f(\boldsymbol{x})$ is the nonlinear objective function, $h_i(\boldsymbol{x})$ are the m_1 equality constraints, $g_j(\boldsymbol{x})$ are the m_2 inequality constraints, and Ω is the feasible set, i.e., the set of points $\boldsymbol{x}$ that satisfy the constraints.

The optimization problem addressed in this work is a constrained Nonlinear Programming (NLP). Thus the objective function $f(\boldsymbol{x})$ is nonlinear and the constraints, h_i and g_j, can be linear or nonlinear.

Figure 3 – Computational graph of Eq. (3) with the partial derivatives from reverse AD annotated.



Source: adapted from (Baydin *et al.*, 2018).

The literature presents many algorithms to solve NLP problems. Two different approaches are presented in this section: a deterministic gradient-based algorithm and a metaheuristic algorithm.

2.2.1 Sequential Quadratic Programming

Sequential Quadratic Programming (SQP) is not a unique algorithm, but a framework to solve constrained NLP problems, i.e., the optimization algorithms need to follow the same basic principle to be considered a SQP method. It is one of the most effective methods to solve this kind of problem, specially with significant nonlinearities in the constraints (Nocedal; Wright, 2006).

The basic principle of SQP algorithms is to obtain the step direction by solving quadratic subproblems (QP), hence the name. The objective function of the NLP problem (7) is approximated by a quadratic function and the constraints are linearized. The quadratic subproblem is solved using adequate algorithms [1] , and each SQP implementation can use a different one.

For each iteration k, the NLP problem is rewritten as a QP one of the form

$$\min_{p} \quad f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}_k p \tag{8a}$$

$$s.t. \quad \nabla h_i(x_k)^T p + h_i(x_k) = 0, \; i = 1, 2, \ldots, m_1 \tag{8b}$$

$$\nabla g_j(x_k)^T p + g_j(x_k) \leq 0, \; j = 1, 2, \ldots, m_2 \tag{8c}$$

where p is the QP optimization variable and the step used for SQP and $\nabla^2_{xx}\mathcal{L}_k$ is the Hessian matrix of Lagrangian function (Nocedal; Wright, 2006). The basic SQP algorithm described in Algorithm 1 is a adaptation of (Nocedal; Wright, 2006, Algorithm 18.1).

[1] Some QP algorithms are presented in (Nocedal; Wright, 2006, Chapter 16).

Algorithm 1: Local SQP Algorithm for solving the optimization problem given by (7).

```
Choose an initial pair $(x_0, \lambda_0)$;
Set $k := 0$;
repeat
    Evaluate $f_k$, $\nabla f_k$, $\nabla^2_{xx}\mathcal{L}_k$, $\nabla h_i(x_k)^T$, $h_i(x_k)$, $\nabla g_j(x_k)^T$, and $g_j(x_k)$;
    Solve (8) to obtain $p_k$ and $\lambda_{k+1}$;
    $x_{k+1} := x_k + p_k$;
    $k := k + 1$
until a convergence test is satisfied;
```

The SQP algorithm used in this work was first developed and implemented by (Kraft, 1988). In this implementation, the hessian matrix $\nabla^2_{xx}\mathcal{L}_k$ is evaluated using an approximation similar to the ones used in quasi-newton methods for unconstrained optimization. The update of this matrix is described in (Kraft, 1988, §2.2.4).

Another specificity of this implementation is that three approaches are proposed to solve the QP subproblems: Primal, Primal/dual, and Dual methods. The Primal/dual method transform the QP problem in a linear least-squares formulation (LSEI). Another work of the same author (Kraft, 1994) implemented this approach in Fortran and named it as SLSQP, because the optimization is solved by sequential linear least-squares problems.

The SLSQP algorithm became popular with the SciPy project (Virtanen *et al.*, 2020), in which the original code was ported to Python. There are also another packages that use this algorithm, e.g., NLopt in which the code was converted to C (Johnson, 2007–2020).

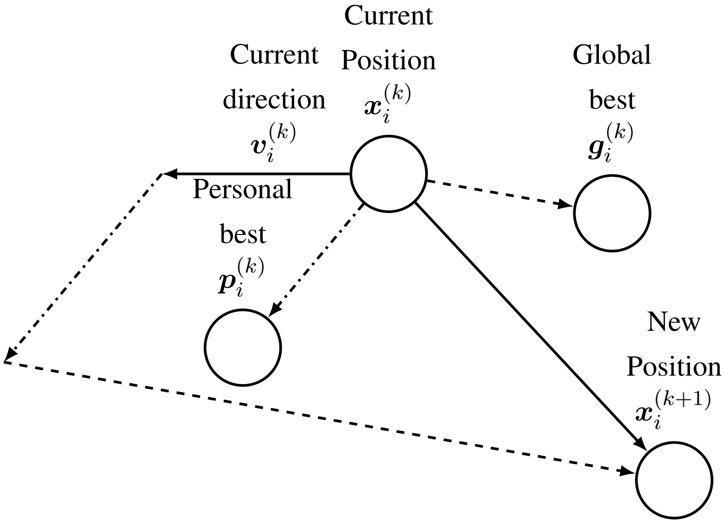
2.2.2 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) is a population-based metaheuristic optimization algorithm. Its working principle simulates the movement behaviour of flocks of birds or schools of fish (Kennedy; Eberhart, 1995). It can also be understood as particles moving in the space, trying to find the best position.

The value of the positions is quantified by the fitness function, that is unique related to the objective function or can also contain the constraints. The position update for each particle at each iteration is determined by its velocity, which is a weighting of it current inertia, an individual intelligence component, and a social intelligence component. The individual intelligence component consists of the position which generates the best fitness value of the particle so far, known as personal best $\boldsymbol{p}_i$, and for the social component the position with the best fitness value among all the particles, known as global best $\boldsymbol{g}$. Figure 4 shows a example of the position updating.

There are different versions of the PSO, but always following these basic principles.

Figure 4 – Position update of a particle using PSO.


Current Position $\boldsymbol{x}_i^{(k)}$
Current direction $\boldsymbol{v}_i^{(k)}$
Global best $\boldsymbol{g}_i^{(k)}$
Personal best $\boldsymbol{p}_i^{(k)}$
New Position $\boldsymbol{x}_i^{(k+1)}$

Source: the author.

The PSO algorithm presented in this work was adapted from (Javed *et al.*, 2020) and (Clerc; Kennedy, 2002), and it is established for a minimization problem and the equations governing the iterative process are indicated below.

In this section, all the variables named as r are a random value between 0 and 1 following a uniform distribution. The iterative process will occur during k iterations, that represents the time, for a population size of np particles, and each particle being composed of n variables, i.e., the n decision variables from the optimization problem. Also, i represents the i^{th} particle of the population and j the j^{th} variable of each particle.

Considering the problem as a bounded one, the population is initialized inside the feasible search space, using

$$x_{i,j}^{(0)} = x_j^{min} + r \cdot (x_j^{max} - x_j^{min}), \tag{9}$$

and the initial velocity for all particles and variables are zero.

The velocity is updated as

$$v_{i,j}^{(k+1)} = \phi_w \cdot v_{i,j}^{(k)} + \phi_1 \cdot r_1 \cdot \left(p_{i,j}^{(k)} - x_{i,j}^{(k)}\right) + \phi_2 \cdot r_2 \cdot \left(g_j^{(k)} - x_{i,j}^{(k)}\right), \tag{10}$$

where ϕ_w is the inertia weight, and ϕ_1 and ϕ_2 are established constants, and it is assured if its value is inside the velocity range using

$$v_{i,j}^{(k+1)} = \begin{cases} v_{i,j}^{(k+1)}, & v_j^{min} \leq v_{i,j}^{(k+1)} \leq v_j^{max} \\ v_j^{min}, & v_{i,j}^{(k+1)} < v_j^{min} \\ v_j^{max}, & v_{i,j}^{(k+1)} > v_j^{max}. \end{cases} \tag{11}$$

Finally, the new particle position is calculated by

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}, \tag{12}$$

and it is verified it its value is between the bounds with

$$x_{i,j}^{(k+1)} = \begin{cases} x_{i,j}^{(k+1)}, & x_j^{min} \leq x_{i,j}^{(k+1)} \leq x_j^{max} \\ x_j^{min}, & x_{i,j}^{(k+1)} < x_j^{min} \\ x_j^{max}, & x_{i,j}^{(k+1)} > x_j^{max}. \end{cases} \tag{13}$$

The procedure of the PSO iterative process is presented in the Algorithm 2, where $f(\cdot)$ is the fitness function.

Algorithm 2: Basic Particle Swarm Optimization algorithm for minimization problem.

```
Initialize population randomly choosing $\boldsymbol{x}_i^{(0)}$ using Eq. (9) and set all $\boldsymbol{v}_i^{(0)}$ to zero;
Calculate the fitness function $f$ for each $\boldsymbol{x}_i^{(0)}$;
Set local optima $\boldsymbol{p}_i^{(0)} := \boldsymbol{x}_i^{(0)}$;
Define global optima as $\boldsymbol{g}^{(0)} = \arg\min(f(\boldsymbol{p}_1^{(0)}), \ldots, f(\boldsymbol{p}_{np}^{(0)}))$;
Set $k = 0$;
repeat
    for $i = 1$ to population size $np$ do
        for $j = 1$ to variable dimension $n$ do
            Update particle velocity using Eq. (10) ;
            Let particle velocity inside bounds using Eq. (11) ;
        Update particle position using Eq. (12) ;
        Let th particle position inside bounds using Eq. (13) ;
        Calculate the fitness value for each $\boldsymbol{x}_i^{(k+1)}$ ;
        if $f(\boldsymbol{x}_i^{(k+1)}) < f(\boldsymbol{p}_i^{(k)})$ then
            Update the personal best for the $i^{th}$ particle $\boldsymbol{p}_i^{(k+1)} = \boldsymbol{x}_i^{(k+1)}$
        Update the global best $\boldsymbol{g}^{(k+1)} = \arg\min(f(\boldsymbol{p}_1^{(k+1)}), \ldots, f(\boldsymbol{p}_{np}^{(k+1)}))$
    $k = k + 1$;
until $k = k^{max} - 1$;
```

The PSO algorithm does not have a direct way to handle the constraints. Therefore, there are some techniques to do it and the one presented here is the penalty. The constraints are added in the fitness function multiplied by a high value constant. In (Nielsen, 2003) the penalty function with n_k constraints has the form

$$f(x) = f(x) \cdot (1 + \sum_{k=1}^{n_k} p_k), \tag{14}$$

where p_k is evaluated as

$$p_k = \max \left[0, \frac{(c_{k,l} - c_k)}{abs(c_{k,l})}, \frac{(c_k - c_{k,u})}{abs(c_{k,u})} \right], \tag{15}$$

where $c_{k,l}$ is the lower bound and $c_{k,u}$ is the upper bound on performance criterion c_k.

2.2.3 Optimization algorithms' performance indicators

In order to compare different optimization algorithms, performance indicators need to be used to compare the algorithms statistically. The first step is to obtain a representative sample of solutions for the studied search space for each algorithm and there are different fashions for different algorithm types. Deterministic algorithms are run for distinct initial points and metaheuristic ones can be run for the same initial population, because they are based on stochastic characteristics that produces different results for the same initial point.

The most basic statistics that can be used as performance indicators are the sample minimum, sample maximum, mean and median values. Others statistics related to the error and dispersion of the sample values can also be used.

In their work, Javed *et al.* (2020) propose the utilization of five statistical parameters to assess the convergence performance and accuracy of optimization algorithms. These indicators are applied to each algorithm sample and are: standard deviation (SD), mean absolute error (MAE), relative error (RLE), root mean square error (RMSE), and mean efficiency of optimization algorithm, given by Equations (16), (17), (18), (20) and (21), respectively. In these equations o_{best} is the best objective function value, which is the sample minimum for minimization problems or the sample maximum for maximization problems, o_k is the objective function value for the k^{th} algorithm execution, K is the number of algorithm executions, and $\bar{o}$ is the mean of the objective function values. In this work, Eq. (18) was modified to represent the mean relative error (MRE), presented in Eq. (19).

$$SD = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(o_k - \bar{o})^2} \tag{16}$$

$$MAE = \frac{\sum_{k=1}^{K}(o_k - o_{best})}{K} \tag{17}$$

$$RLE = \frac{\sum_{k=1}^{K}(o_k - o_{best})}{o_{best}} \tag{18}$$

$$MRE = \frac{\sum_{k=1}^{K}(o_k - o_{best})}{K \cdot o_{best}} \tag{19}$$

$$RMSE = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(o_k - o_{best})^2} \tag{20}$$

$$\eta_{mean}(\%) = 100 \cdot \frac{\sum_{k=1}^{K}\left(\frac{o_{best}}{o_k}\right)}{K} \tag{21}$$

When comparing optimization algorithms, it is also important to evaluate the computational performance, because resources are still limited even with the evolution of computers. Two computational resources that need to be compared between algorithms are computational time and memory allocation. To reduce the noise and error of time and memory estimation, multiple executions of algorithms are required, always in the same machine. There are some packages already implemented to benchmark time and memory of programs and algorithms running them multiple times , as *BenchmarkTools.jl* in Julia language (Chen; Revels, 2016).

2.3 Microgrid modeling

2.3.1 Photovoltaic systems

The PV production for each time instant ($P_{PV}(t)$), in kW, is evaluated by

$$P_{PV}(t) = f_{PV} \cdot P_{PV}^{rtd} \cdot \frac{I_T(t)}{I_S}, \tag{22}$$

where f_{PV} is the PV derating factor, P_{PV}^{rtd} is the rated power of the PV array, in kW, $I_T(t)$ is the global solar radiation (beam plus diffuse) incident on the surface of the PV array for the time t, in kW/m^2, and I_S is the standard amount of radiation used to rate de capacity of the PV array, whose value is 1 kW/m^2. The PV production $P_{PV}(t)$ can also be evaluated as

$$P_{PV}(t) = P_{PV}^{rtd} \cdot P_{PV}^{1k}(t)\,, \tag{23}$$

where $P_{PV}^{1k}(t)$ is the output of a 1 kW PV panel, already including temperature and system loss, for the time t.

For the photovoltaic model, the number of replacements N_{PV}^{rep} depends on its lifetime ℓ_{PV} and also the project lifetime ℓ_{proj}, and is calculated by

$$N_{PV}^{rep} = \left\lceil \frac{\ell_{proj}}{\ell_{PV}} \right\rceil - 1. \tag{24}$$

With the number of replacements, it's possible to calculate the PV's remaining life as

$$\ell_{PV}^{rem} = \ell_{PV} - (\ell_{proj} - \ell_{PV} \cdot N_{PV}^{rep})\,. \tag{25}$$

2.3.2 Wind turbines

Usually the wind turbines' hubs are not installed at the same height of the anemometer. Thus it is necessary to adjust the wind speed to the hub height U_{hub}, that can be calculated for each time instant t using the logarithmic law

$$U_{hub}(t) = U_{anem}(t)\,\frac{\ln(z_{hub}/z_0)}{\ln(z_{anem}/z_0)} \tag{26}$$

or the exponential law

$$U_{hub}(t) = U_{anem}(t)\left(\frac{z_{hub}}{z_{anem}}\right)^{\alpha_{WT}}, \tag{27}$$

where $U_{anem}(t)$ is the wind speed measured at the anemometer height at time t, z_{hub} is the hub height, z_{anem} is the anemometer height, z_0 is the surface roughness length, and α_{WT} is the power law exponent.

The wind turbine power output for each wind speed is determined using power curves from manufacturers or approximations of these curves. In this work, it is used a quadratic approximation given by

$$P_{WT}(t) = \begin{cases} P_{WT}^{rtd} \cdot \left(\frac{U_{hub}(t)-U_{cut-in}}{U_{rtd}-U_{cut-in}}\right)^2, & U_{cut-in} \leq U_{hub}(t) < U_{rtd} \\ P_{WT}^{rtd}, & U_{rtd} \leq U_{hub}(t) \leq U_{cut-out} \\ 0, & otherwise, \end{cases} \tag{28}$$

where P_{WT}^{rtd} is the rated wind turbine power, U_{rtd} is the rated wind speed, U_{cut-in} is the cut-in wind speed, i.e., the wind speed at which the turbine starts to work, and $U_{cut-out}$ is the cut-out wind speed, i.e., the wind speed at which the turbine stops to work.

The number of replacements and remaining life for the wind turbine model are similar to the ones of the photovoltaic model, and are given by

$$N_{WT}^{rep} = \lceil \frac{\ell_{proj}}{\ell_{WT}} \rceil - 1 \tag{29}$$

and

$$\ell_{WT}^{rem} = \ell_{WT} - (\ell_{proj} - \ell_{WT} \cdot N_{WT}^{rep}) \tag{30}$$

respectively, where l_{WT} is the wind turbines lifetime.

2.3.3 Battery energy storage systems

At each time step, the battery energy E_{BT} is updated according to

$$E_{BT}(t+1) = E_{BT}(t) - (P_{BT}(t) + \alpha_{BT} \left|P_{BT}(t)\right|) \Delta t, \tag{31}$$

where $P_{BT}(t)$ is the battery power in generator convention, i.e. the battery is charging when $P_{BT}(t) < 0$ and discharging when $P_{BT}(t) > 0$ and α_{BT} is the linear loss factor.

The charging and discharging power bounds depend on the rate limits and the energy limits of the battery as a result of the discrete time modeling. The superior bound implied by the energy is

$$P_{BT}^{e,max}(t) = \frac{E_{BT}(t) - E_{BT}^{min}}{(1+\alpha_{BT}) \Delta t} \tag{32}$$

where E_{BT}^{min} is the minimum available energy, and the inferior bound is

$$P_{BT}^{e,min}(t) = -\frac{E_{BT}^{max} - E_{BT}(t)}{(1-\alpha_{BT}) \Delta t}, \tag{33}$$

where E_{BT}^{max} is the maximum available energy.

The considered limits are the most restrictive of the two types, rate and energy limits. Therefore, the discharge limit is

$$P_{BT}^{discharge}(t) = \min\left(P_{BT}^{max}, P_{BT}^{e,max}(t)\right) \tag{34}$$

and the charge limit is

$$P_{BT}^{charge}(t) = \max\big(P_{BT}^{min}, P_{BT}^{e,min}(t)\big), \tag{35}$$

where P_{BT}^{max} and P_{BT}^{min} are respectively the discharge and charge power limits, which depend on the battery type chosen.

The maximum available energy E_{BT}^{max} is equal to the rated energy E_{BT}^{rtd} and the minimum available energy E_{BT}^{min} is equal to zero, because it is considered a change of scale to work only with the energy actually available from the battery, and not having to use the state of energy (SOE) concepts.

The battery lifetime depends on the time, ℓ_{BT}^{years}, of use and also in the energy that cycles through it. The throughput lifetime is

$$\ell_{BT}^{thrpt} = \frac{2 \cdot E_{BT}^{rtd} \cdot N_{cycles}}{Q_{BT}^{thrpt}}, \tag{36}$$

where N_{cycles} is the maximum number of complete cycles, i.e. number of the charge and discharge, and Q_{BT}^{thrpt} is the total energy that cycles through the battery yearly, calculated by

$$Q_{BT}^{thrpt} = \sum_{t=1}^{T} |P_{BT}(t)| \cdot \Delta t. \tag{37}$$

The battery lifetime is the most restrictive of the two types, thus it's calculated as

$$\ell_{BT} = \min\left(\ell_{BT}^{thrpt}, \ell_{BT}^{years}\right). \tag{38}$$

Hence, the BT number of replacements N_{BT}^{rep} is

$$N_{BT}^{rep} = \left\lceil \frac{\ell_{proj}}{\ell_{BT}} \right\rceil - 1 \tag{39}$$

and the remaining life is

$$\ell_{BT}^{rem} = \ell_{BT} - \left(\ell_{proj} - \ell_{BT} \cdot N_{BT}^{rep}\right). \tag{40}$$

2.3.4 Diesel generating set

The GS fuel consumption $F_{GS}(t)$ in liters at each time instant is evaluated by

$$F_{GS}(t) = \big(F_0 \cdot P_{GS}^{rtd} + F_1 \cdot P_{GS}(t)\big) \cdot \Delta t \tag{41}$$

where the F_0 is the fuel curve intercept coefficient in $l/h/kW$, F_1 is the fuel curve slope in $l/h/kW$, P_{GS}^{rtd} is the rated power of the generator in kW and $P_{GS}(t)$ is the GS electrical output in kW for the time t.

The total fuel consumption in one year is

$$F_{GS}^{tot,year} = \sum_{t=1}^{T} F_{GS}(t)\,. \tag{42}$$

If the GS is on, the model considers that it works during all the timestep Δt at the power $P_{GS}(t)$. This is expressed mathematically as

$$h_{GS}(t) = \begin{cases} 0, & P_{GS}(t) = 0 \\ \Delta t, & 0 < P_{GS}(t) \leq P_{GS}^{rtd} \end{cases} \tag{43}$$

The GS total operation hours in one year is

$$h_{GS}^{tot,year} = \sum_{t=1}^{T} h_{GS}(t) \tag{44}$$

and during all the project lifetime is

$$h_{GS}^{tot,proj} = h_{GS}^{tot,year} \cdot \ell_{proj}\,. \tag{45}$$

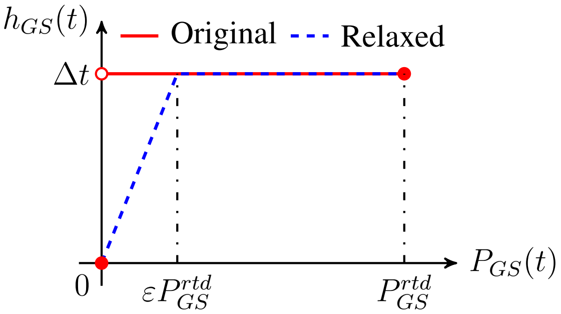
However, the GS operation hours expressed in (43) adds a discontinuity in the microgrid operation model, interfering in the gradient-based algorithms convergence. Thus, a relaxation is implemented as

$$h_{GS}^{rlx}(t) = \begin{cases} \dfrac{\Delta t}{\varepsilon} \dfrac{P_{GS}(t)}{P_{GS}^{rtd}}, & P_{GS}(t) \leq \varepsilon P_{GS}^{rtd} \\ \Delta t, & \varepsilon P_{GS}^{rtd} < P_{GS}(t) \leq P_{GS}^{rtd} \end{cases} \tag{46}$$

where ε is the relaxation factor, and the total operation hours in one year can be calculated by replacing $h_{GS}(t)$ with $h_{GS}^{rlx}(t)$ in (44).

The Fig. 5 shows the curves for the GS operations hours models expressed in (43) and (46).

Figure 5 – Diesel generating set consumption hours.



Source: the author.

The GS number of replacements during the project is

$$N_{GS}^{rep} = \left\lceil \frac{h_{GS}^{tot,proj}}{\ell_{GS}} \right\rceil - 1 \tag{47}$$

where ℓ_{GS} is the GS lifetime in hours of operation, and the remaining life in operation hours is

$$\ell_{GS}^{rem} = \ell_{GS} - \left(h_{GS}^{tot,proj} - \ell_{GS} \cdot N_{GS}^{rep}\right) . \tag{48}$$

2.3.5 Power balance

In the microgrid model used in this work, the losses due to the network are not considered. Thus, the system is seen as a one single node, in which all the power production is entering and all load consumption is exiting.

For a microgrid containing PV, WT, BT and GS, the net power is calculated as

$$P_{net}(t) = P_{PV}(t) + P_{WT}(t) + P_{BT}(t) + P_{GS}(t) - P_{load}(t) , \tag{49}$$

where P_{load} is the load demand.

The power balance is assured with the use of two slack variables, P_{curt} that represents the amount of renewables power that is in excess, i.e. the curtailed renewables power, and P_{shed} represents the unmet load, i.e. the shedding load. The balance is then obtained as

$$P_{net}(t) - P_{curt}(t) - P_{shed}(t) = 0. \tag{50}$$

Due to the nondispatchable characteristic of the renewables sources, PV and WT in this study, another intermediate parameter is established as

$$P_{load}^{net}(t) = P_{load}(t) - P_{PV}(t) - P_{WT}(t) . \tag{51}$$

A positive value of this power indicates the production of the renewable sources was not sufficient to supply all the load, meaning that others powers sources need to be used, and a negative value means that there is a surplus of renewables production. The use of $P_{load}^{net}(t)$ when a battery composes the microgrid is presented in the next section.

2.4 Dispatch strategies

The microgrids are usually composed of more than one energy source, as a form to improve the reliability of the grid. Therefore a power dispatch strategy needs to be set in order to define the sources that will operate to supply the load. The most common implemented strategies are the rule-based and optimal dispatch, as presented in §2.6. As the name suggests, the rule-based strategy is characterized by a fixed rule to decide which source will supply the load. In the optimal strategy, at each time of operation, a optimization problem is solved to determine the source that will operate. Usually the optimization problem aims to minimize the costs.

In a isolated microgrid whose purpose is to increase the use of renewable sources, using the diesel genset only as a backup, a rule-based strategy may be adopted. A reasonable priority order could be:

1. Nondispatchable renewable sources, e.g., photovoltaic, wind power and run-of-river hydro power

2. Dispatchable renewable sources, e.g., biomass and hydro power with reservoir

3. Batteries

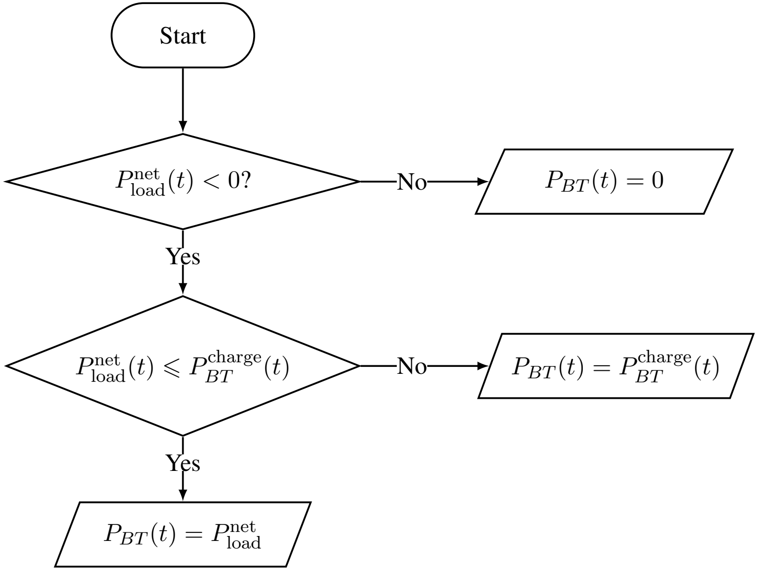
4. Diesel generating set

2.4.1 Battery-charging strategies

Considering a microgrid configuration composed of nondispatchables sources, batteries and a diesel genset, two main deterministc strategies are implemented to determine how the batteries will be charged.

The first one is the load-following strategy, in which the batteries are charged only by the nondispatchables sources, when these produce energy in excess (Lambert; Gilman; Lilienthal, 2006). In this case, the diesel genset will never produce energy to charge the batteries, even though it is operating to supply the load. Figure 6 shows the fluxogram for the battery-charging load-following strategy.

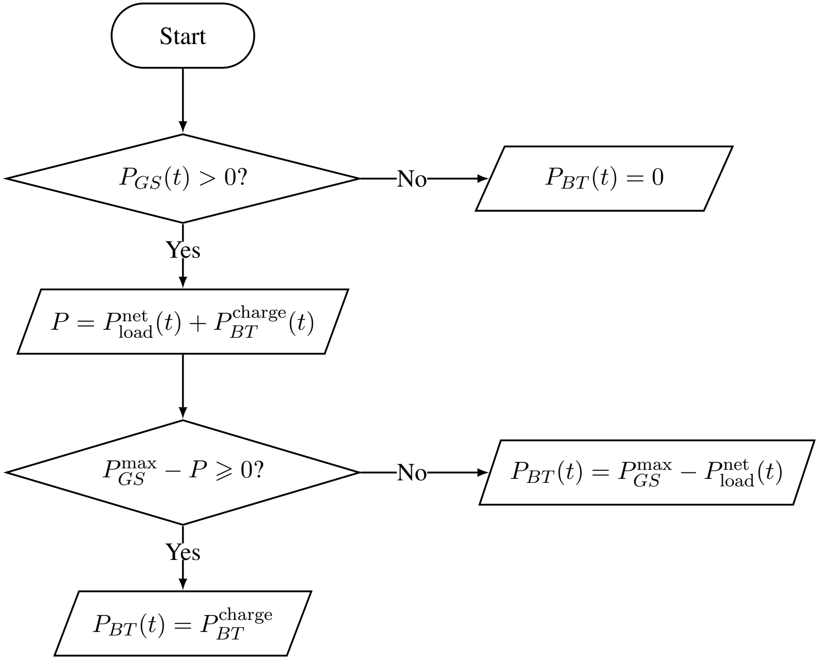
Figure 6 – Load-following dispatch fluxogram.


Start
$P^{\text{net}}_{\text{load}}(t) < 0?$
No
$P_{BT}(t) = 0$
Yes
$P^{\text{net}}_{\text{load}}(t) \leqslant P^{\text{charge}}_{BT}(t)$
No
$P_{BT}(t) = P^{\text{charge}}_{BT}(t)$
Yes
$P_{BT}(t) = P^{\text{net}}_{\text{load}}$

Source: the author.

In the cycle-charging strategy, whenever the rule-based strategy determine the operation of the diesel genset to supply the load, it will also produce more energy to charge the batteries, respecting its operation constraints (Lambert; Gilman; Lilienthal, 2006). The battery-charging process with cycle-charging strategy is shown in Fig. 7.

Figure 7 – Cycle-charging dispatch fluxogram.


Start
$P_{GS}(t) > 0?$
No
$P_{BT}(t) = 0$
Yes
$P = P_{\text{load}}^{\text{net}}(t) + P_{BT}^{\text{charge}}(t)$
$P_{GS}^{\max} - P \geqslant 0?$
No
$P_{BT}(t) = P_{GS}^{\max} - P_{\text{load}}^{\text{net}}(t)$
Yes
$P_{BT}(t) = P_{BT}^{\text{charge}}$

Source: the author.

2.5 Indicators

2.5.1 Energetic indicators

There are some energetic indicators that are used to characterize the microgrids. This section presents two of them, and the first presented is the renewable fraction (RF) which is definied as

$$RF\,(\%) = 100 \cdot \frac{\sum\limits_{t=1}^{T} P_{ren}(t) \cdot \Delta_t}{E_{serv}} \tag{52}$$

where P_{ren} is the sum of the power produced by renewable sources at for time t and E_{serv} is the load energy served in one year.

The other is the shedding fraction (SF), that represents the fraction of demand not supplied and it is calculated by

$$SF\,(\%) = 100 \cdot \frac{\sum\limits_{t=1}^{T} P_{shed}(t)}{\sum\limits_{t=1}^{T} P_{load}(t)} \tag{53}$$

where $P_{shed}(t)$ is the power not supplied at each time t.

2.5.2 Economic indicators

In the economic model, the costs are analyzed in the beginning of the project. Hence, the investment, operation and maintenance (OM) and replacement costs and the salvage value are brought to the present considering the project discount rate d.

In the following equations, $c \in \mathscr{C} = \{PV, WT, BT, GS\}$, and

$$\phi(c) = \begin{cases} P_{PV}^{rtd}, & c = PV \\ P_{WT}^{rtd}, & c = WT \\ E_{BT}^{rtd}, & c = BT \\ P_{GS}^{rtd}, & c = GS \end{cases} \tag{54}$$

The discount factors f_{di} are calculated for each year i of project according to

$$f_{di} = \frac{1}{(1+d)^i}, i \in \{1, 2, \dots, \ell_{proj}\}. \tag{55}$$

We can define the summation of these factors as

$$\sigma_{f_d} = \sum_{y=1}^{\ell_{proj}} f_{dy}. \tag{56}$$

If the number of replacements of the components is different of zero, it is necessary to calculate the years when the replacements happen. The sets that contains these years for each component are defined as

$$\mathscr{Y}_c = \{y \cdot \ell_c \mid y \in \{1, 2, \dots, N_c^{rep}\}\}, \tag{57}$$

and with this set defined, the replacement factors $f_{c,i}^{rep}$ for each component are calculated using

$$f_{c,i}^{rep} = \frac{1}{(1+d)^i}, \ i \in \mathscr{Y}_c. \tag{58}$$

With the factors calculated in (55) and (58), it is possible to evaluated the present costs for each component and the total net present cost (NPC) for the microgrid project.

The present investment cost for each component is

$$C_c^{inv,tot} = C_c^{inv} \cdot \phi(c). \tag{59}$$

The OM present costs are evaluated by the following set of equations

$$C_{PV}^{OM,tot} = C_{PV}^{OM} \cdot P_{PV}^{rtd} \cdot \sigma_{f_d} \tag{60a}$$

$$C_{WT}^{OM,tot} = C_{WT}^{OM} \cdot P_{WT}^{rtd} \cdot \sigma_{f_d} \tag{60b}$$

$$C_{BT}^{OM,tot} = C_{BT}^{OM} \cdot E_{BT}^{rtd} \cdot \sigma_{f_d} \tag{60c}$$

$$C_{GS}^{OM,tot} = C_{GS}^{OM} \cdot P_{GS}^{rtd} \cdot h_{GS}^{tot,year} \cdot \sigma_{f_d} + C_{GS}^{fuel,tot} \tag{60d}$$

where the total fuel cost is

$$C_{GS}^{fuel,tot} = C_{GS}^{fuel} \cdot F_{GS}^{tot,year} \cdot \sigma_{f_d}. \tag{61}$$

The present replacement cost is

$$C_c^{rep,tot} = C_c^{rep} \cdot \phi(c) \cdot \sum_{y \in \mathscr{Y}_c} f_{c,y}^{rep}. \tag{62}$$

If the components have a remaining life different of zero, a salvage value needs to be calculated. The proportional unitary salvage value is

$$S_c = C_c^{rep} \cdot \frac{\ell_c^{rem}}{\ell_c} \tag{63}$$

and the total salvage value is

$$S_c^{tot} = S_c \cdot \phi(c) \cdot f_{d\ell_{proj}}. \tag{64}$$

The total component present cost is

$$C_c^{tot} = C_c^{inv,tot} + C_c^{OM,tot} + C_c^{rep,tot} - S_c^{tot}. \tag{65}$$

Finally, the microgrid NPC is calculated by the summation

$$NPC = \sum_{c \in \mathscr{C}} C_c^{tot}. \tag{66}$$

Another economic indicator that can be used is the levelized cost of energy (COE) given by

$$COE = \frac{NPC}{E_{serv}^{proj}} \cdot (\sigma_{f_d})^{-1}. \tag{67}$$

where the E_{serv}^{proj} is the total energy served during the project lifetime.

2.6 Literature review

A genetic algorithm was implemented by (Zhao *et al.*, 2014) to obtain the optimal sizing of a islanded microgrid. The results were used to developed a real microgrid on Dongfushan Island, China. The energy resources considered in the optimization were the photovoltaic system (PV), wind turbines (WT), battery storage (BS), and a diesel generating set (GS). A particularity of the system is a seawater desalination system (SWDS), modeled as a controllable load allowing better use of the nondispatchable renewable energy sources. The problem was posed as a continuous multicriteria optimization, minimizing a objective function composed of the normalized and weighted cost of energy (COE), renewables fraction (RF) and diesel genset greenhouse gases emission. Seven weight sets were tested to obtain the best values for different criteria priorities, where the resulted RF ranged between 55.26% to 61.20%. As the rated power and capacity of commercial components are discrete, the actual chosen solution differ from the ones found with GA. This and the uncertainties of climate data lead to an actual lower RF during the period between August 2011 to December 2011. Therefore, a stochastic analysis of the uncertainties or the consideration of different scenarios is important to a well designed microgrid.

The Distributed Energy Resources Customer Adoption Model (DER-CAM) program, developed by Lawrence Berkeley National Laboratory/USA, determines the optimal technology portfolio, the optimal technology placement, and the associated optimal dispatch developed. An extension of the matemathical formulation of DER-CAM was made in (Mashayekh *et al.*, 2017). In this work the optimization objective is to minimize the costs associate with a multi-energy microgrid, i.e. electrical and thermal energies, considering the power flow equations. Some components capacities are modeled as continuous variables, e.g. photovoltaic (PV), and others as discrete variables, e.g. micro-turbines (MT). Since the MILP formulation leads to a high variables number, and it is dependent with the number of nodes and points in demand and climate hourly data, a typical year of operation is modeled with 3 representative days for each month (864 different hourly values) and a network with 5 nodes.

In (Bukar; Tan; Lau, 2019) work, the Grasshopper Optimization Algorithm (GOA), a metaheuristic optimization algorithm, was used to design a optimal islanded microgrid. The microgrid configuration is composed of a photovoltaic system (PV), wind turbines (WT), a battery storage system (BESS) and a diesel generating set (GS), which is important to provide a reliable autonomous system. A rule-based with cycle-charging dispatch strategy is used, where the dispatch priority is: renewable sources, battery, and diesel genset. It is used three representative daily demand profiles, one for each regional season, e.g., cold, hot, and rainy. Thus, the yearly demand is only represented by 72 different values, a small sample considering that the year could have 8760 different values. The hourly climate data used were obtained from measurements for a specific year. The objective function is the cost of energy (COE), and the deficiency of power supply probability (DPSP) and renewable energy fraction (RF) are constraints, and the variables were modeled as integer values. For the optimal sizing, the demand was supplied with the following distribution: 44% by the PV, 14% by the WT, 26% by the BSS and 16% by the GS supply.

A performance comparison of four metaheuristic algorithms for optimization of islanded microgrids was presented in (Javed *et al.*, 2020). It used statistical indicators, standard deviation (SD), mean absolute error (MAE), relative error (RLE), root mean square error (RMSE) and mean efficiency (η_{mean}), (given by Equations (16), (17), (18), (20) and (21)) to evaluate the exploration and exploitation of each algorithm. The case studies were performed for a microgrid composed of photovoltaic system (PV), wind turbines (WT), battery (BT) and pumped hydro storage (PHS). The objective function to be minimized is the net present cost (NPC) and the constraint is the loss of power supply probability (LPSP), related to the system reliability. The demand data is hourly for a year, containing seasonal variation, and the solar and wind data are hourly from measurements. The firefly algorithm (FA) and particle swar optimization (PSO) presented the best results. FA presented the lowest RLE and PSO was capable of found the lower value for the objective function. In terms of mean efficiency, both have the same value of 96.20%.

A hybrid metaheuristic optimization algorithm, PSO-GWO, was proposed by (Suman; Guerrero; Roy, 2021) to solve the islanded microgrid optimal sizing problem. The algorithm was composed of elements from Particle Swarm Optimization (PSO) and Grey Wolf Optimizer (GWO), and its performance was compared with five others metaheuristic algorithms. The electric sources considered are photovoltaic system, wind turbines, a biogenerator, batteries, and a diesel genset. It was used a single demand curve, representing the hourly variations in a day, and hourly climate data for a year. Two criteria are considered as objective functions, the COE and the DPSP, that were aggregated into a single objective by linear scalarisation. The renewable factor (RF) was used as a constraint and it was handled by not allowing infeasible solutions in the population. The same performance indicators proposed by (Javed *et al.*, 2020) were used for the algorithms comparison, and the proposed algorithm presented the best mean efficiency η_{mean} for the majority of case studies, reaching values of 97.54% and 97.84%.

2.6.1 Observations and considerations about the presented studies

As could be observed, the most used algorithm for microgrid optimal sizing problem is the metaheuristic ones. Linearizations are also applied in the models in order to use existent solvers for MILP. Among the presented studies, the one by (Mashayekh *et al.*, 2017) was the only one that implemented a multi-node approach, enabling the use of power flow equations and the optimal placement. However, the MILP formulation makes the dispatch problem anticipative, because all future renewables production is already know in the time of the actual dispatch decision.

Table 4 shows the electrical resources used by each work and the ones implemented in this study, where:

- Reference A - (Zhao *et al.*, 2014)
- Reference B - (Mashayekh *et al.*, 2017)
- Reference C - (Bukar; Tan; Lau, 2019)
- Reference D - (Javed *et al.*, 2020)
- Reference E - (Suman; Guerrero; Roy, 2021)
- Reference * - This study

Table 4 – Comparison of electric resources used.

Electric resource	A	B	C	D	E	*
Electric grid		×				
Photovoltaic system	×	×	×	×	×	×
Wind turbine	×		×	×	×	×
Battery	×	×	×	×	×	×
Pumped hydro storage				×		
Bio generator					×	
Diesel genset	×		×		×	×
CHP-enabled ICE		×				

Gradient-based algorithms are also deterministic as the MILP ones, i.e. the same result is obtained for the same initial conditions. They use the gradients of objective and constraints functions to determine the direction of the next step. According to (Nielsen, 2003; Hemker *et al.*, 2008) gradient-based algorithms are not suitable for simulation-based optimization problems, because it usually fails in the presence of discontinuities or non differentiable functions. Besides, if an analytical function for the gradient is not available, it has to be obtained numerically. The finite difference is frequently used as the numerical method, but the approximation errors and extra function evaluations in each iteration make this a low quality approach (Nielsen, 2003).

The Automatic Differentiation (AD) reduces the approximation errors and the computation time needed for function evaluations. Moreover, after an extensive research, there are no studies showing real evidence that gradient-based algorithms do not work for islanded microgrid optimal sizing problems. Therefore, this work presents the empirical analysis of the convergence of this approach, as well as the implication of a relaxation to reduce discontinuities.

3 METHODOLOGY

This chapter describes the proposed methodology to design an optimal islanded microgrid. First, the MG simulator architecture is presented, with the considerations made in its development. Afterward, the proposed optimization algorithm is introduced, detailing the coupling between the MG Simulator, AD package and gradient-based optimization algorithm. Finally, it is presented the procedure to compare the results obtained with the proposed methodology and the ones obtained with PSO algorithm.

3.1 Microgrid simulator

The most used simulator software to design microgrids is the commercial software HOMER. However, its proprietary characteristic does not allow significant modifications in the power sources and storage models, much less an interaction with other optimization algorithms and packages. Therefore, a Microgrid Simulator was developed following some major characteristics present in HOMER.

The MG Simulator architecture is presented in Fig. 8, which will be used to explain its implementation. The first step of the MG Simulator is to provide the necessary data to perform the energetic simulation, which are represented by items *Input Data* and *Sizing Data* in Fig. 8. Although the *Sizing Data* are also *Input Data*, they are represented separately because of their future use in optimization, where they will be decision variables.

The *Input Data* consists of:

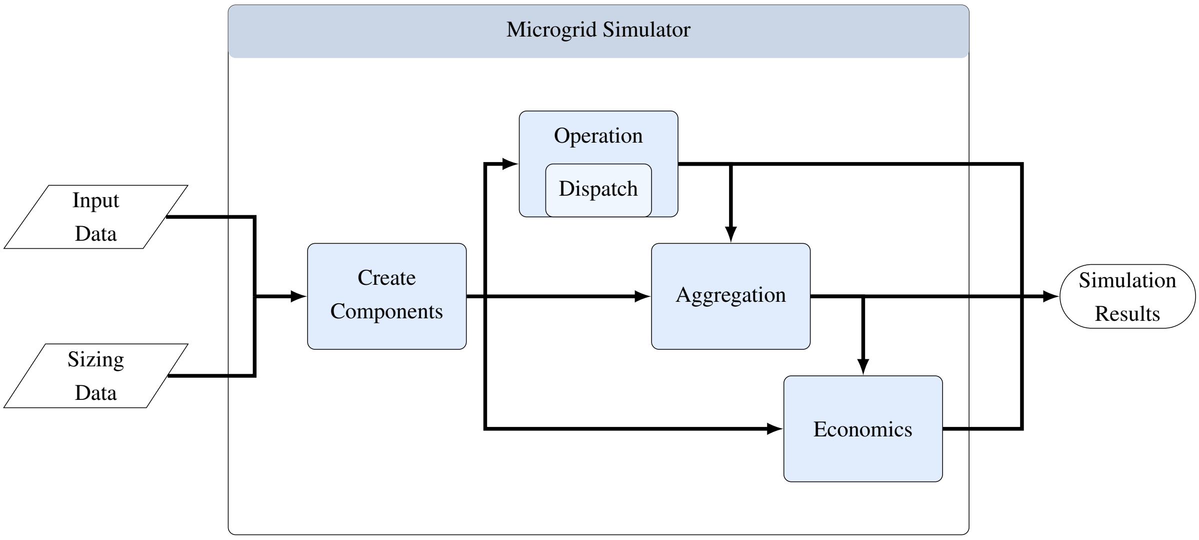
- load demand curve;
- climate data: incident global solar radiation, wind speed;
- photovoltaic data: derating factor, lifetime;
- wind turbine data: rated, cut-in and cut-out speeds, hub height, wind speed measurement height, roughness length, lifetime;
- battery data: initial energy, minimum energy level, maximum charge and discharge power, linear loss factor, maximum number of cycles, lifetime;

- diesel generating set data: minimum load ratio, fuel curve intercept coefficient, fuel curve slope, lifetime;
- costs for each component: installation, operation and maintenance, fuel and replacement costs, and salvage unitary value;
- project specification: lifetime, discount rate, simulation time step.

The *Sizing Data* consists of:

- the photovoltaic rated power;
- the wind turbine rated power;
- the battery rated energy capacity;
- the diesel generating set rated power.

Figure 8 – Microgrid Simulator architecture.



Source: the author.

After the inclusion of the data, the next step of the MG simulator, is to create the components, *Create Components* in Fig. 8. To facilitate the calculations, these components are programmed as new *variables types*, that are named as *Photovoltaic*, *WindPower*, *Battery*, *DieselGenerator*, *Project* and *Microgrid*. The *Microgrid* type is composed by the others ones, including all the components information needed for the functions *Operation*, *Aggregation* and *Economics*. It is not obligatory to have all the power sources or storage, thus, this is the step in which the microgrid configuration is chosen.

The simulation of microgrid operation actually happens in the step *Operation*. The production of the nondispatchable power sources are computed for a typical year, and the dispatch of the diesel gensets or batteries is decided for each time step accordingly with the dispatch strategy implemented, represented by *Dispatch* in Fig. 8. The outputs from this step are the operation results for each time instant, namelly:

- net load after dispatch;
- load shedding power;
- diesel generating set power;
- battery energy and power;
- battery minimum and maximum power limits for charge and discharge, and;
- renewables curtailment power.

Using the results from *Operation*, the *Aggregation* computes the totals, maximums and other energetic indicators. They are:

- the load energy served in one year;
- the maximum load shedding power;
- the maximum consecutive duration of load shedding;
- the load shedding energy in one year;
- the shedding fraction for the typical year;
- the number of diesel generating set operation hours in one year;
- the diesel generating set's fuel consumption in one year;
- the number of completed battery cycles in one year;
- the maximum renewables curtailment power, and;
- the renewables rate for the typical year.

The last step, *Economics* in Fig. 8, is the computation of the economics indicators and total costs. Using the components lifetime information, it is verified the necessity of components replacements during the project lifetime. Then it is obtained the present investment, replacement, operation and maintenance and total costs, and salvage value, for each component. These costs are also computed for the whole microgrid configuration, as well as the levelized cost of energy.

The results from *Operation*, *Aggregation* and *Economics*, are the MG simulator outputs. They can be used to graphical creation and analysis, comparison between different microgrid configuration, sensitivity analysis, or as input for optimization algorithms, as in this study.

A validation of the developed microgrid simulator is presented in Appendix A.

3.2 Optimization algorithm

The main idea of the proposed methodology is to use a gradient-based algorithm to solve a simulation based optimization problem, specifically the optimal design of an islanded microgrid considering its operation. To do so, three elements are required:

1. a microgrid simulator;
2. a gradient-based optimization algorithm, and;
3. a package to evaluate the gradients.

As mentioned before, the development of a MG Simulator was performed to allow the interface with different packages and optimization algorithms, which is a common practice when designing optimal microgrids. In this work, the major differences from other studies are in the 2^{nd} and 3^{rd} elements.

As presented in §2.6, usually this type of problem uses metaheuristic optimization algorithms instead of gradient-based ones. The expectation of using a gradient-based algorithm is to improve the optimization process in respect to the convergence, constraints handling and processing time.

In the 3^{rd} element, an Automatic Differentiation package is used, which is an efficient way to evaluate the gradients. Thus, no significant losses of the performance improvement are expected.

Recalling, the microgrid optimal sizing problem is generic described as a nonlinear constrained optimization problem of the form

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ s.t. \quad & \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} \\ & \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0} \\ & \boldsymbol{x} \in \Omega \end{aligned}$$

where $\boldsymbol{x}$ is the vector of optimization variables, $f(\boldsymbol{x})$ is the objective function, $\boldsymbol{h}(\boldsymbol{x})$ is the vector of equality constraints and $\boldsymbol{g}(\boldsymbol{x})$ is the vector of inequality constraints.

The first step is to initialize the MG Simulator furnishing the *Input Data*, that are fixed parameter during the optimization. After, it is provided to the optimization algorithm the optimization parameters, e.g., if it is a minimization or maximization problem, the stopping criteria, which are the decision variables and its bounds, and which are the objective and constraints functions. These functions can be any output from *Aggregation* or *Economics* MG Simulator steps. The decision variables are from the *Sizing Data*, and they are chosen accordingly with the microgrid configuration being optimized.

Figure 9 illustrates the interaction among the three methodology elements. After the properly initialization described above, the optimization algorithm calls the MG Simulator and the AD package, and sends them the *Sizing Data*, here represented by $\boldsymbol{x}$. The MG Simulator calculates the objective and the constraints functions for the Sizing values (1b). The AD package has access to the MG Simulator, and so forth is able to evaluate the gradients $\nabla f(\boldsymbol{x})$, $\nabla h_i(\boldsymbol{x})$ and $\nabla g_i(\boldsymbol{x})$. These last two steps can be executed in an arbitrary order or even in parallel, depending on the implementation of the optimization algorithm. The optimization algorithm receives all these values and is capable of calculating the step and obtain the next decision variable vector $\boldsymbol{x}$. This iterative process continues until the stopping criteria is satisfied, and then the optimization algorithm returns the optimal point $\boldsymbol{x}^\star$, i.e., the optimal microgrid sizing.

3.2.1 Optimization convergence assessment

As mentioned before, the objective and constraints functions need to be smooth for a gradient-based algorithm work properly. However, the used models implemented in the MG Simulator introduces discontinuities in the indicators used as these functions.

For this reason, relaxations in the modeling are frequently used to smooth the functions and improve the correct operation of the gradient-based algorithm. The intensity of this smoothing establishes a trade-off between the algorithm convergence and model accuracy. Therefore, the optimization algorithm convergence and the model accuracy need to be assessed.

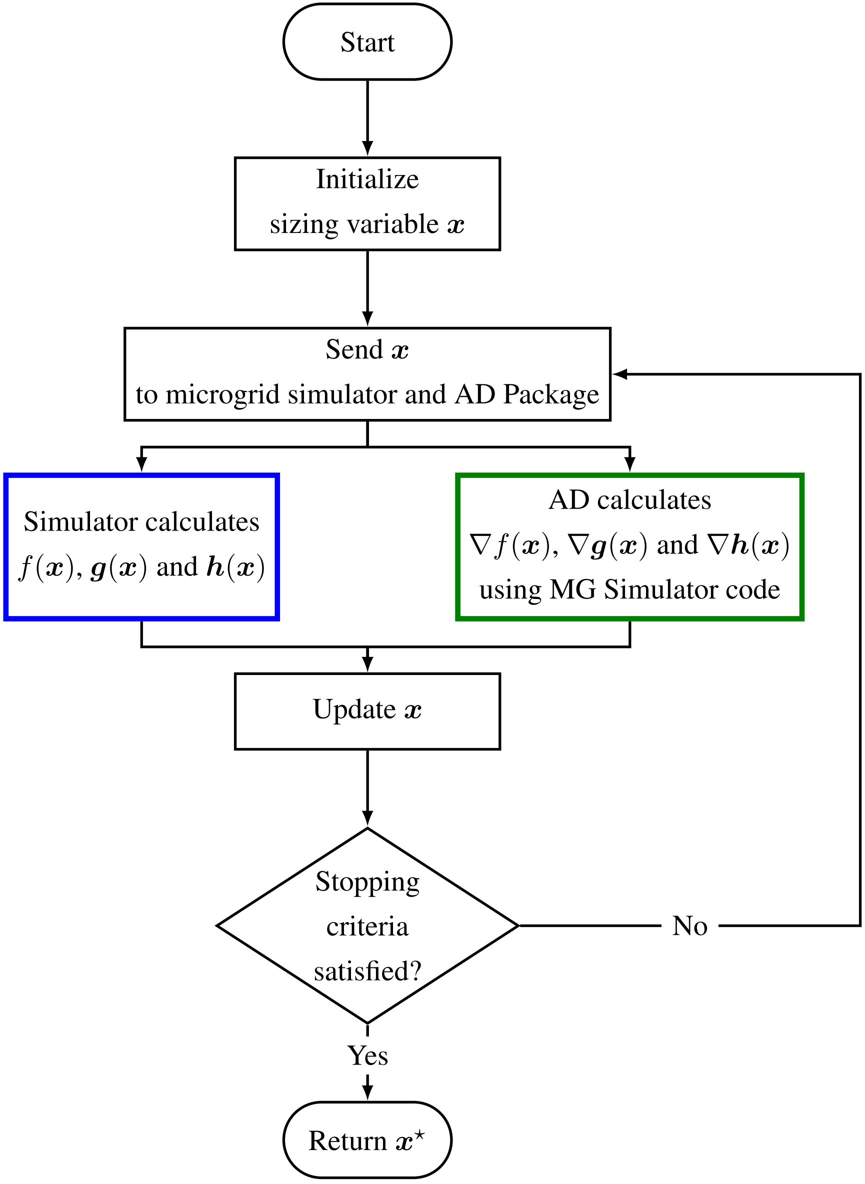
Analyzing the microgrid models, the component modeling that seems to interpose more discontinuity problem is the diesel generating set one. More specifically, due to the Eq. (43), and that is why a relaxation was proposed in Eq. (46). Therefore, there is an original microgrid modeling and a relaxed microgrid modeling to be analyzed.

Further, the initial point plays an important role in the convergence of gradient-based optimization algorithms, especially in non-convex problems because of the existence of local optimums. Thus, an evaluation for many initial points is also necessary.

The process of convergence assessment for the original modeling and the relaxed modeling is presented in Fig. 10, where x^0 is the initial decision variables vector, x^* is the optimal point obtained with the original modeling, x^*_{rlx} is the optimal point obtained with the relaxed modeling, $f(x^*)$ is the original modeling's objective function evaluated at x^*, $f_{rlx}(x^*_{rlx})$ is the relaxed modeling's objective function evaluated at x^*_{rlx}, and $f(x^*_{rlx})$ is the original modeling's objective function evaluated at x^*_{rlx}.

To evaluate the convergence behavior, N initial points i uniformly distributed in the

Figure 9 – Fluxogram of the proposed gradient-based+AD algorithm.


Start
Initialize
sizing variable $\boldsymbol{x}$
Send $\boldsymbol{x}$
to microgrid simulator and AD Package
Simulator calculates
$f(\boldsymbol{x})$, $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$
AD calculates
$\nabla f(\boldsymbol{x})$, $\nabla \boldsymbol{g}(\boldsymbol{x})$ and $\nabla \boldsymbol{h}(\boldsymbol{x})$
using MG Simulator code
Update $\boldsymbol{x}$
Stopping
criteria
satisfied?
No
Yes
Return $\boldsymbol{x}^\star$

Source: the author.

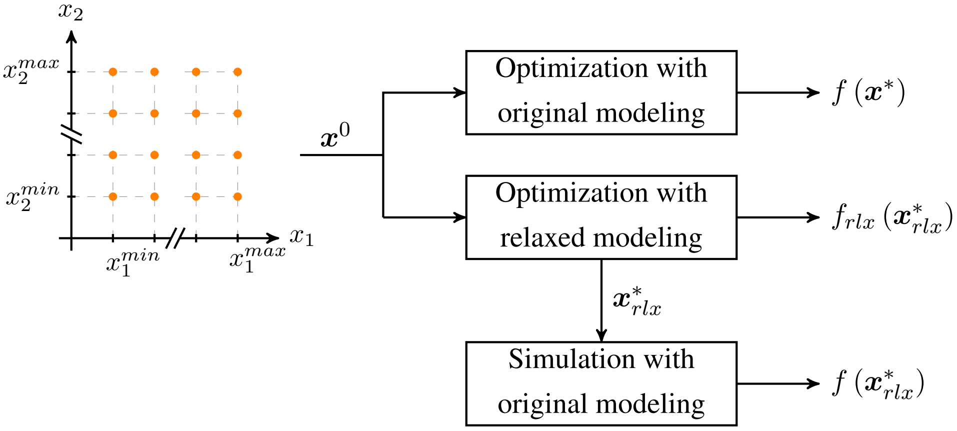
search space, as illustrated in Fig. 10, are tested. For each of these points, the optimization with the original and relaxed modeling are performed and the results are saved for a subsequent analysis. The MG Simulator is executed with the original modeling for the relaxed optimal points x^*_{rlx}, to obtain the indicators values for the original modeling and determine the errors associated with the relaxation.

The convergence error for each point i is measured with relative errors. For the optimizations performed with the original modeling, it is calculated as

$$\delta_i = \frac{f(x_i^*) - f(x^*)^{min}}{f(x^*)^{min}}, \tag{69}$$

where $f(x^*)^{min}$ is the minimum value between $f(x_1^*), \ldots, f(x_i^*), \ldots, f(x_N^*)$, and for the

Figure 10 – Procedure for optimization convergence assessment.


x_2
x_2^{max}
x_2^{min}
x_1^{min}
x_1^{max}
x_1
$\boldsymbol{x}^0$
Optimization with original modeling
$f(\boldsymbol{x}^*)$
Optimization with relaxed modeling
$f_{rlx}(\boldsymbol{x}^*_{rlx})$
$\boldsymbol{x}^*_{rlx}$
Simulation with original modeling
$f(\boldsymbol{x}^*_{rlx})$

Source: the author.

optimization with the relaxed modeling as

$$\delta_{rlx,i} = \frac{f_{rlx}(x^*_{rlx,i}) - f_{rlx}(x^*_{rlx})^{min}}{f_{rlx}(x^*_{rlx})^{min}}, \tag{70}$$

where $f_{rlx}(x^*_{rlx})^{min}$ is the minimum value between $f_{rlx}(x^*_{rlx,1}), \ldots, f_{rlx}(x^*_{rlx,i}), \ldots, f_{rlx}(x^*_{rlx,N})$.

It is not enough a better convergence of the optimization with the relaxed model, i.e. errors obtained with Eq. (70) lower than the ones obtained with Eq. (69), to prove that the relaxation improved the microgrid optimization performance. Thus, it is also calculated the relaxation error associated with the results obtained after the original modeling simulation with the relaxed optimal values. This error is calculated as

$$\delta_{ori \to rlx,i} = \frac{f(x^*_{rlx,i}) - f_{known}(x^*)^{min}}{f_{known}(x^*)^{min}}, \tag{71}$$

where $f_{known}(x^*)^{min}$ is the minimum value between $f(x^*_1), \ldots, f(x^*_i), \ldots, f(x^*_N)$ and $f(x^*_{rlx,1}), \ldots, f(x^*_{rlx,i}), \ldots, f(x^*_{rlx,N})$.

3.3 Comparison with the Particle Swarm Optimization algorithm

To evaluate the performance of the proposed methodology, it is important to compare the results with the ones from a well established optimization algorithm. As mentioned in §2.6, the one of the most used optimization algorithm for the microgrid optimal sizing problem is the PSO.

In this work, the PSO is implemented as in Algorithm 2, where the fitness value is calculated with the outputs from the MG Simulator. The fitness function is defined accordingly with the case study, i.e. it is problem dependent, because it needs to include the constraints. Thus, it is defined in the next section.

Due to PSO stochastic characteristic, it is necessary to run the optimization many times and statistically evaluate its convergence. More specifically, the PSO algorithm is run k

times. For comparison purposes, the gradient-based+AD algorithm is run for k random initial values. The performance of each algorithm is evaluated using Eqs. (16), (17), (19), (20) and (21), and the results are compared for the case studies.

4 CASE STUDIES

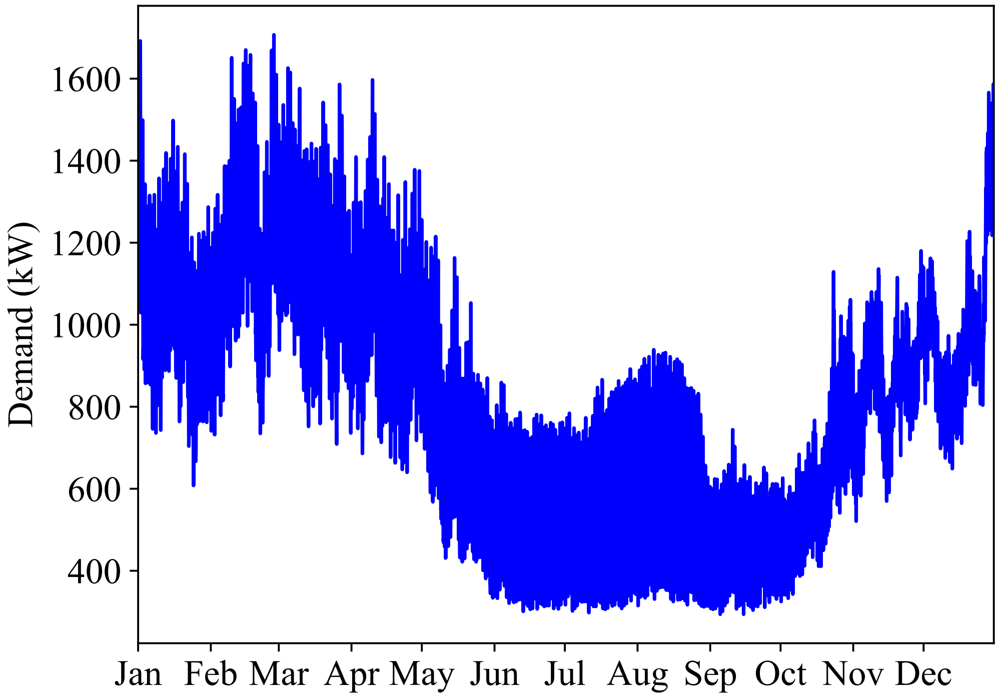
In this section, case studies are established to evaluate the performance of the proposed methodology in the optimal sizing of a islanded microgrid. The results of the convergence performance of the proposed gradient-based+AD algorithm are presented, as well as the PSO results for comparison.

4.1 Microgrid data and considerations

The hourly load and climate data used are from the Ushant Island, located in France. Although the microgrid optimization is usually performed for representative days of operation and climate, e.g. averages days from historical data or averages days for different seasons and load levels, in this work the data is purely the measured ones for year 2016. This allows a more realistic simulation of the microgrid power dispatch and operation, with high data diversity. The PV data was obtained with PVGIS-SARAH database (European Commission Joint Research Centre, 2022), for a 1 kW panel, with a loss of 14%, panel slope of 40°, and a 0° azimuth.

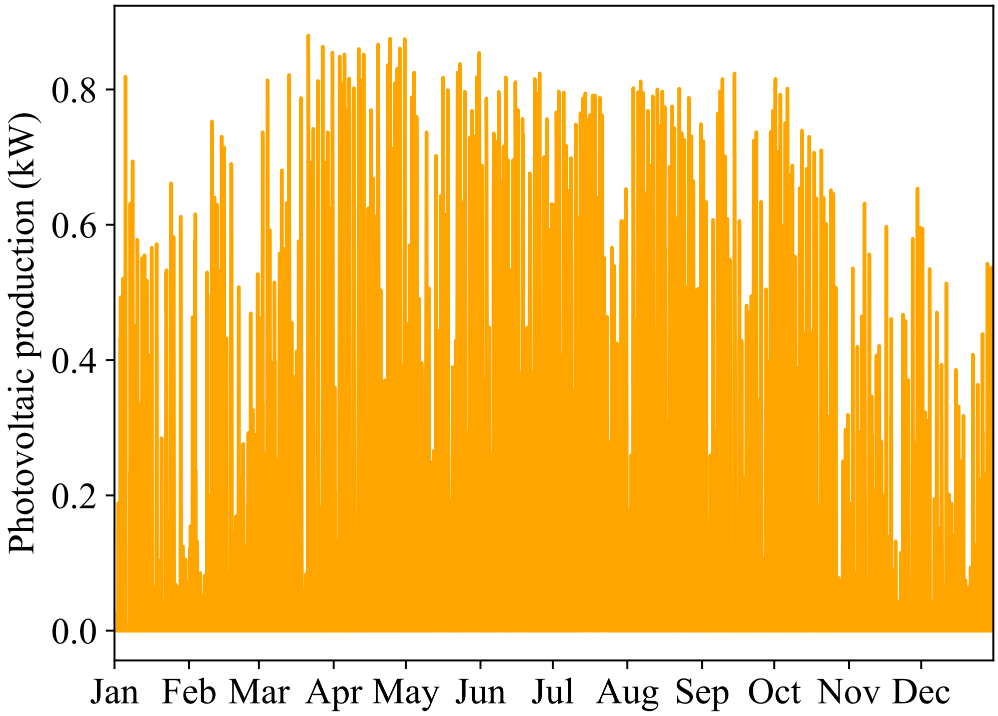
Figure 11 presents the demand curve used. The peak demand is 1707 kW and occurs in 27/02/2016 at 22h. The demand is higher in late hours and during the colder months, November to May, due to the necessity of heating. The photovoltaic production is presented in Fig. 12, where can be observed a difference in the production capacity between the summer and winter months. Differently from the photovoltaic power potential, the wind speed is lower during summer months, as can be seen in Fig. 13, indicating a decrease of wind power production during this time period. From the photovoltaic and wind speed behavior, it is expected a complementary production of this energy sources.

Figure 11 – Demand curve for the Ushant Island.



Source: the author.

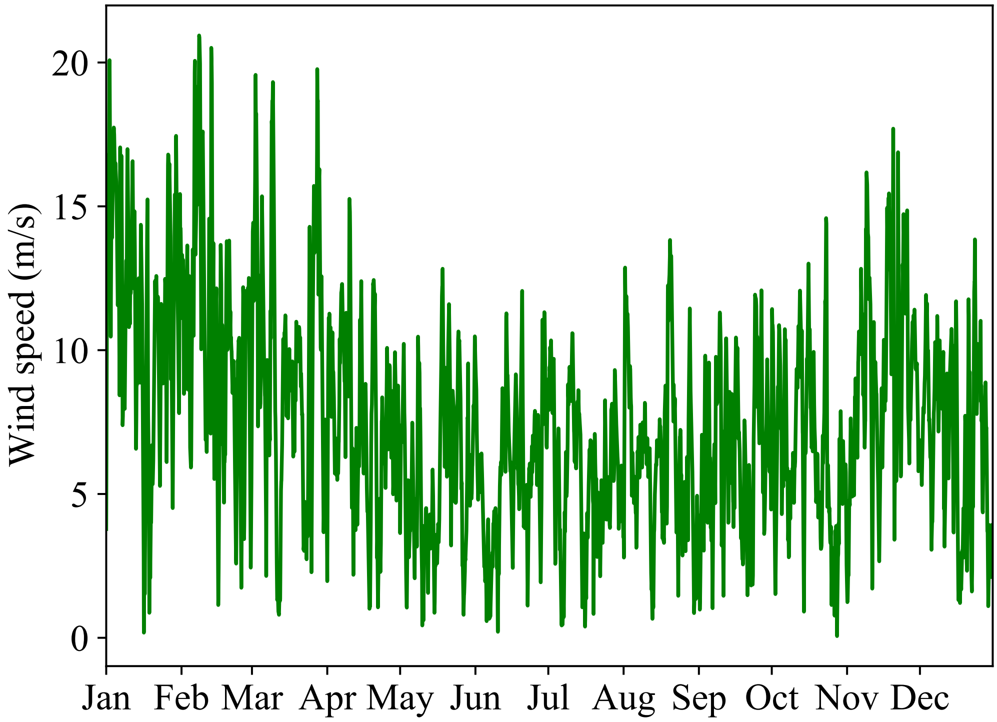
Figure 12 – Photovoltaic production of a 1 kW panel for the Ushant Island.



Source: the author.

For the power dispatch, a rule-based strategy was implemented. As microgrids with dispatchable renewable sources were not studied in this work, the following order applies

Figure 13 – Wind speed for the Ushant Island.



Source: the author.

for all scenarios:

1. Nondispatchable renewable sources

2. Batteries

3. Diesel generating set

With regard to the battery charging strategy, the load-following, presented in Fig. 6 from §2.4.1, is used. This reduces the diesel genset use, as it will only operate to supply the load when required.

The technical and economic parameters for the project and each component are presented in Table 5. As the PV data is already given in kW, with loss and temperature consideration, it is used the Eq. (23) and no additional loss are considered. The technical data for the WT are from the wind power ENERCON E-70, which is adapted for sites in coastal areas with high wind speeds (ENERCON GmbH, 2016) and already used in a case study for the Ushant Island (Mohammed *et al.*, 2016). The costs are the mean values from (Lazard, 2020a,b), excepted for the GS, that all parameters are for the medium generator model from HOMER. The components lifetimes and BT linear loss factor and maximum cycles are also from (Lazard, 2020a,b).

Table 5 – Technical and economic parameters.

	Parameters	Values
Project	Discount rate	5%
	Timestep	1 h
	Lifetime	25 years
Photovoltaic system	Derating factor	100%
	Lifetime	30 years
	Investment cost	1400.00 \$/kW
	O&M cost	14.00 $\frac{\$}{\text{kW}\cdot\text{year}}$
Wind turbine	Cut-in speed	5 m/s
	Cut-out speed	25 m/s
	Rated speed	15 m/s
	Hub height	85 m
	Measurement height	10 m
	Power law exponent	0.2
	Lifetime	20
	Investment cost	1250.00 \$/kW
	O&M cost	33.25 $\frac{\$}{\text{kW}\cdot\text{year}}$
Battery	Initial energy	0 kWh
	Minimum energy level	0 kWh
	Maximum charge power	-0.5·E_{BT}^{rtd}
	Maximum discharge power	0.5·E_{BT}^{rtd}
	Linear loss factor	6%
	Maximum cycles	5000
	Lifetime	10 years
	Investment cost	359.50 \$/kWh
	O&M cost	2.85 $\frac{\$}{\text{kW}\cdot\text{year}}$
Diesel genset	Minimum load ratio	0
	F0	0
	F1	0.240
	Lifetime	15000 h
	Investment cost	400.00 \$/kW
	O&M cost	0.02 \$/(kW·h$_{oper}$)
	Fuel cost	1.00 \$/L

4.2 Employed tools

The programming language chosen to implement all the methodology of this research was the Julia language. This allows an appropriate integration between the MG Simulator, optimization algorithms and automatic differentiation packages. The primarily advantage of this language is that it is open-source, enabling a more collaborative development of packages, the language itself, and the research reproduction or continuation. Its just-in-time (JIT) compilation makes its performance attractive when compared with another scientific programming languages as Matlab and Python (Bezanson *et al.*, 2012).

The package *ForwardDiff.jl* was selected between the Automatic Differentiation packages available for Julia. It is a consolidated one, thus, more tests were performed with it and less bugs are expected. Further, it has a better performance than the consolidated back-propagation package *ReverseDiff.jl* for few input variables (Revels; Pearson, 2016).

The *NLopt.jl* Julia module was used to perform the gradient-based optimization. It is a open-source library for nonlinear optimization, in which many algorithms are available, including gradient-free and gradient-based ones. Among its gradient-based algorithms, SLSQP is the only one whose supports directly in its formulation equality and inequality constraints (Johnson, 2007–2020).

The Microgrid Simulator was implemented in Julia following the architecture presented in §3.1 and the gradient-based+AD algorithm as in §3.2. To compare the results with the ones from a PSO algorithm, the *Metaheuristics.jl* package was used (Mejía; Satman; Monticone, 2022).

The time computation and memory assessment was made with the *BenchmarkTools.jl*, which enables to compute the time and memory considering a set number of samples. The computer used was a notebook with Intel® Core™ i7-9750H CPU and 8,00 GB RAM.

4.3 Case studies definition

The microgrid configuration can be composed of: two nondispatchable renewable sources, photovoltaic systems (PV) and wind turbines (WT); one non-renewable source, the diesel generating set (GS); and, one type of storage, the batteries (BT). All four scenarios have the PV and BT as components to be optimized and two scenarios have also the WT. All scenarios have the GS component, that has a fixed rated power in two of them and it is a variable to be optimized in the other two. To evaluate the convergence behavior of the original modeling and the relaxed one, two cases were tested: one without GS operation hours relaxation and other with GS operation hours relaxation. The case studies scenarios are defined as follows:

- Scenario 1: PV + BT + Fixed GS
- Scenario 2: PV + BT + GS

- Scenario 3: PV + WT + BT + Fixed GS
- Scenario 4: PV + WT + BT + GS

To study the influence of the relaxation factor in the convergence results, the optimization was performed with six values for each scenario: 0.01, 0.05, 0.10, 0.25, 0.50 and 0.75.

The parameters used in the PSO were taken from (Javed *et al.*, 2020). They are 100 particles, φ_1 and φ_2 equal to 2, a constant ϕ_w with the value of the maximum used by (Javed *et al.*, 2020), 100 iterations for scenarios 1 and 2, and 50 iterations for scenarios 3 and 4. The fitness function for Scenarios 1 and 3 are equal the objective functions. For scenarios 2 and 4, the constraint is handled as a penalty function, and the fitness function is defined as

$$NPC(\cdot) = NPC(\cdot) + 5 \cdot 10^7 \cdot p_{SF}, \tag{72}$$

where p_{SF} is obtained with Eq. (15).

For convergence performance comparison, the PSO algorithm was executed 30 times and the gradient-based+AD was executed for 30 random initial points. Meanwhile, for the time computation comparison, the optimization was performed 10 times for each algorithm, i.e., configuration of 10 samples in the *BenchmarkTools.jl*.

4.4 Scenario 1: PV + BT + Fixed GS

In this scenario, the optimization variables are the PV rated power P_{PV}^{rtd} and the BT rated capacity E_{BT}^{rtd}. The GS has a fixed rated power P_{GS}^{rtd} of 1800 kW, a power higher than the maximum load leading to a zero shedding fraction (SF). Therefore, the optimization problem is defined as

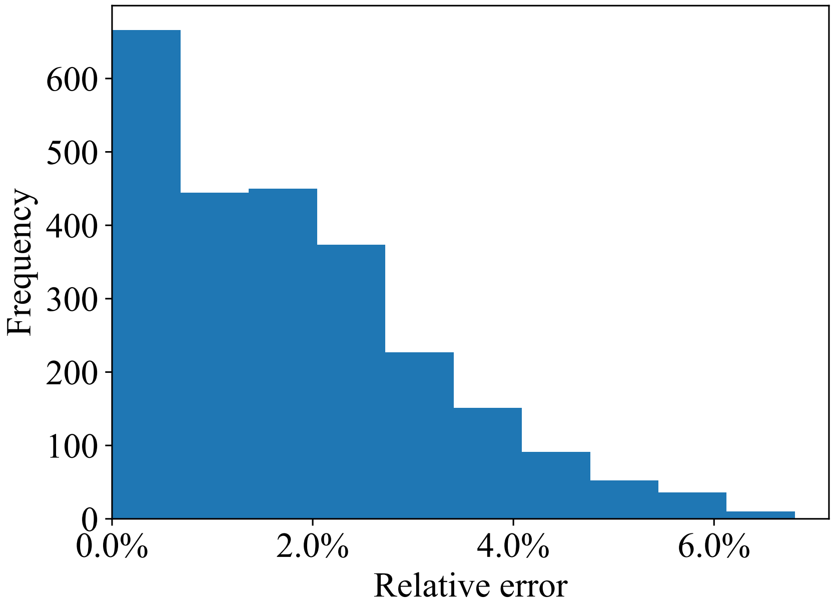
$$\begin{aligned} \min \quad & NPC\left(P_{PV}^{rtd}, E_{BT}^{rtd}\right) && \text{(73a)} \\ s.t. \quad & 0 \leq P_{PV}^{rtd} \leq 10000\, kW && \text{(73b)} \\ & 0 \leq E_{BT}^{rtd} \leq 10000\, kWh. && \text{(73c)} \end{aligned}$$

The initial points were created with a step of 200 kW for the P_{PV}^{rtd} and 200 kWh for the E_{BT}^{rtd}. This resulted in 2500 initial points, thus 2500 optimization executions with the gradient-based+AD algorithm.

4.4.1 Without GS relaxation

The relative error δ_i was computed for NPC optimal value obtained with each of the 2500 initial points. Figure 14 shows the histogram of the relative errors δ_i. It can be observed that there is a large occurrence of error between zero and 3%, with a maximum value of 6.81%. Table 6 also shows the percentiles 0.5, 0.95 and 0.99. These results support the need of using a relaxation approach to see if the algorithm convergence improves.

Figure 14 – Histogram of relative errors δ_i for Scenario 1.



Source: the author.

Table 6 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors δ_i of Scenario 1 without GS operation hours relaxation.

Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.00%	1.58%	4.53%	5.93%	6.81%

4.4.2 With GS relaxation

Table 7 shows statistical values for the $\delta_{rlx,i}$ relative errors for the six relaxation factors. For the two lowest ones, the error decreases compared with the original optimization. However, the δ_{rlx} is still elevated comparing with the others relaxation factors. On the other hand, for the two highest relaxation factors the δ_{rlx} are low, showing that the relaxation improved the algorithm convergence. Though, the $\delta_{ori \to rlx}$ errors, presented in Table 8, also need to be low. For the relaxation factors 0.50 and 0.75, the $\delta_{ori \to rlx}$ are higher in spite of lower $\delta_{rlx,i}$ errors, indicating that higher relaxation factors cause the simulation behavior to diverge from the one of the original model. The best relaxation factor for this scenario was the ε equal to 0.10, then this value was used for the PSO performance comparison. The histograms of the relative errors $\delta_{rlx,i}$ and $\delta_{ori \to rlx,i}$, for all relaxation factors studied, are presented in §B.1.

4.4.3 Comparison with PSO

Table 9 presents the performance indicators for Scenario 1. The gradient-based+AD found a NPC of \$28,333,848.48 for P_{PV}^{rtd} equal to 4520.05 kW and E_{BT}^{rtd} equal to 6553.47 kWh, and the PSO found a NPC equal to \$28,319,969.78 for P_{PV}^{rtd} equal to 4528.96 kW and E_{BT}^{rtd}

Table 7 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors $\delta_{rlx,i}$ of Scenario 1 with GS operation hours relaxation.

ε	Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.01	0.00%	0.07%	2.11%	2.45%	2.89%
0.05	0.00%	0.00%	0.04%	0.16%	2.01%
0.10	0.00%	0.00%	0.02%	0.02%	0.03%
0.25	0.00%	0.00%	0.00%	0.00%	0.00%
0.50	0.00%	0.00%	0.01%	0.01%	0.02%
0.75	0.00%	0.00%	0.00%	0.00%	0.00%

Table 8 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors $\delta_{ori \to rlx,i}$ of Scenario 1 with GS operation hours relaxation.

ε	Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.01	0.00%	0.09%	2.14%	2.47%	2.91%
0.05	0.01%	0.04%	0.08%	0.21%	2.11%
0.10	0.04%	0.06%	0.09%	0.09%	0.10%
0.25	0.21%	0.23%	0.24%	0.24%	0.25%
0.50	1.52%	1.98%	1.99%	2.00%	2.01%
0.75	3.93%	3.95%	3.96%	3.97%	3.99%

equal to 6651.19 kWh. All indicators had best values for the PSO algorithm, because it presented the best objective function value. The SD shows the dispersion around the mean value, indicating that the optimal values obtained with the PSO algorithm are closer to the mean value than the ones obtained with the gradient-based+AD. The performance indicators MAE, RLE and RMSE, are related with the dispersion around the minimum optimal value obtained. Comparing these indicators for the NPC_{best}^{GAD} and NPC_{best}^{PSO}, the PSO algorithm give results 5 times nearer the minimum value obtained than the GB+AD. However, when calculating these indicators with the minimum optimal value obtained between the two algorithms, i.e., the minimum value obtained with PSO, the difference in the dispersion is of almost 18 times. Nevertheless, the mean efficiency result of the proposed algorithm is close to the PSO one, indicating that its performance is almost as good as the already established PSO algorithm. Moreover, the difference between the optimal sizes is not so high if it is considered that the components available in the market have discrete values.

The time results are summarized in Table 10. The proposed algorithm is at least 43 times faster than the PSO. However, using the median values that have less influence from the outliers, the speed difference is of almost 300 times.

Table 9 – Scenario 1 performance indicators for gradient-based+AD and PSO algorithms.

Indicator	Gradient-based+AD		PSO	
	NPC_{best}^{GAD}	NPC_{best}^{both}	NPC_{best}^{PSO}	NPC_{best}^{both}
SD ($)	3708.84	-	672.99	-
MAE ($)	5700.28	19578.98	1193.07	1193.07
MRE (%)	0.020	0.069	0.004	0.004
RMSE ($)	6766.84	19915.65	1364.27	1364.27
$\eta_{mean}(\%)$	99.990	99.965	99.998	99.998

Table 10 – Time execution for gradient-based+AD and PSO algorithms.

Time	Gradient-based+AD	PSO
Minimum	486.586 ms	142.469 s
Median	502.087 ms	149.478 s
Mean	844.279 ms	149.286 s
Maximum	3.527 s	153.408 s

4.5 Scenario 2: PV + BT + GS

In addition to variables optimized in Scenario 1, the GS rated power P_{GS}^{rtd} is also optimized. In this case, the shedding fraction (SF) is equal to 100% if a constraint is not added, because the trivial optimal values for the components size, P_{PV}^{rtd}, E_{BT}^{rtd} and P_{GS}^{rtd}, are equal to zero. Therefore, a constraint function for the SF is included, resulting in an optimization problem defined as

$$\min \quad NPC\left(P_{PV}^{rtd}, E_{BT}^{rtd}, P_{GS}^{rtd}\right) \tag{74a}$$

$$s.t. \quad SF \leq 0.10\% \tag{74b}$$

$$0 \leq P_{PV}^{rtd} \leq 10000\,kW \tag{74c}$$

$$0 \leq E_{BT}^{rtd} \leq 10000\,kWh \tag{74d}$$

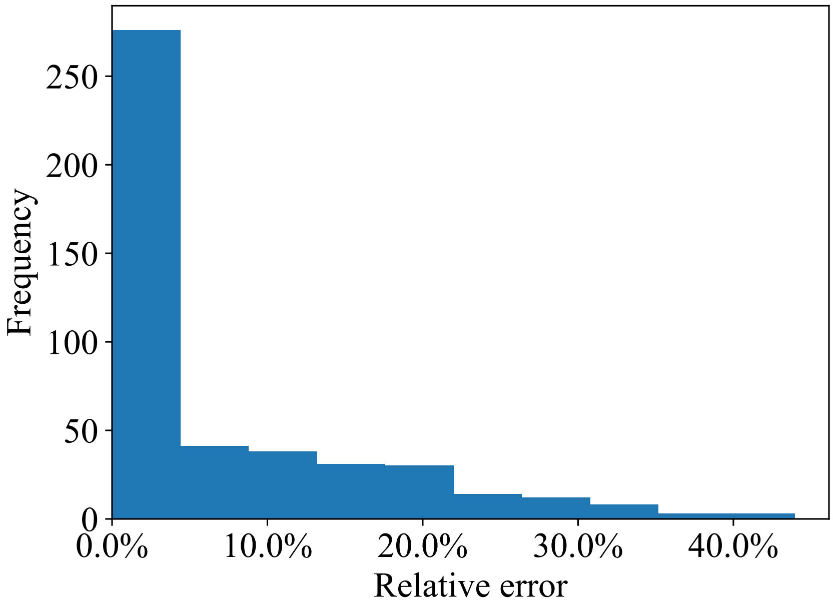
$$0 \leq P_{GS}^{rtd} \leq 2000\,kW. \tag{74e}$$

The initial points were created with a step of 1000 kW for the P_{PV}^{rtd}, 1000 kWh for the E_{BT}^{rtd} and 200 kW for the P_{GS}^{rtd}. This resulted in 1000 initial points to be tested. In addition to the characteristics already analyzed for Scenario 1, it is also verified whether the constraint is satisfied or not. It is necessary because round-off errors allow SLSQP algorithm to converge even if constraints are slightly disrespected. For this, it was considered that 0.101% is a tolerable value for the maximum shedding fraction.

4.5.1 Without GS relaxation

For this scenario, 54.4% of the initial points converged to optimal points that did not respected the constraint. Moreover, the achieved δ_i relative errors are unacceptable for an optimization algorithm. The histogram for the remaining values is presented in Fig. 15, where can be seen a high frequency between 0 and 5% of relative error. Observing Table 11, 50% of the initial points lead to errors up to 4.29%, but for the other half the errors can reach up to 37.10%, considering the percentile 0.99.

Figure 15 – Histogram of relative errors δ_i for Scenario 2.



Source: the author.

Table 11 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors δ_i of Scenario 2 without GS operation hours relaxation.

Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.00%	4.29%	26.92%	37.10%	43.96%

4.5.2 With GS relaxation

The first step was to eliminate the results that did not respected the constraint. This lead to the rejections rate of 2.9%, 0.4% and 0.2% for the relaxation factors equal to 0.01, 0.05 and 0.10, respectively. There was no rejection for the other relaxation factors. Therefore, the GS operation hours relaxation helped with the fulfillment of the constraint.

In Table 12, it can be observed that for half of the relaxation factors the maximum error have a considerable magnitude, but the expected behavior of decreasing for higher ε values was preserved. Table 13 presents the relative errors after reinserting the optimal values in the original model. Differently from Scenario 1, the relaxation factor that presents the lowest error is ε equal to 0.25, thus it was used for PSO performance comparison.

The histograms of the relative errors $\delta_{rlx,i}$ and $\delta_{ori \rightarrow rlx,i}$, for all relaxation factors studied, are presented in §B.2.

Table 12 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors $\delta_{rlx,i}$ of Scenario 2 with GS operation hours relaxation.

ε	Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.01	0.00%	0.19%	1.17%	5.41%	30.64%
0.05	0.00%	0.01%	0.92%	1.12%	13.67%
0.10	0.00%	0.00%	0.91%	0.91%	7.62%
0.25	0.00%	0.00%	0.00%	0.00%	0.77%
0.50	0.00%	0.00%	0.01%	0.01%	0.15%
0.75	0.00%	0.00%	0.00%	0.00%	0.00%

Table 13 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors $\delta_{ori \rightarrow rlx,i}$ of Scenario 2 with GS operation hours relaxation.

ε	Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.01	0.00%	0.20%	1.18%	5.43%	30.61%
0.05	0.00%	0.04%	1.00%	1.18%	13.61%
0.10	0.00%	0.05%	1.01%	1.02%	7.59%
0.25	0.00%	0.03%	0.05%	0.05%	1.00%
0.50	0.47%	0.93%	0.94%	0.94%	1.20%
0.75	1.70%	1.81%	1.81%	1.82%	1.85%

4.5.3 Comparison with PSO

The 30 optimal results obtained with the gradient-based+AD satisfied the constraint, and the best NPC value found was \$27,236,455.38 for P_{PV}^{rtd} equal to 4289.92 kW, E_{BT}^{rtd} equal to 6036.31 kWh and P_{GS}^{rtd} equal to 1439.95 kW. The PSO found a NPC equal to \$27,213,977.84 for P_{PV}^{rtd} equal to 4386.42 kW, E_{BT}^{rtd} equal to 6317.49 kWh and P_{GS}^{rtd} equal to 1439.02 kW. Table 14 presents the performance indicators for Scenario 2. The SD is lower for the gradient-based+AD, indicating that the range of solutions was less dispersed around the mean when using this algorithm. The performance indicators MAE, MRE, RMSE and η_{mean}, have the same order of magnitude when analyzing the results of NPC_{best}^{GAD} and NPC_{best}^{PSO} columns from Table 14. However, when comparing the results from the columns NPC_{best}^{both}, the PSO presented lower values for the MAE, MRE and η_{mean} performance indicators. That occurred because the minimum value was obtained with the PSO algorithm and the dispersion from the best individual optimal is closer for both algorithms.

Table 14 – Scenario 2 performance indicators for gradient-based+AD and PSO algorithms.

Indicator	Gradient-based+AD		PSO	
	NPC_{best}^{GAD}	NPC_{best}^{both}	NPC_{best}^{PSO}	NPC_{best}^{both}
SD ($)	2213.30	-	3501.20	-
MAE ($)	4103.63	26581.17	4028.55	4028.55
MRE (%)	0.015	0.098	0.015	0.015
RMSE ($)	4644.90	26670.09	5298.96	5298.96
$\eta_{mean}(\%)$	99.992	99.951	99.993	99.993

The processing times are summarized in Table 15. Comparing with Scenario 1, the addition of one variable almost doubled the computation time. The proposed algorithm is at least 57 times faster than the PSO, but using the median values the difference is bigger than 340 times.

Table 15 – Time execution for gradient-based+AD and PSO algorithms.

Time	Gradient-based+AD	PSO
Minimum	858.372 ms	290.039 s
Median	869.996 ms	297.087 s
Mean	1.394 s	297.163 s
Maximum	5.313 s	303.236 s

4.6 Scenario 3: PV + WT + BT + Fixed GS

In this scenario, the optimization variables are the PV rated power P_{PV}^{rtd}, WT rated power P_{PV}^{rtd}, and the BT rated capacity E_{BT}^{rtd}. As in Scenario 1, the GS has a fixed rated power P_{GS}^{rtd} of 1800 kW. Therefore, the optimization problem does not have a constraint for SF, and it is defined as

$$\min \quad NPC\left(P_{PV}^{rtd}, P_{WT}^{rtd}, E_{BT}^{rtd}\right) \tag{75a}$$

$$s.t. \quad 0 \leq P_{PV}^{rtd} \leq 10000\, kW \tag{75b}$$

$$0 \leq P_{WT}^{rtd} \leq 10000\, kW \tag{75c}$$

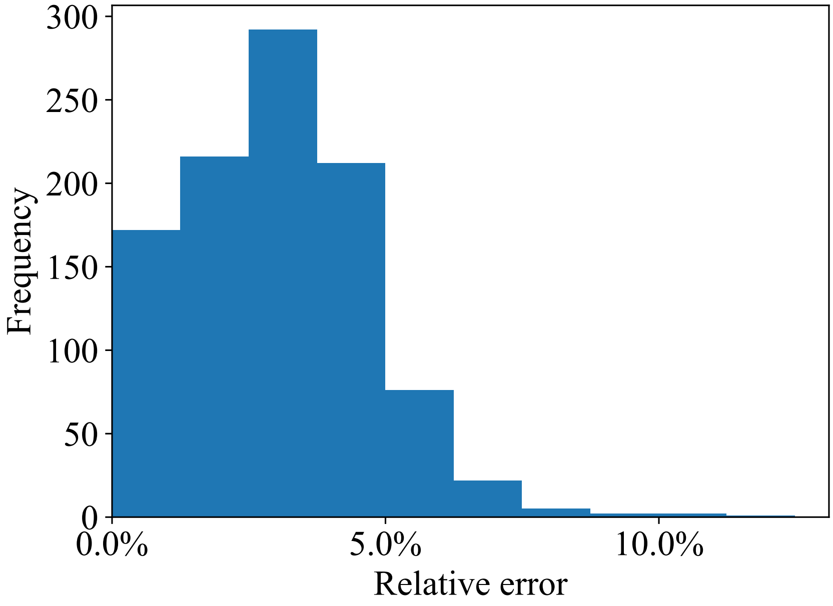
$$0 \leq E_{BT}^{rtd} \leq 10000\, kWh. \tag{75d}$$

The initial points were created with a step of 1000 kW for the P_{PV}^{rtd} and P_{WT}^{rtd}, and 1000 kWh for the E_{BT}^{rtd}, resulting in 1000 initial points.

4.6.1 Without GS relaxation

The histogram for the δ_i errors for the optimization without GS operation hours relaxation is presented in Fig. 16. The histogram form differ from Scenario 1, the other one without GS Power Rated optimization, with the highest frequency happening around 3% and not 0%. Table 16 presents some statistical results for these errors. The maximum value was 12.49%, higher than the one for Scenario 1. However, the percentile 0.99 indicates that only 10 initial points converged to optimal values with a error higher than 7.41%.

Figure 16 – Histogram of relative errors δ_i for Scenario 3.



Source: the author.

Table 16 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors δ_i of Scenario 3 without GS operation hours relaxation.

Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.00%	3.01%	5.81%	7.41%	12.49%

4.6.2 With GS relaxation

For the optimization with MG model with GS operation hours relaxation, the results of $\delta_{rlx,i}$ and $\delta_{rlx \to ori,i}$ are presented in Table 17 and 18, respectively. As expected for the $\delta_{rlx,i}$ relative errors, its values decrease for higher relaxation factors values. The $\delta_{rlx,i}$ behavior was the same of the previous scenarios, higher values for the extreme tested relaxation factors. As in the Scenario 1, the lowest errors were obtained for a relaxation factor ε equal to 0.10, and it was used for the comparison performance with PSO. The histograms of the relative errors $\delta_{rlx,i}$ and $\delta_{ori \to rlx,i}$, for all relaxation factors studied, are presented in §B.3.

Table 17 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors $\delta_{rlx,i}$ of Scenario 3 with GS operation hours relaxation.

ε	Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.01	0.00%	0.16%	1.46%	2.38%	4.73%
0.05	0.00%	0.01%	0.17%	0.18%	0.21%
0.10	0.00%	0.00%	0.02%	0.02%	0.07%
0.25	0.00%	0.00%	0.00%	0.00%	0.00%
0.50	0.00%	0.00%	0.00%	0.00%	0.00%
0.75	0.00%	0.00%	0.00%	0.00%	0.00%

Table 18 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors $\delta_{ori \to rlx,i}$ of Scenario 3 with GS operation hours relaxation.

ε	Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.01	0.00%	0.18%	1.48%	2.43%	4.71%
0.05	0.01%	0.07%	0.19%	0.28%	0.30%
0.10	0.01%	0.05%	0.09%	0.10%	0.15%
0.25	0.21%	0.25%	0.27%	0.27%	0.30%
0.50	1.62%	1.73%	1.76%	1.77%	1.79%
0.75	3.82%	3.89%	3.92%	3.96%	3.98%

4.6.3 Comparison with PSO

Table 19 presents the performance indicators for Scenario 3. The gradient-based+AD found a NPC of \$15,852,508.66 for P_{PV}^{rtd} equal to 1456.26 kW, P_{WT}^{rtd} equal to 1847.91 kW and E_{BT}^{rtd} equal to 1980.42 kWh, and the PSO found a NPC equal to \$15,850,249.73 for P_{PV}^{rtd} equal to 1514.87 kW, P_{WT}^{rtd} equal to 1835.88 kW and E_{BT}^{rtd} equal to 2188.71 kWh. Just like in the previous scenarios, the PSO algorithm found the best objective function value and, as in Scenario 1, all indicators had best values for the PSO. That indicates the results obtained with the gradient-based+AD algorithm are more disperse around the mean and the best value than the results obtained with the PSO. However, the difference between the indicators decreased from 5 and 18 times, achieved in Scenario 1, to approximately 3 times in this scenario. Moreover, the mean efficiency η_{mean} result of the proposed algorithm is still close to the PSO one.

The processing times are summarized in Table 20. The proposed algorithm is at least 13 times faster than the PSO. However, the difference is of almost 50 times, if it is considered the median time. It is worth remembering that the maximum number of iterations was reduced from 100 to 50 for this scenario and the next. This caused a lower difference between the proposed algorithm and PSO, both in terms of convergence performance and

Table 19 – Scenario 3 performance indicators for gradient-based+AD and PSO algorithms.

Indicator	Gradient-based+AD		PSO	
	NPC_{best}^{GAD}	NPC_{best}^{both}	NPC_{best}^{PSO}	NPC_{best}^{both}
SD (\$)	3896.52	-	1511.82	-
MAE (\$)	7096.68	9355.61	3472.79	3472.79
MRE (%)	0.045	0.059	0.022	0.022
RMSE (\$)	8064.72	10109.61	3777.52	3777.52
$\eta_{mean}(\%)$	99.978	99.971	99.989	99.9890

time computation.

Table 20 – Time execution for gradient-based+AD and PSO algorithms.

Time	Gradient-based+AD	PSO
Minimum	1.661 s	80.337 s
Median	1.684 s	83.984 s
Mean	2.163 s	83.921 s
Maximum	6.496 s	86.452 s

4.7 Scenario 4: PV + WT + BT + GS

This last scenario optimizes the size of all modeled components, i.e. PV, WT, BT and GS. As in Scenario 2, it was included a constraint for the shedding fraction (SF), resulting in an optimization problem defined as

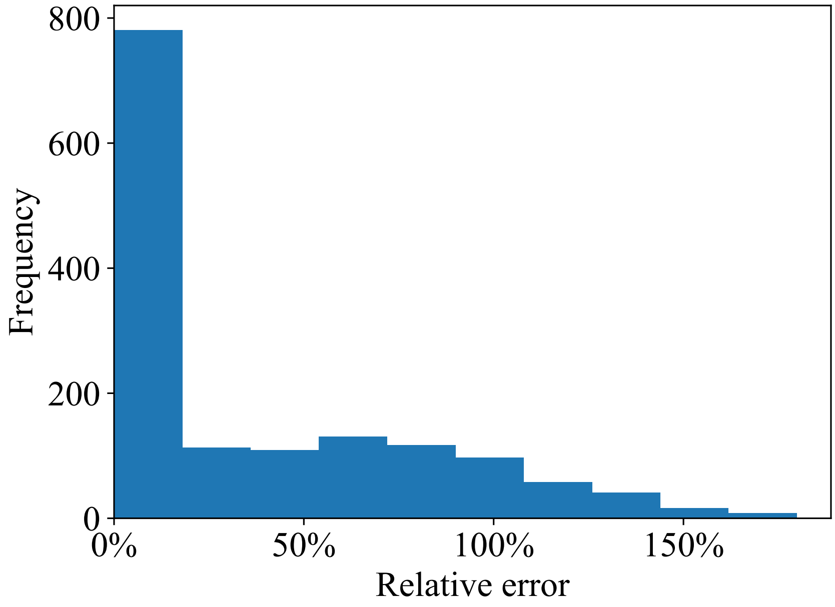
$$\begin{aligned}
\min \quad & NPC\left(P_{PV}^{rtd}, P_{WT}^{rtd}, E_{BT}^{rtd}, P_{GS}^{rtd}\right) && \text{(76a)}\\
s.t. \quad & SF \leq 0.10\% && \text{(76b)}\\
& 0 \leq P_{PV}^{rtd} \leq 10000\, kW && \text{(76c)}\\
& 0 \leq P_{WT}^{rtd} \leq 10000\, kW && \text{(76d)}\\
& 0 \leq E_{BT}^{rtd} \leq 10000\, kWh && \text{(76e)}\\
& 0 \leq P_{GS}^{rtd} \leq 2000\, kW. && \text{(76f)}
\end{aligned}$$

The initial points were created with a step of 1250 kW for the P_{PV}^{rtd} and P_{WT}^{rtd}, 1250 kWh for the E_{BT}^{rtd} and 500 kW for the P_{GS}^{rtd}, resulting in 2048 initial points.

4.7.1 Without GS relaxation

For this scenario, 28.2% of the initial points converged to optimal points that did not respected the constraint, a lower percentage compared to the Scenario 2. However, the relative errors are even higher than for Scenario 2, which can be seen in the Fig. 17 and Table 21. Even for percentile 0.50 the errors reach up to 14.51% and the maximum value is 179.87%.

Figure 17 – Histogram of relative errors δ_i for Scenario 4.



Source: the author.

Table 21 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors δ_i of Scenario 4 without GS operation hours relaxation.

Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.00%	14.51%	122.81%	151.49%	179.87%

4.7.2 With GS relaxation

With GS operation hours relaxation, the percentages of results rejection were 0.6% and 0.05% for the relaxation factors 0.01 and 0.05, respectively. The relative errors $\delta_{rlx,i}$ presented a similar behavior with the others scenarios, apart from the relaxation factor ε equal to 0.75. This can be observed in Table 22.

The relative errors $\delta_{ori \to rlx,i}$ are presented in Table 23. For this scenario there was not a relaxation factor with the lowest value for all statistical indicators considered. The relaxation factor equal to 0.25 presented the lowest values for the minimum and percentiles 0.5, 0.95 and 0.99, and the ε equal to 0.50 presented the lowest maximum value. As the percentile 0.99 represents a major part of the potential errors, the relaxation factor ε equal to 0.25 was chosen to be used in the PSO comparison.

The histograms of the relative errors $\delta_{rlx,i}$ and $\delta_{ori\rightarrow rlx,i}$, for all relaxation factors studied, are presented in §B.4.

Table 22 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors $\delta_{rlx,i}$ of Scenario 4 with GS operation hours relaxation.

ε	Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.01	0.00%	0.12%	0.62%	3.67%	149.95%
0.05	0.00%	0.01%	0.10%	0.15%	142.01%
0.10	0.00%	0.00%	0.01%	0.07%	80.43%
0.25	0.00%	0.00%	0.00%	0.01%	3.80%
0.50	0.00%	0.00%	0.00%	0.00%	0.01%
0.75	0.00%	0.00%	0.00%	0.00%	114.95%

Table 23 – Minimum, maximum e percentiles 0.5, 0.95 and 0.99, for relative errors $\delta_{ori\rightarrow rlx,i}$ of Scenario 4 with GS operation hours relaxation.

ε	Minimum	Q(0.5)	Q(0.95)	Q(0.99)	Maximum
0.01	0.00%	0.13%	0.63%	3.70%	149.87%
0.05	0.00%	0.08%	0.13%	0.20%	141.22%
0.10	0.00%	0.06%	0.11%	0.12%	79.32%
0.25	0.00%	0.06%	0.08%	0.11%	3.06%
0.50	0.63%	0.79%	0.80%	0.80%	0.90%
0.75	1.56%	1.76%	1.80%	1.84%	97.85%

4.7.3 Comparison with PSO

The convergence performance indicators are presented in Table 24. The gradient-based+AD found a NPC of \$14,894,669.36 for P_{PV}^{rtd} equal to 1226.17 kW, P_{WT}^{rtd} equal to 1831.37 kW, E_{BT}^{rtd} equal to 1181.60 kWh and P_{GS}^{rtd} equal to 1330.54 kW, and the PSO found a NPC equal to \$14,894,193.34 for P_{PV}^{rtd} equal to 1352.20 kW, P_{WT}^{rtd} equal to 1807.44 kW, E_{BT}^{rtd} equal to 1657.58 kWh and P_{GS}^{rtd} equal to 1334.89 kW. Again, the PSO found the best objective function value. Nevertheless, the best indicator values were obtained for the proposed algorithm. This is due to the closeness of the best objective function values and less dispersed results for the proposed algorithm. In other words, the PSO found the lower NPC, but for most of the 30 executions, the values found were further away the optimum than the ones obtained with the gradient-based+AD algorithm.

The time results are summarized in Table 25. The proposed algorithm is at least 43 times faster than the PSO. However, using the median values the computation time difference can reach almost 78 times.

Table 24 – Scenario 4 performance indicators for gradient-based+AD and PSO algorithms.

Indicator	Gradient-based+AD		PSO	
	NPC_{best}^{GAD}	NPC_{best}^{both}	NPC_{best}^{PSO}	NPC_{best}^{both}
SD (\$)	4011.89	-	77163.94	-
MAE (\$)	5368.21	14575.90	29627.65	29627.65
MRE	0.036	0.098	0.199	0.199
RMSE (\$)	6661.56	15100.19	81446.89	81446.89
$\eta_{mean}(\%)$	99.982	99.951	99.902	99.902

Table 25 – Time execution for gradient-based+AD and PSO algorithms.

Time	Gradient-based+AD	PSO
Minimum	2.201 s	171.679 s
Median	2.239 s	174.567 s
Mean	2.815 s	175.773 s
Maximum	6.888 s	182.856 s

4.8 Microgrid's optimal sizing and operation analysis

The optimal results are presented in Table 26. Recalling, in scenarios 1 and 3 there is an diesel generating set with a fixed P_{GS}^{rtd} equal to 1800 kW. It can be observed that the biggest absolute differences in the optimization variables, obtained with the gradient-based+AD (GB+AD) and PSO, happened for the P_{PV}^{rtd} and E_{BT}^{rtd}. This suggests that a relaxation in the PV and BT models may be necessary to reduce that difference. As mentioned before, the differences between the objective values are not so significant when thinking in terms of components available in the market. This can be also concluded from the observation in Table 27 of the levelized cost of energy (COE), for which a deviation between the indicator values is presented only for Scenario 2 and is of 0.4$\%$. Some energetic indicators are also presented in Table 27. They are the renewable fraction (RF), shedding factor (SF), maximum load shedding power P_{shed}^{max} and the maximum consecutive duration of load shedding T_{shed}^{max}.

If Scenarios 1 and 2 (MG with PV, BT and GS) are compared with Scenarios 3 and 4 (MG with PV, WT, BT and GS), the NPC, COE and RF are very different. For the scenarios with the addition of WT as renewable source, the values of the economic indicators, NPC and COE, are reduced by almost half, and the value of RF had an absolute increase around of 25$\%$. Therefore, it is important to choose the resources that are appropriate for the location's climate, both in terms of economic and environmental efficacy.

Table 26 – Comparison of the obtained optimal results.

Scenario	Algorithm	P_{PV}^{ini}	P_{WT}^{ini}	E_{BT}^{ini}	P_{GS}^{ini}	P_{PV}^{opt}	P_{WT}^{opt}	E_{BT}^{opt}	P_{GS}^{opt}	NPC
Scenario 1	GB+AD	3531	-	6549	-	4520	-	6553	-	$ 28,333,848.48
	PSO	-	-	-	-	4529	-	6651	-	$ 28,319,969.78
Scenario 2	GB+AD	1357	-	7913	1193	4290	-	6036	1440	$ 27,236,455.38
	PSO	-	-	-	-	4386	-	6317	1439	$ 27,213,977.84
Scenario 3	GB+AD	1773	7228	8423	-	1456	1848	1980	-	$ 15,852,508.66
	PSO	-	-	-	-	1515	1836	2189	-	$ 15,850,249.73
Scenario 4	GB+AD	2035	7671	9389	432	1226	1831	1182	1331	$ 14,894,669.36
	PSO	-	-	-	-	1352	1807	1658	1335	$ 14,894,193.34

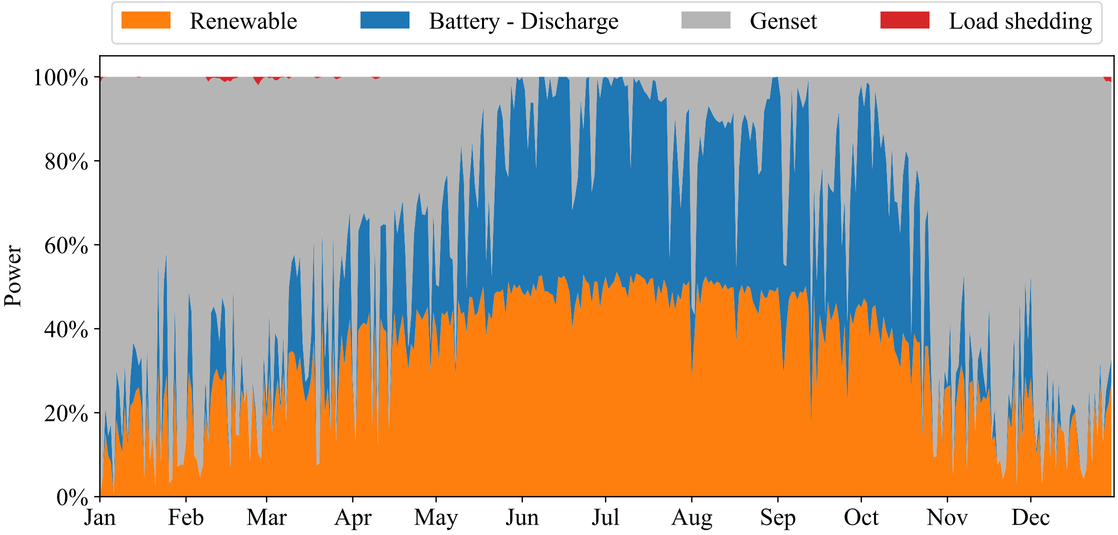
Table 27 – Comparison of economic and energetic indicators obtained with the optimal results.

Scenario	Algorithm	COE	RF	SF	P_{shed}^{max}	T_{shed}^{max}
Scenario 1	GB+AD	0.297 $/kWh	50.00%	0.00%	0 kW	0 h
	PSO	0.297 $/kWh	50.19%	0.00%	0 kW	0 h
Scenario 2	GB+AD	0.286 $/kWh	48.09%	0.10%	267 kW	6 h
	PSO	0.285 $/kWh	49.04%	0.10%	268 kW	6 h
Scenario 3	GB+AD	0.166 $/kWh	77.57%	0.00%	0 kW	0 h
	PSO	0.166 $/kWh	78.12%	0.00%	0 kW	0 h
Scenario 4	GB+AD	0.156 $/kWh	75.07%	0.10%	361 kW	9 h
	PSO	0.156 $/kWh	76.41%	0.10%	357 kW	9 h

The electrical generation mixes for Scenario 2 and 4 are showed in Figures 18 and 19, respectively. The demand supplied by PV and WT sources are represented by their sum, *Renewable* in the figures. It is important to remember that the BT is charged only by renewable sources. Moreover, the load shedding P_{shed} is also presented to show when the load was unmet by the system. The hourly simulation results were aggregated for each day to reduce the data noise and improve the visualization of the mixes of the entire year.

It can be observed in Figure 18 that the GS is responsible for the major part of demand supply in the colder months, i.e., November to May. During the other months, PV production increases and the demand is supplied mainly by it and the energy storage in the BT. There are some days where the demand is not completely met and the duration of these load shedding can reach until 6 continuous hours (see Table 27).

Figure 18 – Daily electrical generation mix for Scenario 2.

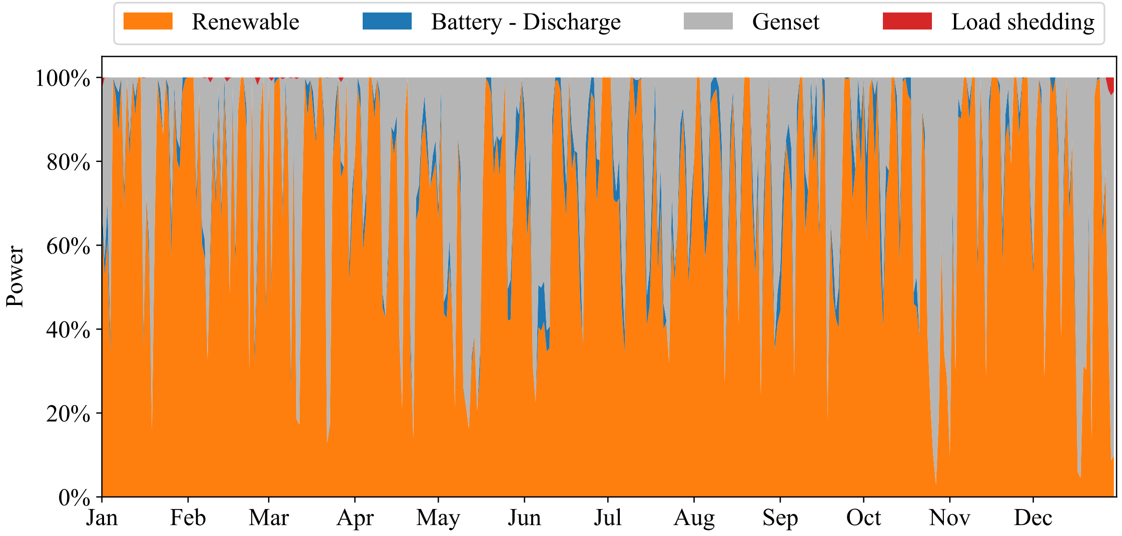


Source: the author.

For Scenario 4, presented in Figure 19, the electrical mix is different from the one presented for Scenario 2. The generation pattern is similar for all months, with a high participation of the renewable sources (PV and WT), with the exception of some days that the climate conditions were not favorable. The consecutive duration of load shedding is 9 hours and absolute value of P_{shed} is lower between February and April than for Scenario 2, but higher in late December.

Another difference from the previous scenario is in the battery use. As the PV is the only renewable source in Scenario 2 and its production is not simultaneous with the peak demand, the power and capacity of PV and BT, respectively, are almost 3 times greater than the ones of Scenario 4 results (see Table 26). The addition of WT in Scenario 4 reduces the need of PV and BT, because its production is more uniform over the year.

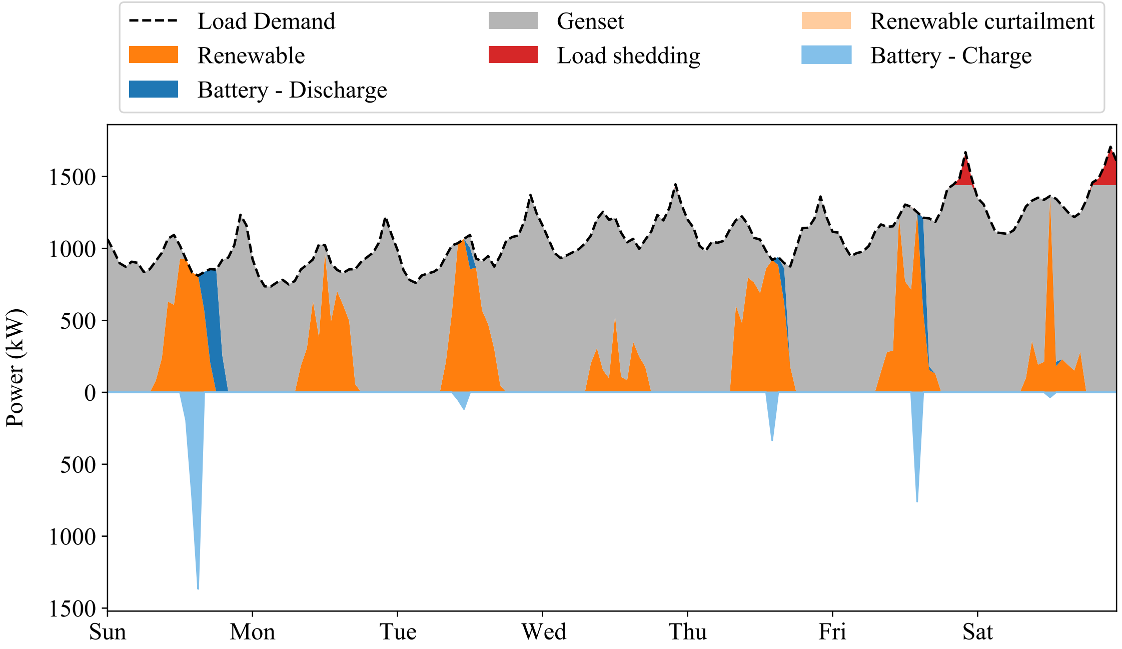
Figure 19 – Daily electrical generation mix for Scenario 4.



Source: the author.

In addition of the power presented in Figures 18 and 19, the following figures also show the hourly renewable curtailment P_{curt} and the battery charging for some chosen days. Figures 20 and 21 present the operation for the week from 21st February to 27th February. For Scenario 2, it is possible to observe that the PV production is low for most days, with a power surplus only in a few days. This surplus is entirely used to recharge the battery, which is discharged as soon as the PV stops to produce to avoid the GS operation. Also, there is no renewable curtailment and the load shedding occurs only in two days.

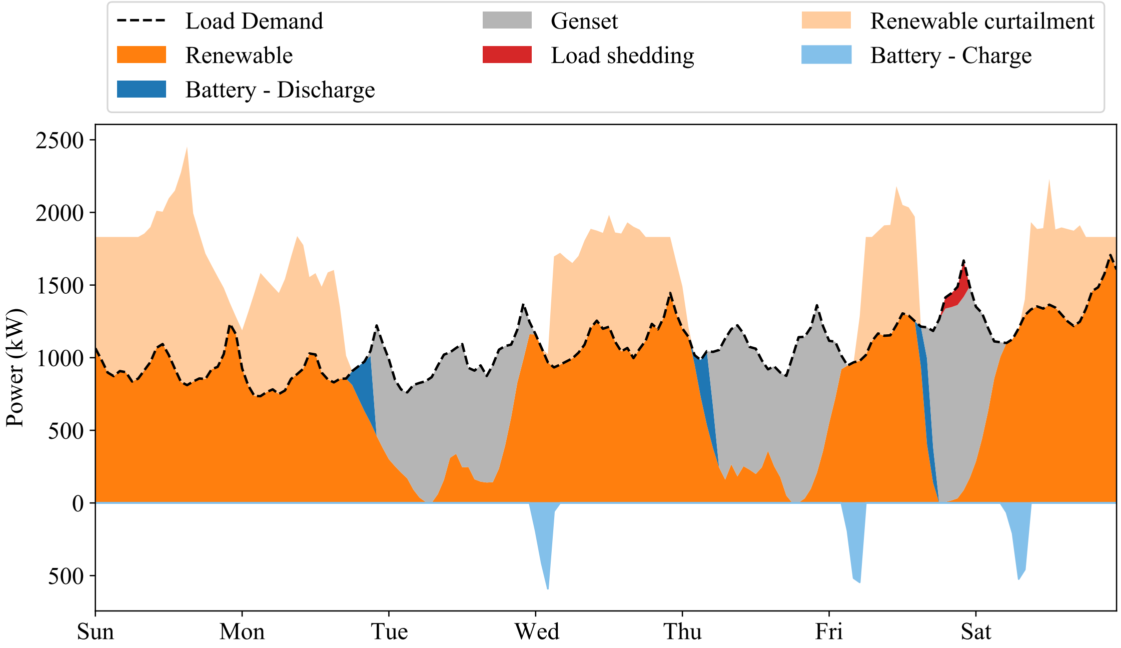
Figure 20 – Hourly dispatch for week from 21st February to 27th February for Scenario 2.



Source: the author.

With Scenario 4 simulation results for this same week, the renewable production is higher, then the renewable curtailment is present. The battery participation is not so significant because it can store less energy and deliver less power than the one sized in Scenario 2. Despite these difference, during only one day the load was unmet with the Scenario 4 configuration.

Figure 21 – Hourly dispatch for week from 21st February to 27th February for Scenario 4.

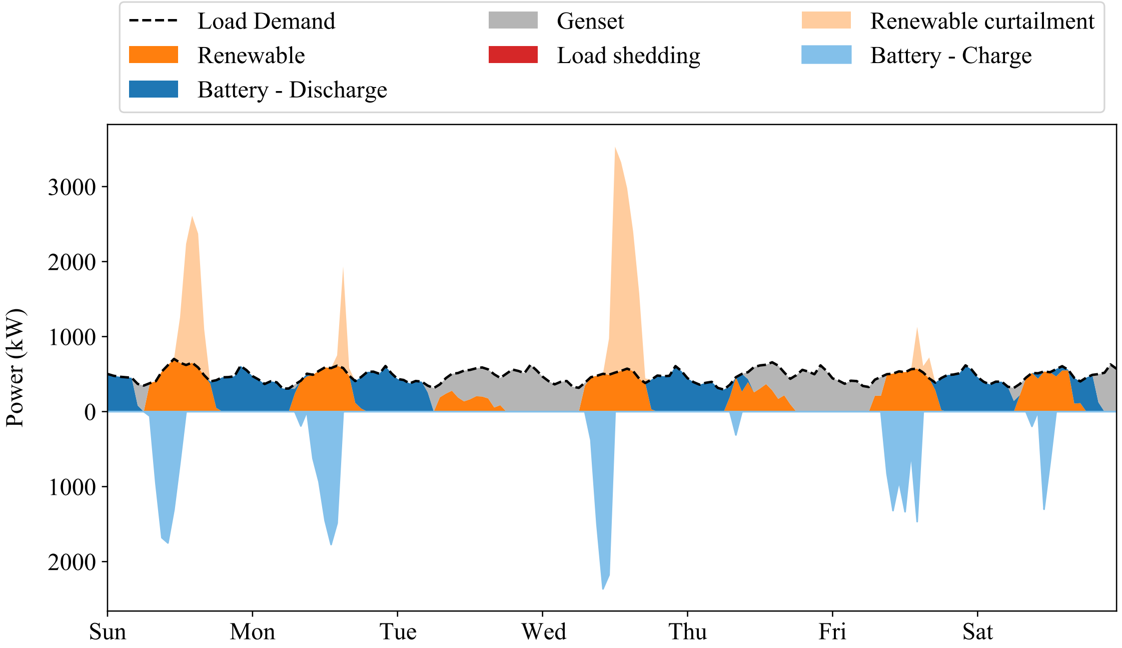


Source: the author.

The simulation results for a week during summer (11th to 17th September) are also presented, in Figures 22 and 23, to show the seasonal effects on the microgrid operation. The first difference is that there are not load shedding in none of the presented scenarios and the second is that the renewable curtailment is also present in Scenario 2. In Figure 22, the microgrid operation is presented for Scenario 2. There are four days in which the BT can supply almost the entire demand from the night beginning to the early morning.

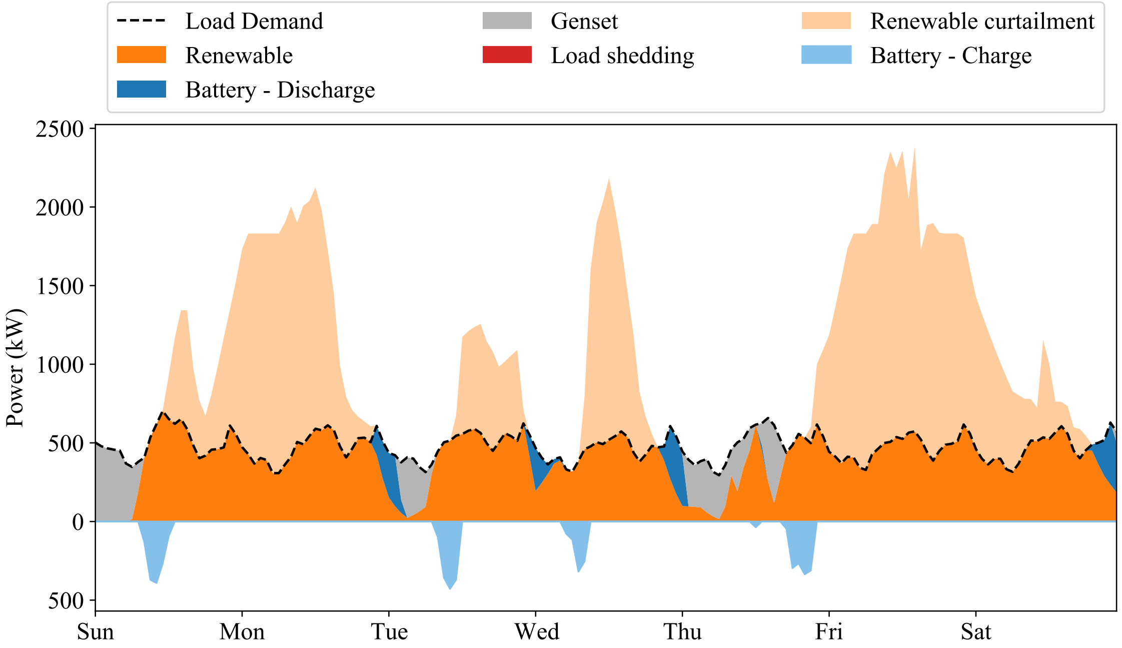
During this week for Scenario 4,the renewable curtailment is more present than during the winter week (Figure 22). Also, the BT is more requested to supply the load during peak hours. Even with a high renewable production, the genset still needs to operate. Add the P_{curt} to the optimization problem can reduce its value by reducing its installed power or augmenting the BT storage capacity. The optimal solution between these two is the one with lowest cost that respect the constraints, but a higher BT storage capacity can reduce the use of the GS.

Figure 22 – Hourly dispatch for week from 11th September to 17th September for Scenario 2.



Source: the author.

Figure 23 – Hourly dispatch for week from 11th September to 17th September for Scenario 4.



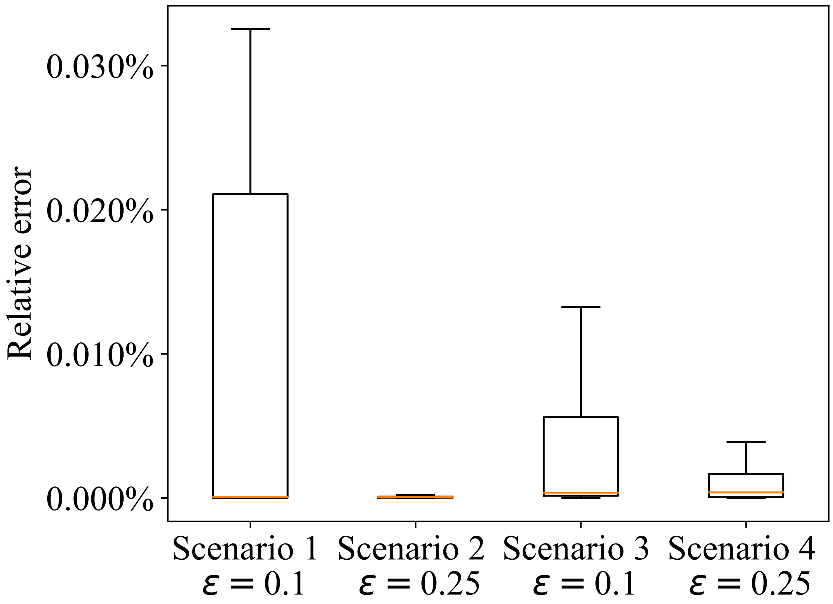
Source: the author.

4.9 Results summary

This section presents the summary of the obtained results. Figure 24 presents the boxplot of the relative errors $\delta_{rlx,i}$ and Fig. 25 of $\delta_{ori \to rlx,i}$, for all studied scenarios with the best relaxation factors ε. In this kind of graph, the base of the box is the first quartile (Q1) and the top the third quartile (Q3). The orange line is the median, also known as the second quartile (Q2). The difference between Q1 and Q3 is the interquartile range (IQR), and it is used to determine the minimum as $Q1 - 1.5 \cdot IQR$ and the maximum as $Q3 + 1.5 \cdot IQR$, which are the whiskers of the boxplot. Values outside the range between minimum and maximum are considered outliers and are not shown in the graph.

Both errors, $\delta_{rlx,i}$ and $\delta_{ori \to rlx,i}$, presents low values, implying that the relaxation reduced the discontinuities and allowed a better convergence of the gradient-based algorithm. Further, having low $\delta_{ori \to rlx,i}$ indicates that the relaxation made in the GS operation hours is a good approximation of the original modeling, for the presented relaxation factors ε. However, the best relaxation factor ε was not the same for all scenarios, suggesting that a tuning of this parameter is essential. This is a concern already presented in metaheuristic algorithms, in which many parameters need to be tuned, e.g., three parameters for PSO (the inertia weight and the individual and social intelligence constants).

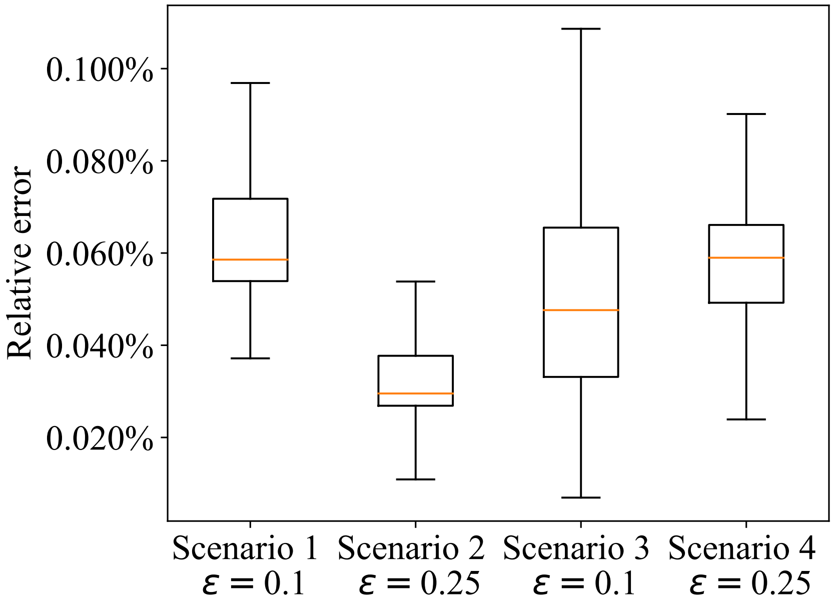
Figure 24 – Boxplot of the relative errors $\delta_{rlx,i}$ for the 4 scenarios with the best relaxations factors ε.



Source: the author.

Table 28 presents the median execution time for all studied scenarios. It can be observed an expressive difference between the times for the gradient-based+AD and the PSO. It is noteworthy to mention the scenarios 3 and 4 presented lower execution time than scenarios 1 and 2 for PSO, because the number of iterations was reduced from 100 to 50. Following

Figure 25 – Boxplot of the relative errors $\delta_{ori \rightarrow rlx,i}$ for the 4 scenarios with the best relaxations factors ε.



Source: the author.

the time tendency of the first two scenarios, Scenario 4 would present an execution time in the order of 600 s with 100 iterations.

Table 28 – Comparison of median execution time for all scenarios with gradient-based+AD and PSO algorithms.

	Gradient-based+AD	PSO
Scenario 1	502.087 ms	149.478 s
Scenario 2	869.996 ms	297.087 s
Scenario 3	1.684 s	83.984 s
Scenario 4	2.239 s	174.567 s

Considering all the presented results, the proposed methodology is suitable to solve the islanded microgrid optimal sizing problem. Optimal values are reached for low execution times, and the results are in the same magnitude order of those obtained with the well established PSO algorithm. Therefore the gradient-based+AD approach is a better choice when many scenarios and cases need to be tested. Another application is when the problem is modeled as a multi-objective optimization problem, and multi-objective algorithms that evaluate multiple times a single-objective function are used to solve it, i.e., ε-constrained algorithm.

5 CONCLUSION

This work proposed a methodology to obtain the optimal components sizing for a islanded microgrid. The approach consists of the combined use of a Microgrid Simulator, a gradient-based optimization algorithm and the Automatic Differentiation method to obtain the gradients.

First, the MG Simulator was developed, comprising various indicators that can be used as functions for the optimization problem. Further, case studies were carried for an isolated location, the french Ushant Island. The performance of the proposed method was analyzed for four different scenarios, using the NPC as objective function and the shedding factor (SF) as constraint when needed.

For all of them, the optimization with the original modeling presented high errors. However, the proposed relaxation of the diesel genset operation hours was sufficient to reduce the model discontinuities and improve the convergence performance. The mean efficiency of the proposed algorithm has the same order of magnitude than the PSO, one of the most used metaheuristic algorithms for this kind of problem.

The computation time for the proposed algorithm is much lower than that of PSO, enabling a faster tuning of the gradient-based+AD algorithm parameter ε than the tuning of PSO parameters. The proximity of the optimal results attained with the two algorithms is enough to recommend the use of the gradient-based+AD algorithm, specially when there are more variables in the optimization problem or many different cases need to be considered. Further, the simulation time of the microgrid's operation is very low with the developed MG Simulator, allowing to quickly perform sensitivity analysis of the optimal result for various parameters.

The results suggest that it is possible to use gradient-based algorithms to solve the microgrid sizing problem. However, proper relaxations in the models need to be made to reduce the discontinuities.

5.1 Future works

Based on the analysis of the obtained results in this work, the following topics are proposed for further research and exploration:

- add other electric resources models in the Microgrid Simulator;
- change the modeling approach from single-node to multi-node, including the power flow equations;
- use the proposed methodology with the ε-constrained algorithm to solve a multi-objective optimization problem;
- assess the methodology performance with other gradient-based optimization algorithms, e.g. IPOPT;
- implement a hybrid algorithm with gradient-based+AD and a multi-start optimization algorithm.

5.2 Derivative works

The following works were developed as a result of the presented research:

- Publication – Under Review

  (Antunes *et al.*, 2022) Conference article for the IEEE PES ISGT Europe 2022

- Julia Package

  (Antunes; Sadou; Haessig, 2021) Microgrids.jl - Microgrid simulation package in Julia language

REFERENCES

Antunes, E. G.; Sadou, N.; Haessig, P. **Microgrids.jl**. 2021.

Antunes, E. G. *et al.* Optimal Microgrid Sizing using Gradient-based Algorithms with Automatic Differentiation. *In*: ISGT EUROPE 2022, NOVI SAD, SERBIA, 2022. **Proceedings [. . .]** [S.l.: s.n.], 2022.

Baydin, A. G. *et al.* Automatic differentiation in machine learning: a survey. **Journal of machine learning research**, [S.l.], v. 18, 2018.

Bezanson, J. *et al.* Julia: A Fast Dynamic Language for Technical Computing. **arXiv:1209.5145 [cs]**, [S.l.], Sept. 2012. arXiv: 1209.5145.

Bukar, A. L.; Tan, C. W.; Lau, K. Y. Optimal sizing of an autonomous photovoltaic/wind/battery/diesel generator microgrid using grasshopper optimization algorithm. **Solar Energy**, [S.l.], v. 188, p. 685–696, 2019.

Chen, J.; Revels, J. Robust benchmarking in noisy environments. **arXiv e-prints**, [S.l.], Aug 2016.

Clerc, M.; Kennedy, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. **IEEE Transactions on Evolutionary Computation**, [S.l.], v. 6, n. 1, p. 58–73, 2002.

ENERCON GmbH. **ENERCON E-70 E4**. [S.l.]: ENERCON GmbH, 2016. Accessed: 2022-01-20.

EPE. **Balanço Energético Nacional 2021**: ano base 2020. [S.l.: s.n.], 2021.

EPE. **Planejamento do Atendimento aos Sistemas Isolados Horizonte 2026 - Ciclo 2021**. [S.l.: s.n.], 2022.

European Commission Joint Research Centre. **Photovoltaic Geographical Information System (PVGIS)**. 2022.

Gamarra, C.; Guerrero, J. M. Computational optimization techniques applied to microgrids planning: A review. **Renewable and Sustainable Energy Reviews**, [S.l.], v. 48, p. 413–424, Aug. 2015.

Gebremedhin, A. H.; Walther, A. An introduction to algorithmic differentiation. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, [S.l.], v. 10, n. 1, p. e1334, 2020.

Gilbert, J. C.; Le Vey, G.; Masse, J. **La différentiation automatique de fonctions représentées par des programmes**. 1991. Tese (Doutorado em Engenharia Elétrica) — INRIA, 1991.

Godsil, C.; Royle, G. F. **Algebraic Graph Theory**. New York, NY: Springer New York, 2001.

Griewank, A.; Walther, A. **Evaluating derivatives**: principles and techniques of algorithmic differentiation. [S.l.]: SIAM, 2008.

Hemker, T. *et al.* A mixed-integer simulation-based optimization approach with surrogate functions in water resources management. **Optimization and Engineering**, [S.l.], v. 9, n. 4, p. 341–360, Dec 2008.

Javed, M. S. *et al.* Performance comparison of heuristic algorithms for optimization of hybrid off-grid renewable energy systems. **Energy**, [S.l.], v. 210, p. 118599, 2020.

Johnson, S. G. **The NLopt nonlinear-optimization package**. 2007–2020.

Kennedy, J.; Eberhart, R. Particle swarm optimization. *In*: ICNN'95 - International Conference ON Neural Networks, 1995. **Proceedings [. . .]** [S.l.: s.n.], 1995. v. 4, p. 1942–1948 vol.4.

Kraft, D. **A software package for sequential quadratic programming**. Oberpfaffenhofen: [s.n.], 1988.

Kraft, D. Algorithm 733: TOMP – Fortran modules for optimal control calculations. **ACM Transactions on Mathematical Software**, [S.l.], v. 20, p. 262–281, Sept. 1994.

Lambert, T.; Gilman, P.; Lilienthal, P. Micropower system modeling with HOMER. In: **Integration of alternative sources of energy**. [S.l.: s.n.], 2006. v. 1, p. 379–385. Publisher: John Wiley & Sons New York, NY, USA.

Lazard. **Lazard's Levelized cost of energy analysis - Version 14.0**. [S.l.]: Lazard, 2020.

Lazard. **Lazard's Levelized cost of storage analysis - Version 6.0**. [S.l.]: Lazard, 2020.

Luenberger, D. G.; Ye, Y. **Linear and Nonlinear Programming**. Cham: Springer International Publishing, 2016. (International Series in Operations Research & Management Science, v. 228).

Mashayekh, S. *et al.* A mixed integer linear programming approach for optimal DER portfolio, sizing, and placement in multi-energy microgrids. **Applied Energy**, [S.l.], v. 187, p. 154–168, Feb. 2017.

Mejía, J.; Satman, M. H.; Monticone, P. **jmejia8/Metaheuristics.jl**: v3.2.6. [S.l.]: Zenodo, 2022.

Mohammed, O. H. *et al.* Optimal sizing and energy management of hybrid wind/tidal/PV power generation system for remote areas: application to the ouessant french island. *In*: IECON 2016 - 42ND ANNUAL CONFERENCE OF THE IEEE INDUSTRIAL ELECTRONICS SOCIETY, 2016. **Proceedings [...]** [S.l.: s.n.], 2016. p. 4205–4210.

Nielsen, T. **Optimization of buildings with respect to energy and indoor environment**. 2003. Tese (Doutorado em Engenharia Elétrica) — , 2003. (R-036).

Nocedal, J.; Wright, S. J. **Numerical optimization**. 2nd. ed. New York: Springer, 2006. (Springer series in operations research).

Revels, J.; Pearson, J. **ReverseDiff.jl**. 2016.

Suman, G. K.; Guerrero, J. M.; Roy, O. P. Optimisation of solar/wind/bio-generator/diesel/battery based microgrids for rural areas: A PSO-GWO approach. **Sustainable Cities and Society**, [S.l.], v. 67, p. 102723, Apr. 2021.

Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in python. **Nature Methods**, [S.l.], v. 17, p. 261–272, 2020.

Zhao, B. *et al.* Optimal sizing, operating strategy and operational experience of a stand-alone microgrid on Dongfushan Island. **Applied Energy**, [S.l.], v. 113, p. 1656–1666, Jan. 2014.

APPENDIX A MICROGRID SIMULATOR VALIDATION

A validation of the developed microgrid simulator was realized. The simulation was performed for a microgrid composed of PV (4063 kW), BT (6744 kWh) and GS (900 kW). The load demand and photovoltaic production were the same from the §4.1, Figures 11 and 12, respectively, and the other parameters are presented in Table 29.

Some indicators values obtained with the MG Simulator were compared with the ones obtained with the HOMER software. They are presented in Table 30 with the deviations, calculated as

$$D = \frac{I_{\text{MG Simulator}} - I_{\text{HOMER}}}{I_{\text{HOMER}}}. \tag{77}$$

It can be observed that the highest difference is 9% for the NPC values. The deviations are probably caused by some modeling differences, as the battery model and the dispatch strategy, which is optimal in HOMER and not rule-based as in the MG Simulator. However, the indicators are all in the same order of magnitude, validating the developed simulator.

Table 29 – Technical and economic parameters used for microgrid simulator validation.

	Parameters	Values
Project	Discount rate	5%
	Timestep	1 h
	Lifetime	25 years
Photovoltaic system	Rated power	4063 kW
	Derating factor	100%
	Lifetime	25 years
	Investment cost	1200.00 $/kW
	O&M cost	20.00 $\frac{\$}{\text{kW}\cdot\text{year}}$
Battery	Rated capacity	6744 kWh
	Initial energy	6744 kWh
	Minimum energy level	0 kWh
	Maximum charge power	-E_{BT}^{rtd}
	Maximum discharge power	E_{BT}^{rtd}
	Linear loss factor	5%
	Maximum cycles	3000
	Lifetime	15 years
	Investment cost	350.00 $/kWh
	O&M cost	10.00 $\frac{\$}{\text{kW}\cdot\text{year}}$
Diesel genset	Power rated	900 kW
	Minimum load ratio	0
	F0	0
	F1	0.240
	Lifetime	15000 h
	Investment cost	400.00 $/kW
	O&M cost	0.02 $/(kW·h$_{oper}$)
	Fuel cost	1.00 $/L

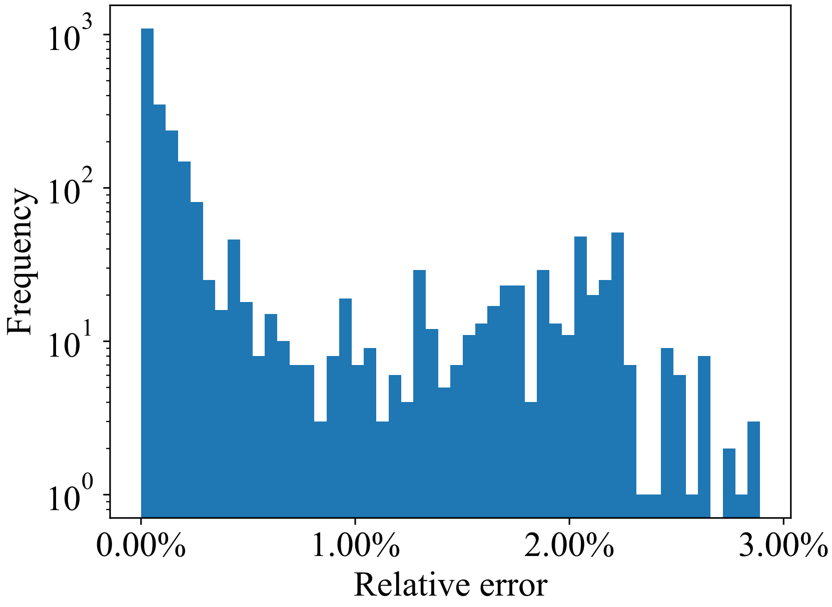
Table 30 – Comparison of indicators values obtained with the developed microgrid simulator and the HOMER software.

Indicator	MG Simulator	HOMER	Deviation
NPC	$ 25,920,253.01	$ 23,879,010.00	9%
COE	0.273 $/kWh	0.263 $/kWh	3%
$h_{GS}^{tot,year}$	4409 h	4514 h	-2%
SF	5.14%	5.07%	2%
RF	50.88%	50.54%	1%

APPENDIX B RELATIVE ERRORS HISTOGRAMS

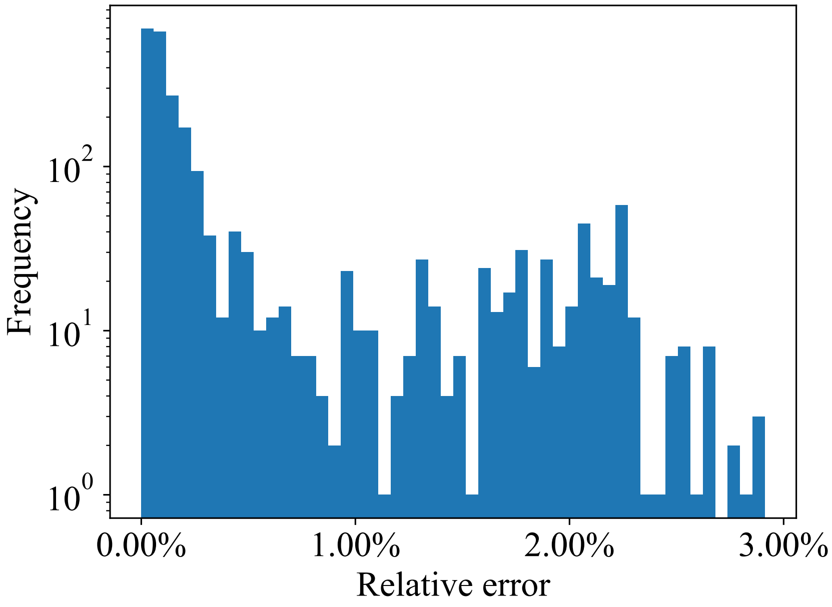
B.1 Scenario 1

Figure 26 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.01.



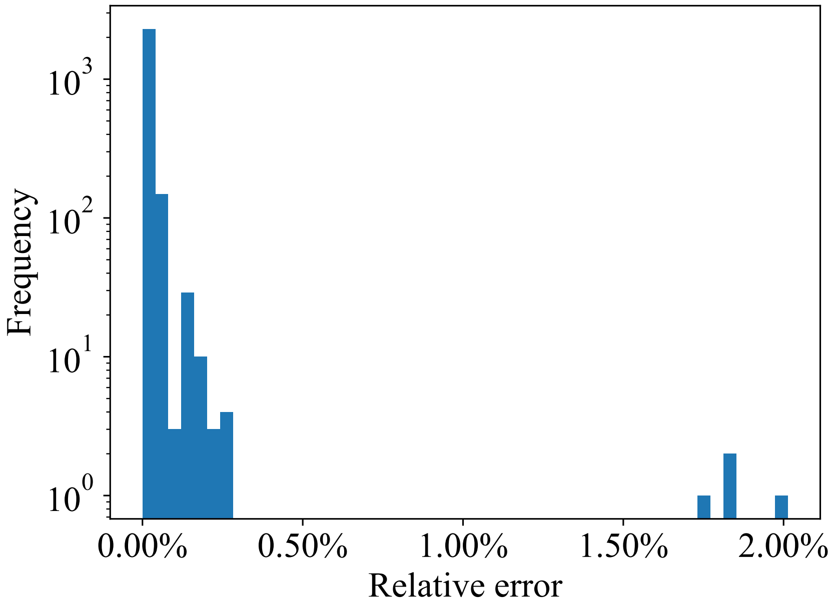
Source: the author.

Figure 27 – Histogram of relative errors $\delta_{ori\to rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.01.



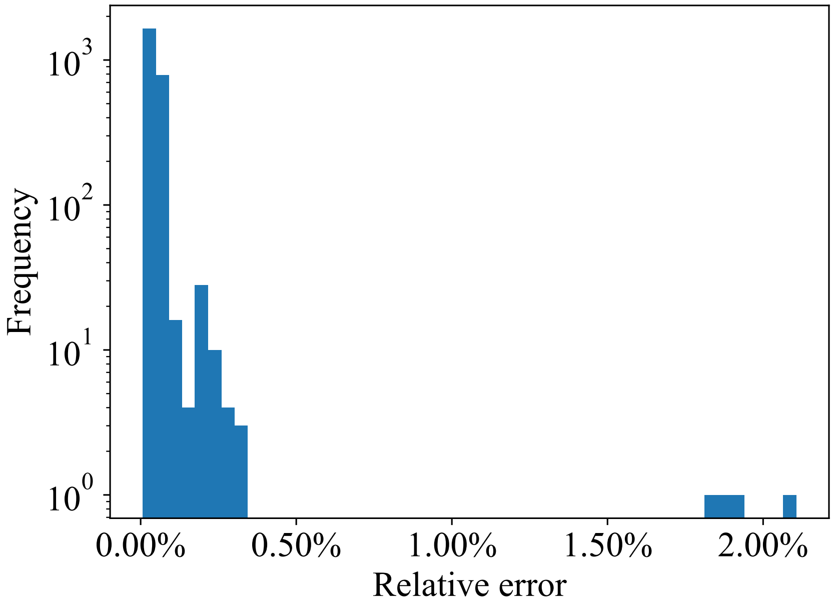
Source: the author.

Figure 28 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.05.



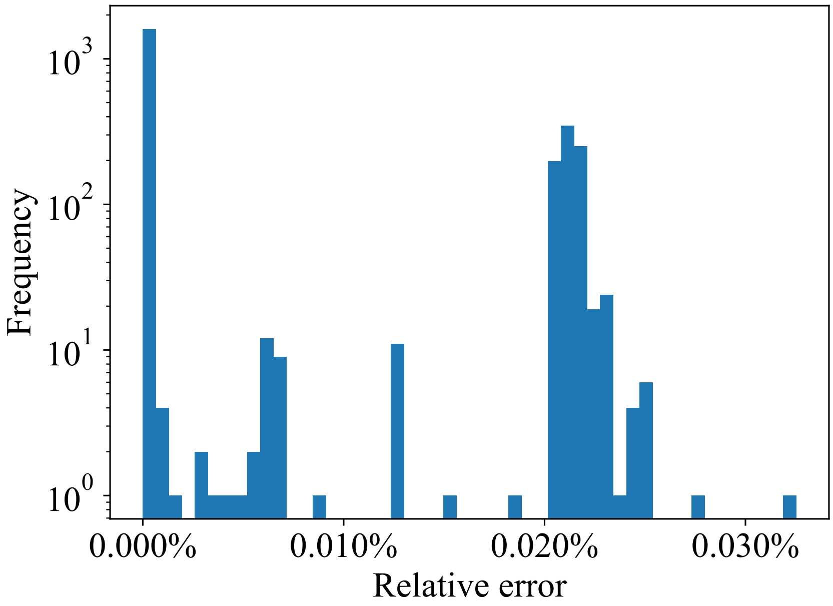
Source: the author.

Figure 29 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.05.



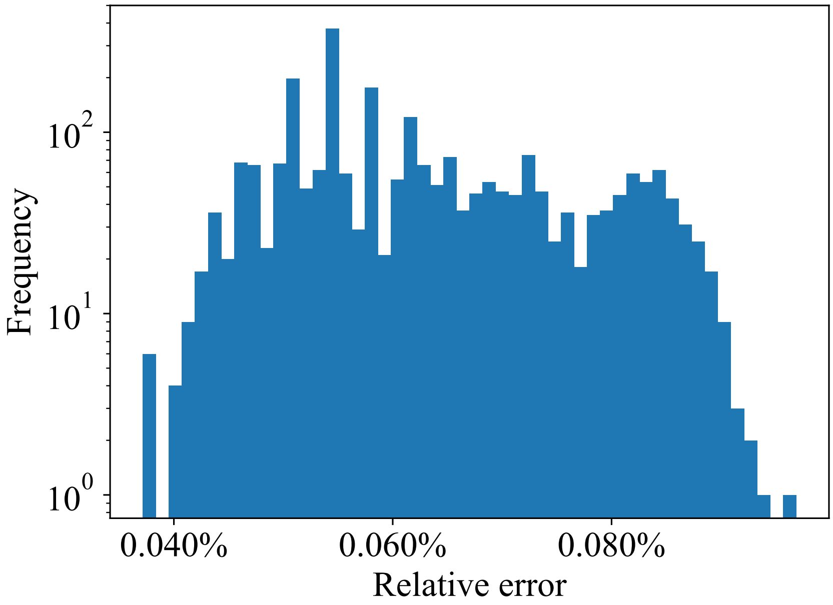
Source: the author.

Figure 30 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.10.



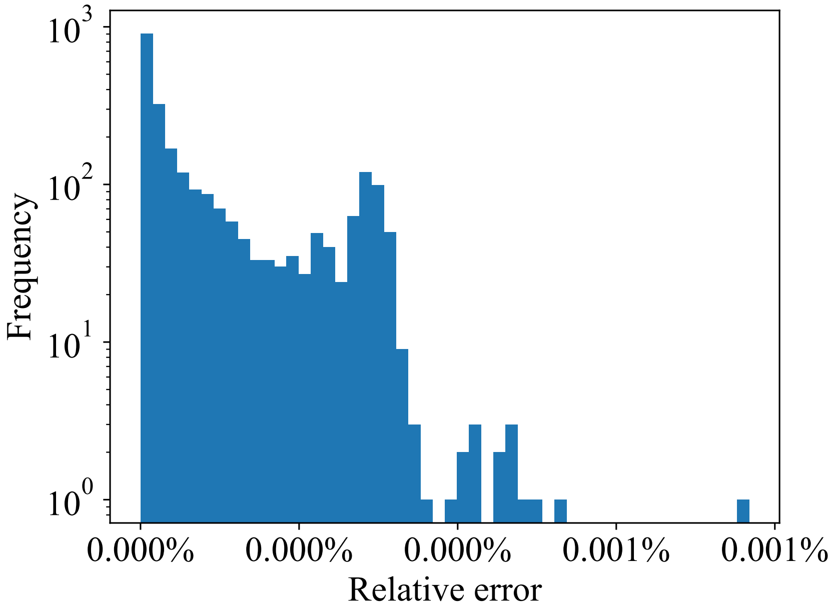
Source: the author.

Figure 31 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.10.



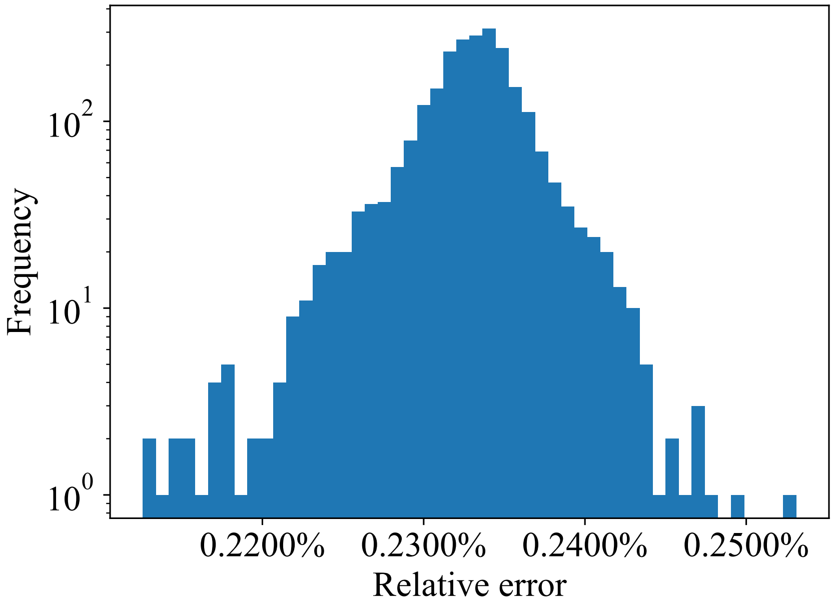
Source: the author.

Figure 32 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.25.



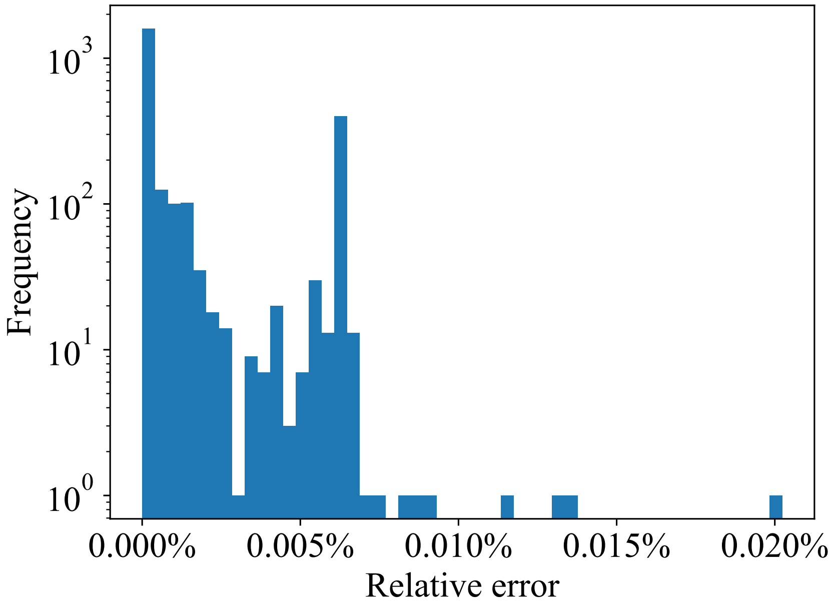
Source: the author.

Figure 33 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.25.



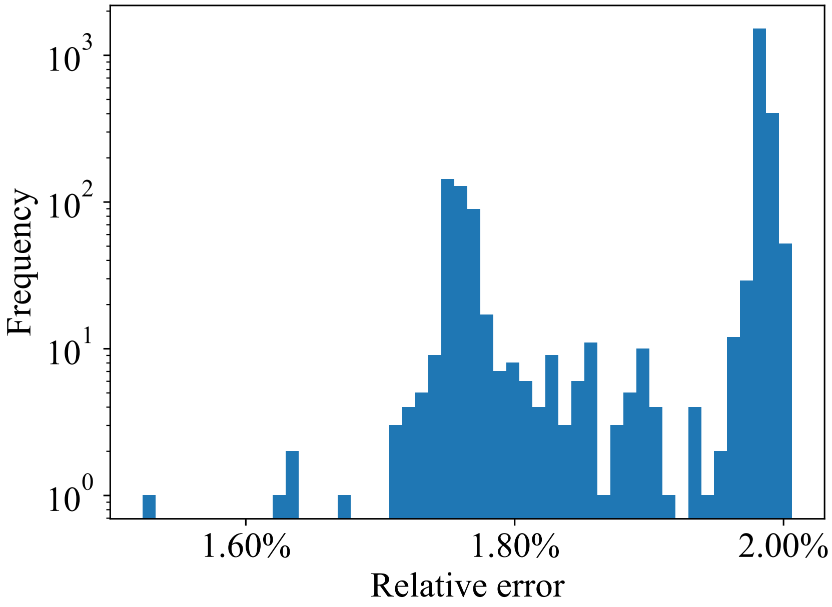
Source: the author.

Figure 34 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.50.



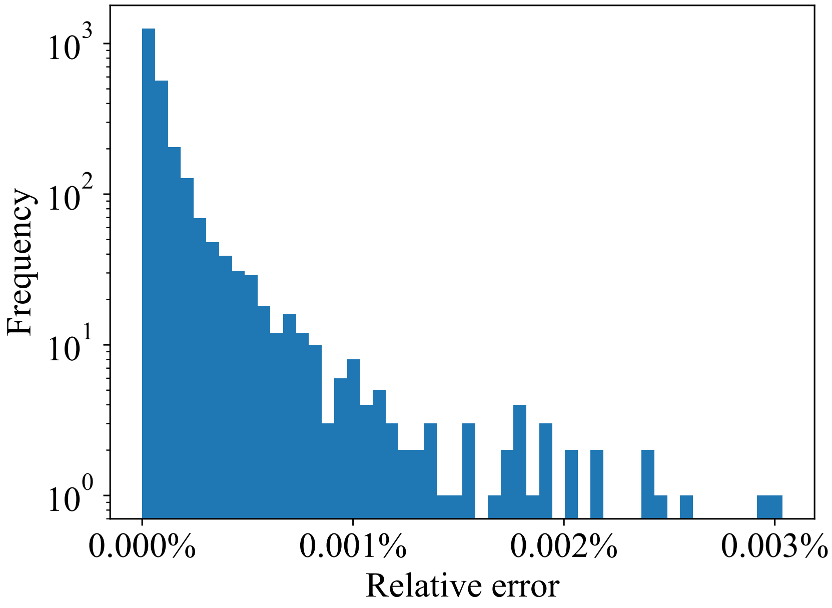
Source: the author.

Figure 35 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.50.



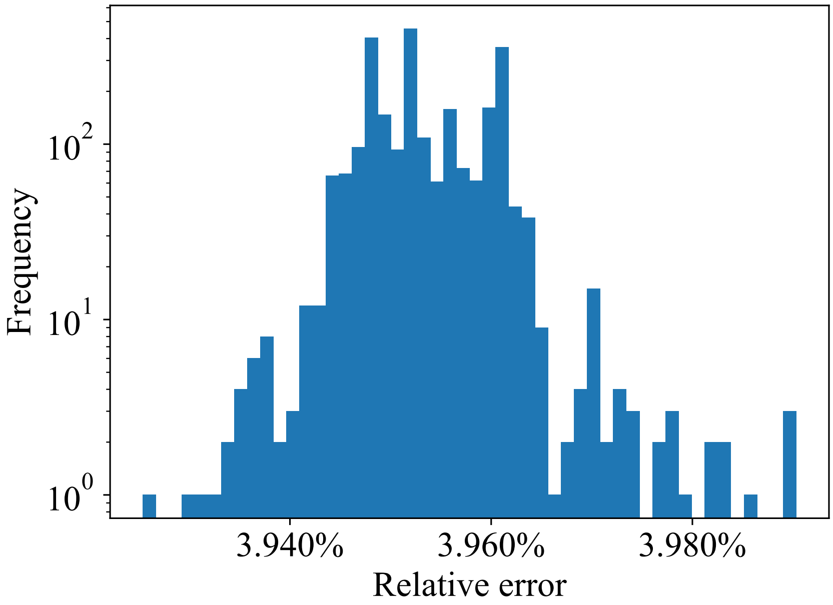
Source: the author.

Figure 36 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.75.



Source: the author.

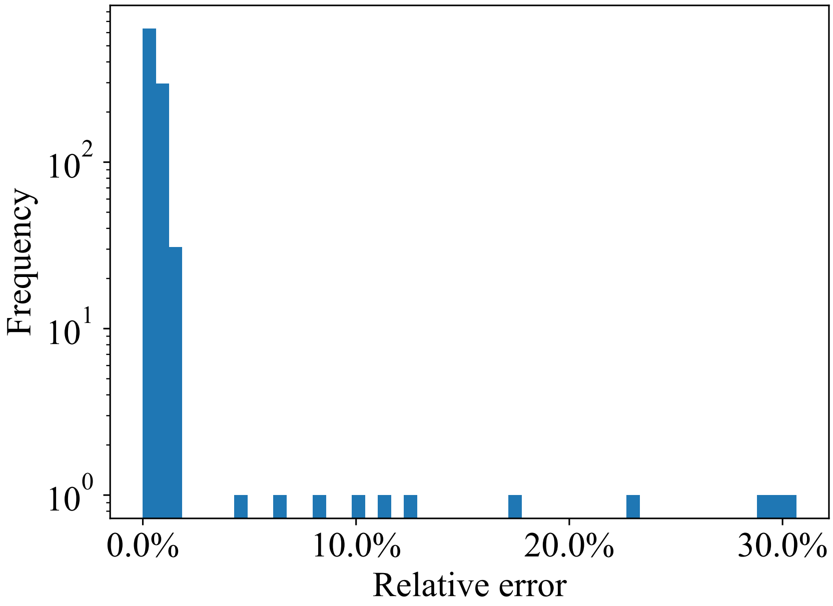
Figure 37 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 1 with relaxation factor ε equal to 0.75.



Source: the author.

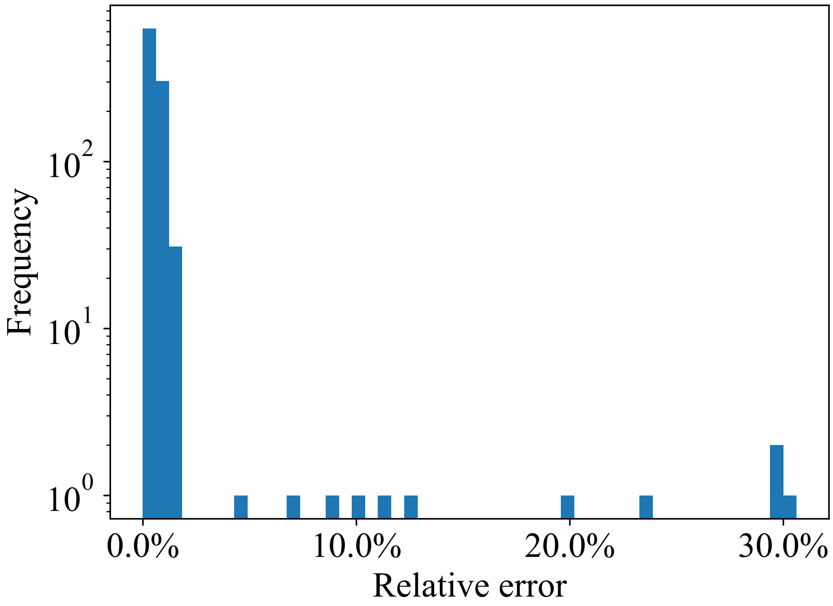
B.2 Scenario 2

Figure 38 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.01.



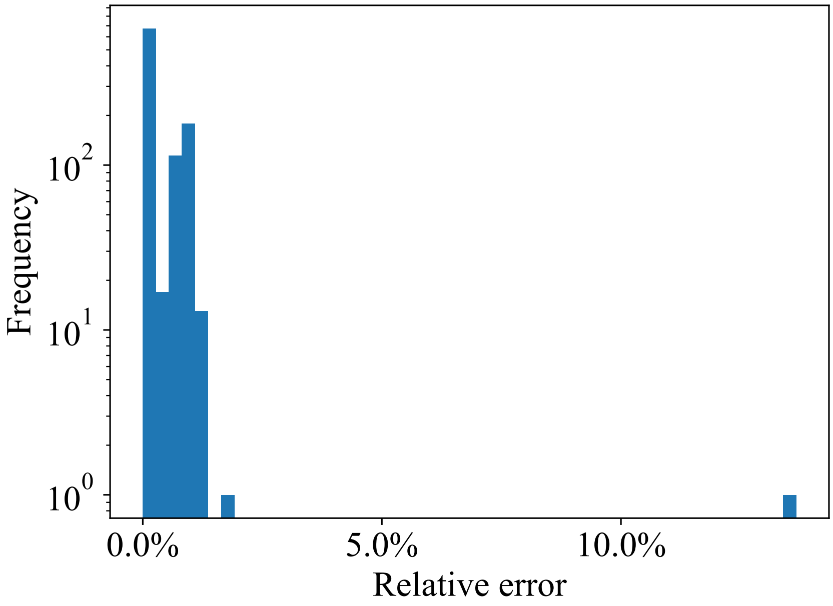
Source: the author.

Figure 39 – Histogram of relative errors $\delta_{ori\rightarrow rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.01.



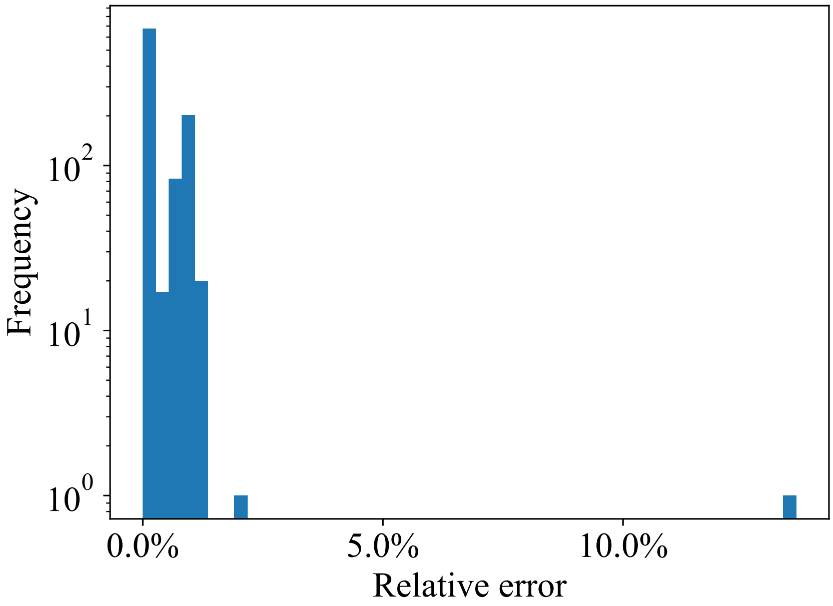
Source: the author.

Figure 40 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.05.



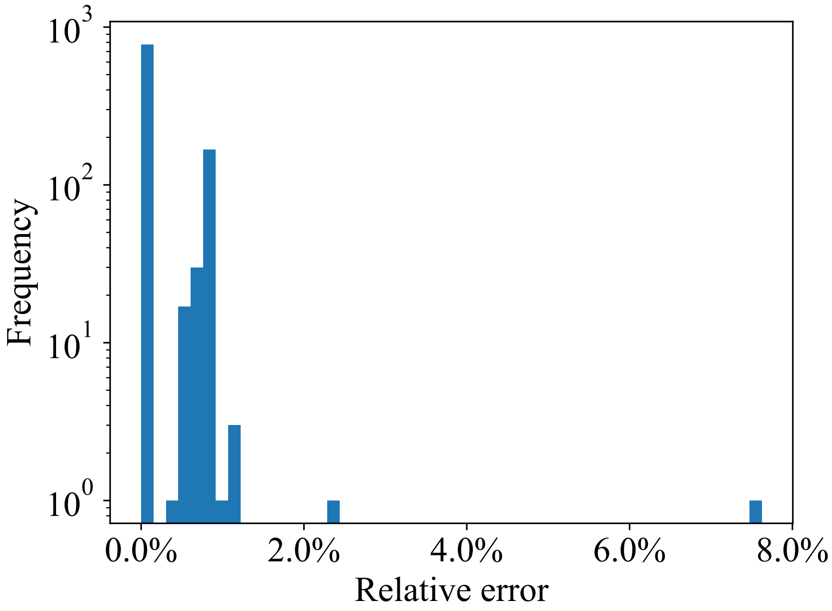
Source: the author.

Figure 41 – Histogram of relative errors $\delta_{ori\to rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.05.



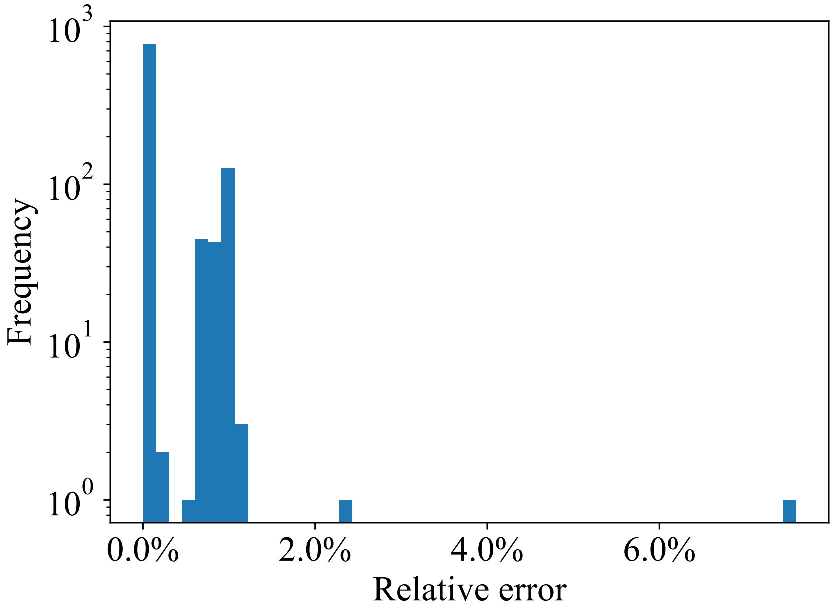
Source: the author.

Figure 42 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.10.



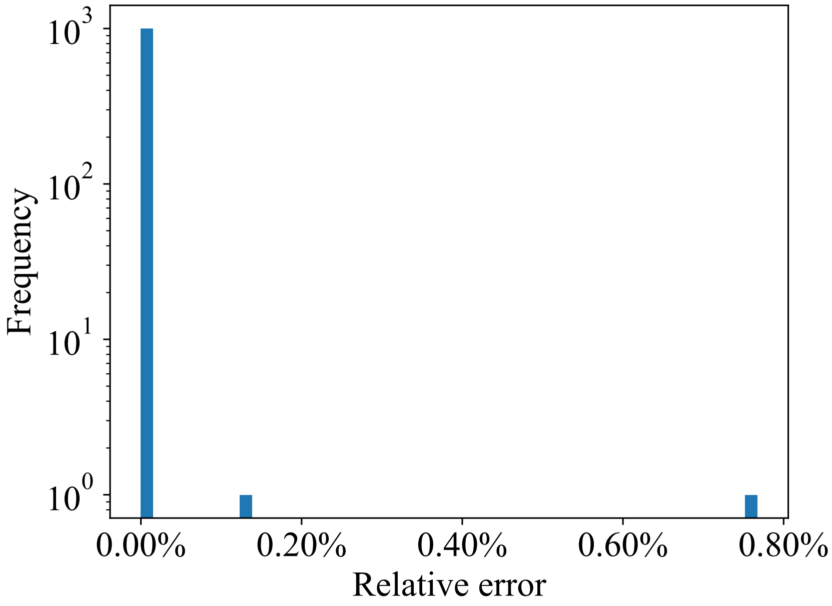
Source: the author.

Figure 43 – Histogram of relative errors $\delta_{ori\rightarrow rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.10.



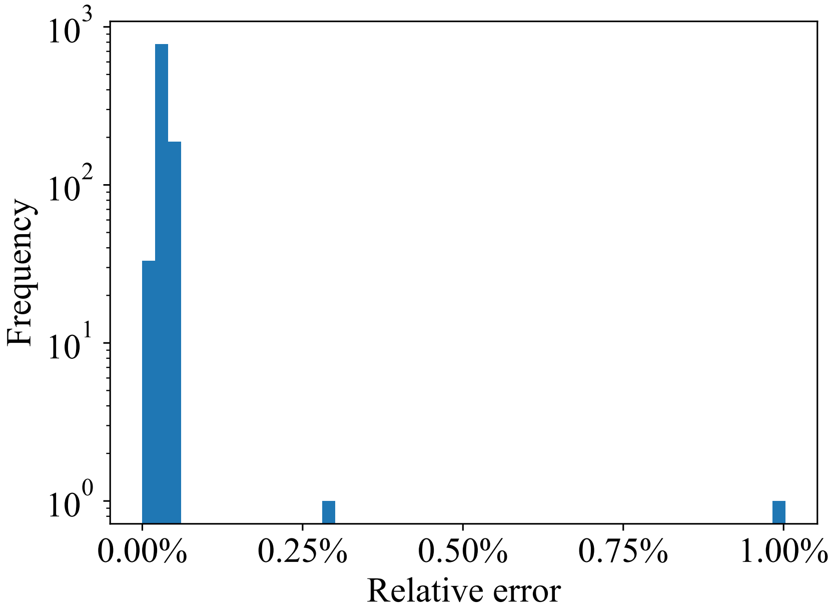
Source: the author.

Figure 44 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.25.



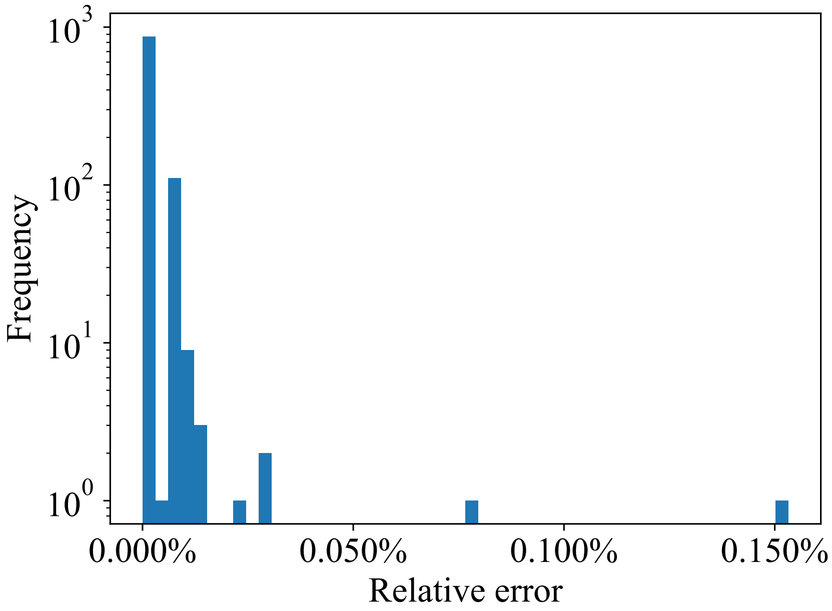
Source: the author.

Figure 45 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.25.



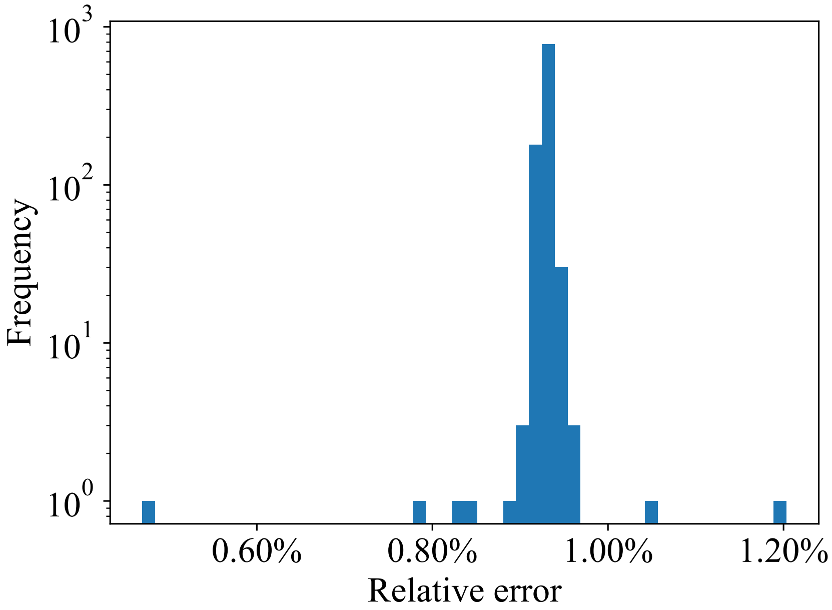
Source: the author.

Figure 46 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.50.



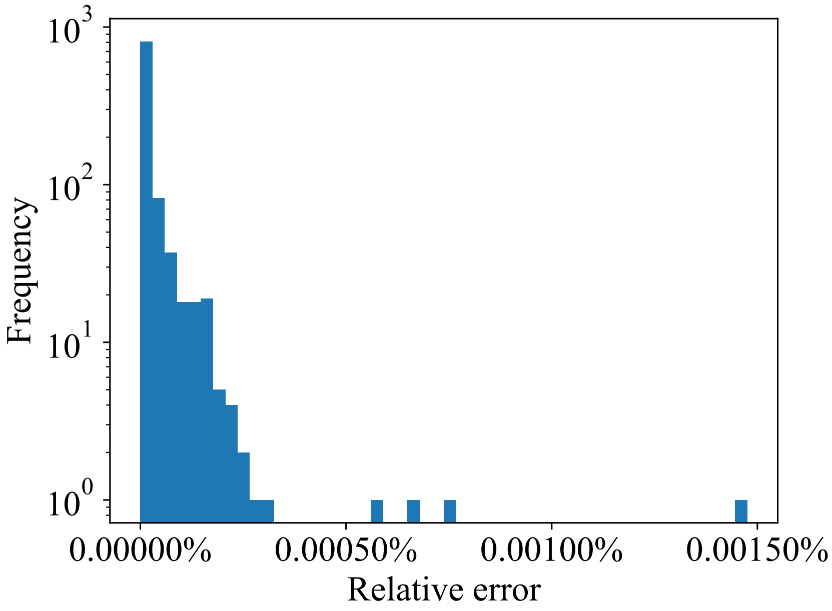
Source: the author.

Figure 47 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.50.



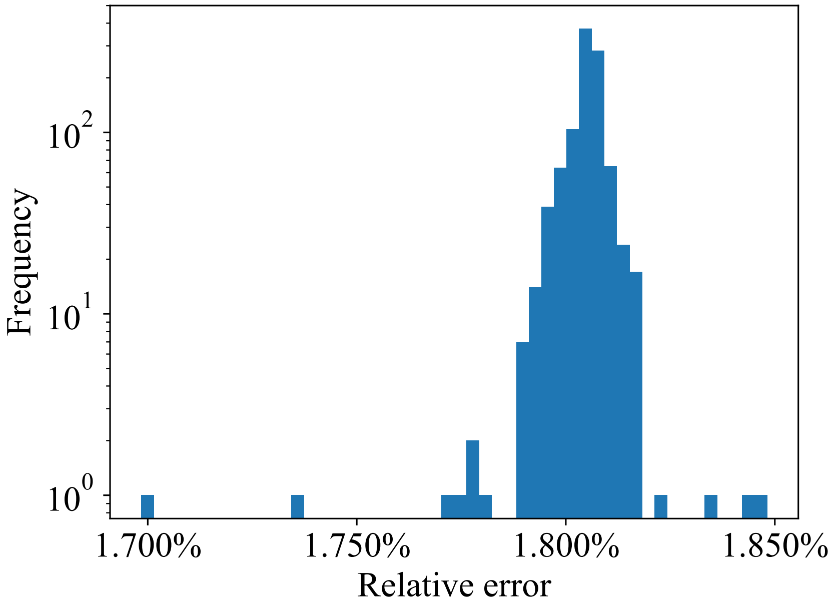
Source: the author.

Figure 48 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.75.



Source: the author.

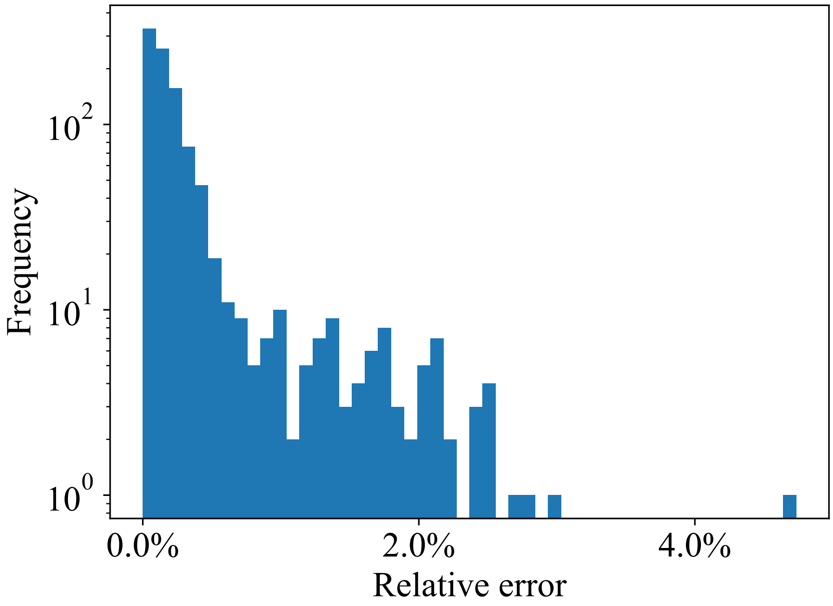
Figure 49 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 2 with relaxation factor ε equal to 0.75.



Source: the author.

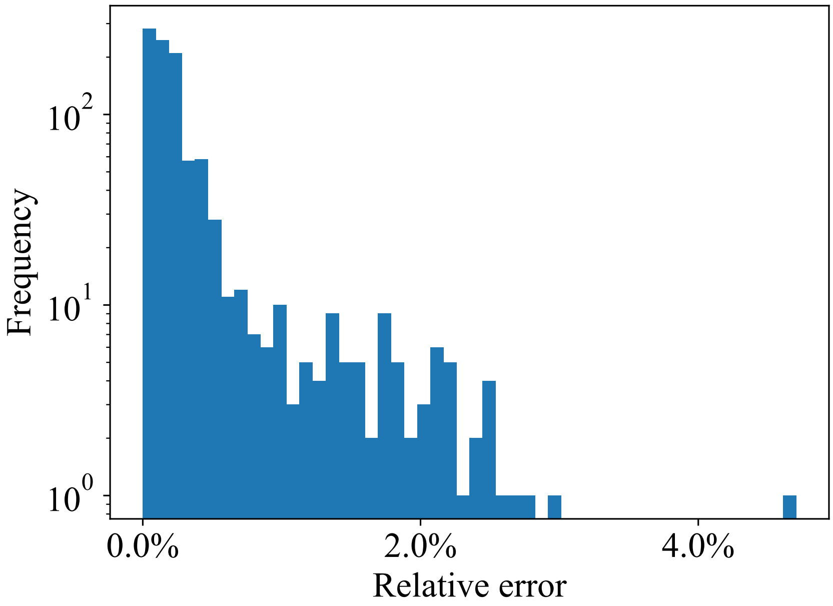
B.3 Scenario 3

Figure 50 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.01.



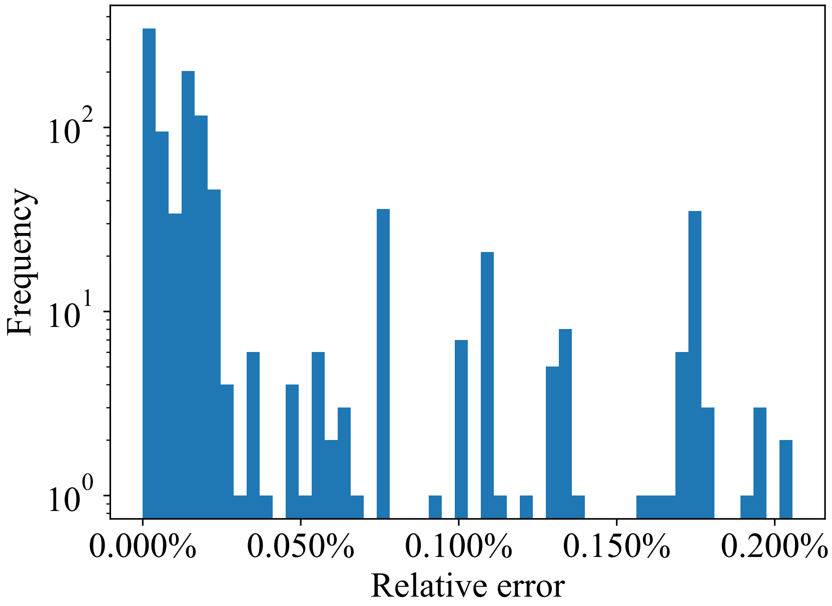
Source: the author.

Figure 51 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.01.



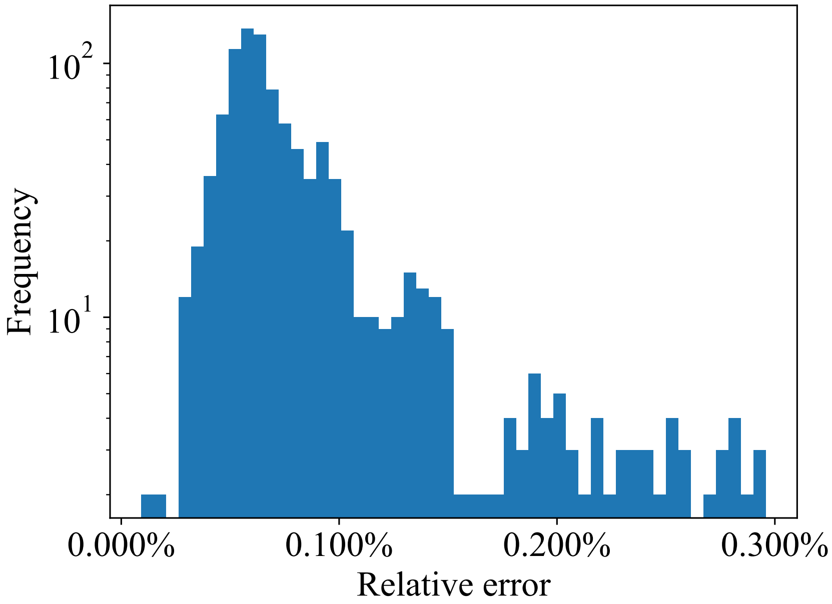
Source: the author.

Figure 52 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.05.



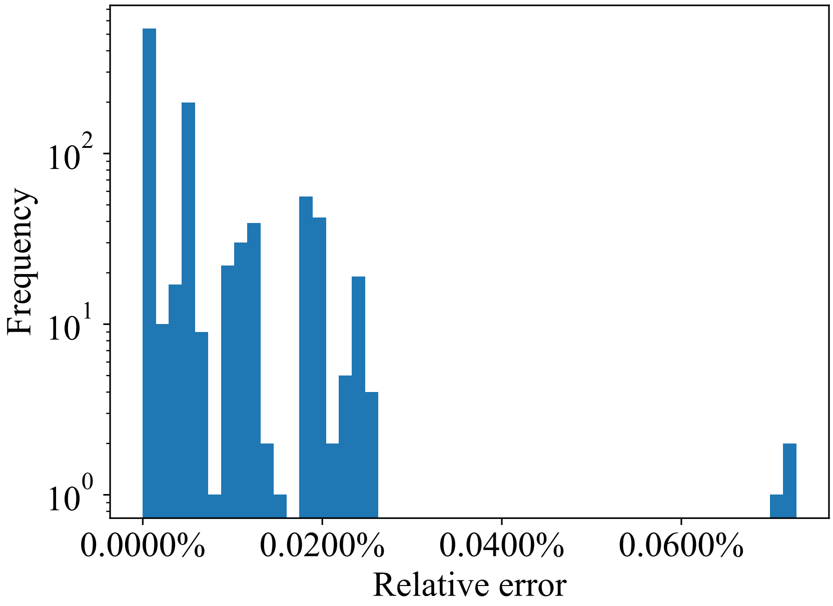
Source: the author.

Figure 53 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.05.



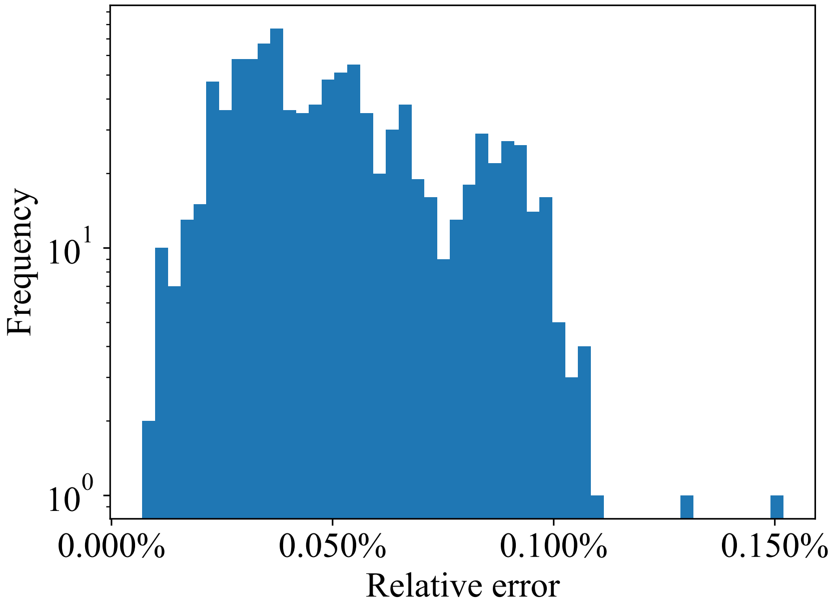
Source: the author.

Figure 54 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.10.



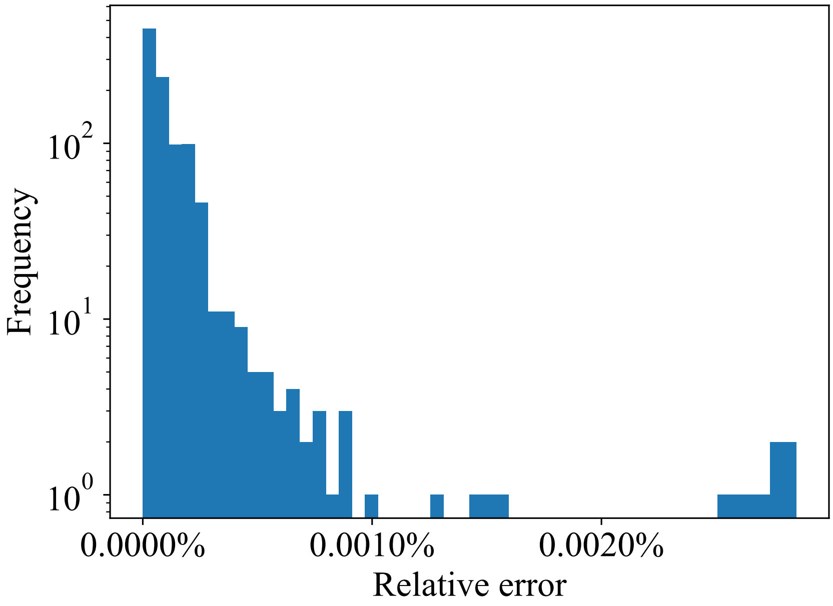
Source: the author.

Figure 55 – Histogram of relative errors $\delta_{ori \rightarrow rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.10.



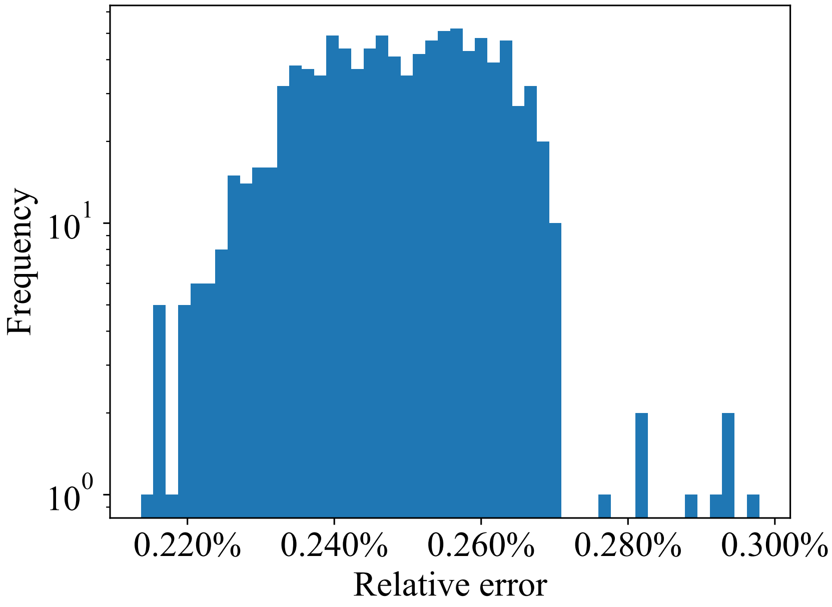
Source: the author.

Figure 56 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.25.



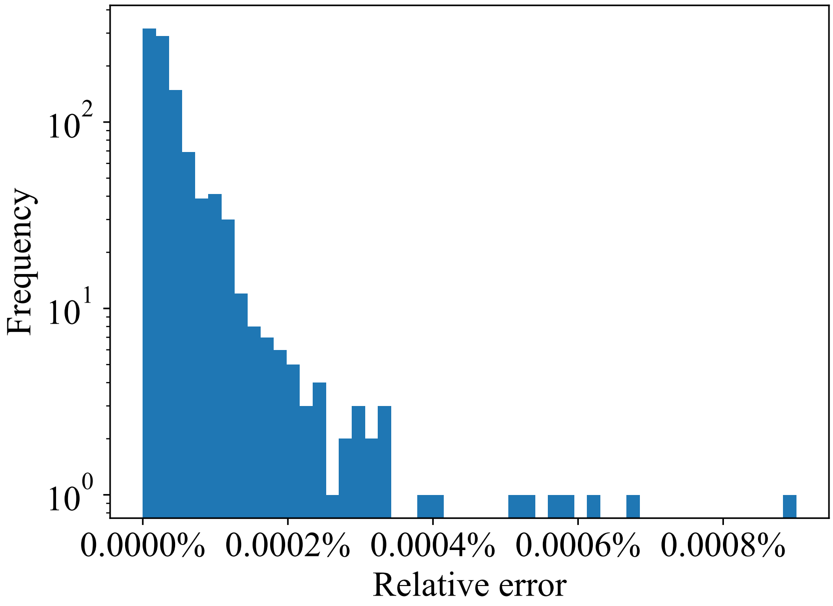
Source: the author.

Figure 57 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.25.



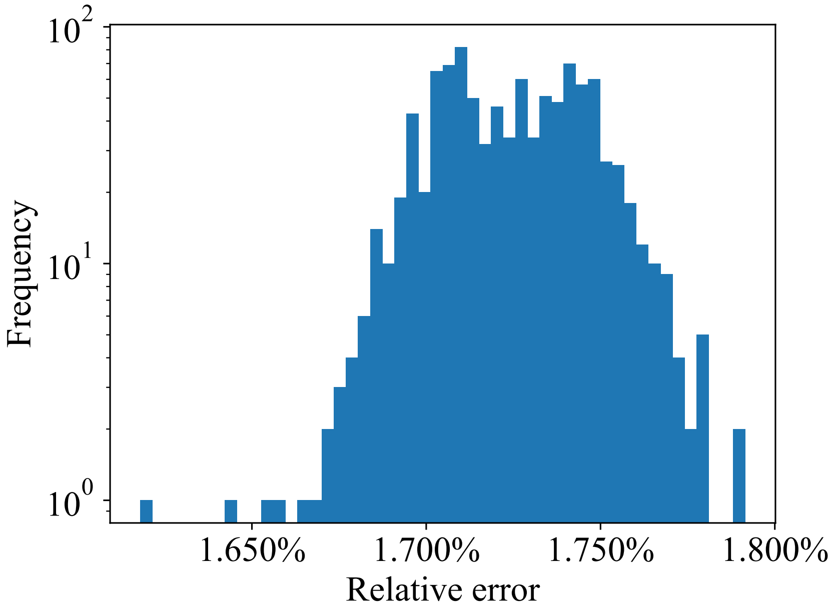
Source: the author.

Figure 58 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.50.



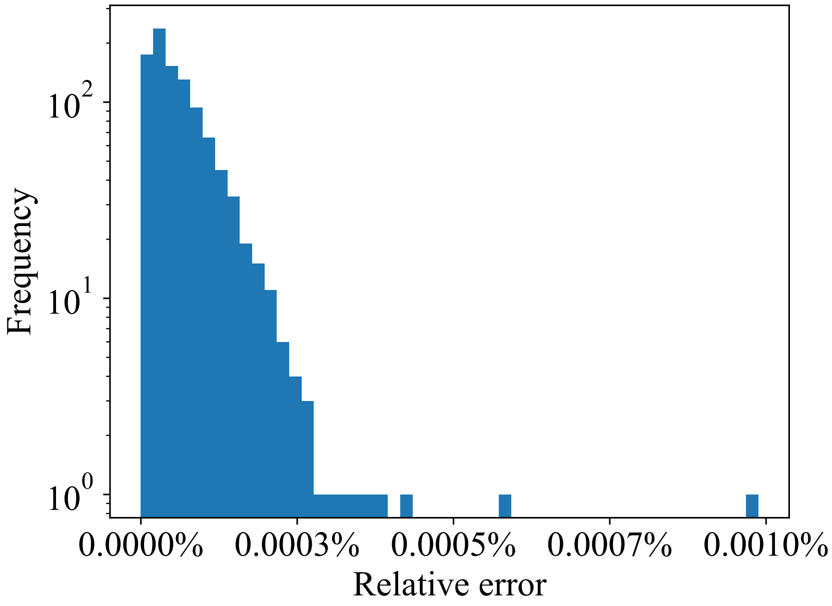
Source: the author.

Figure 59 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.50.



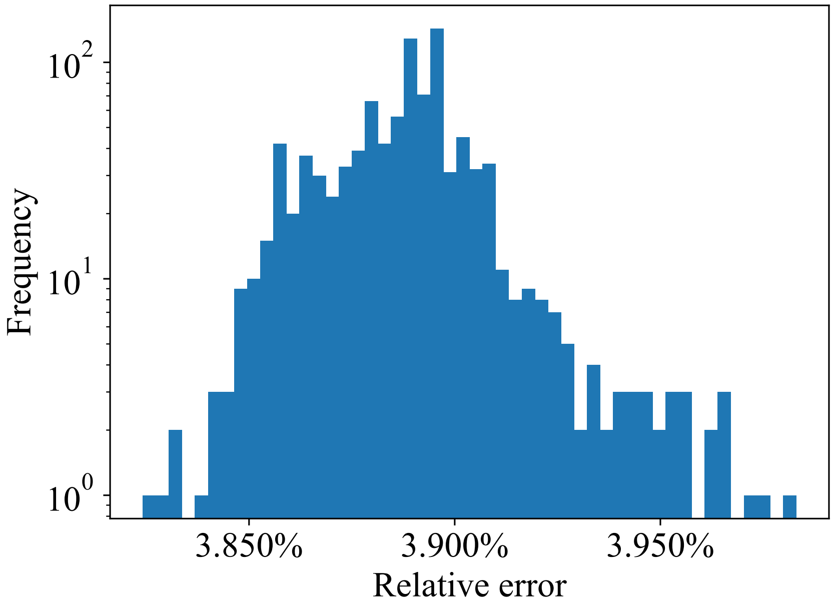
Source: the author.

Figure 60 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.75.



Source: the author.

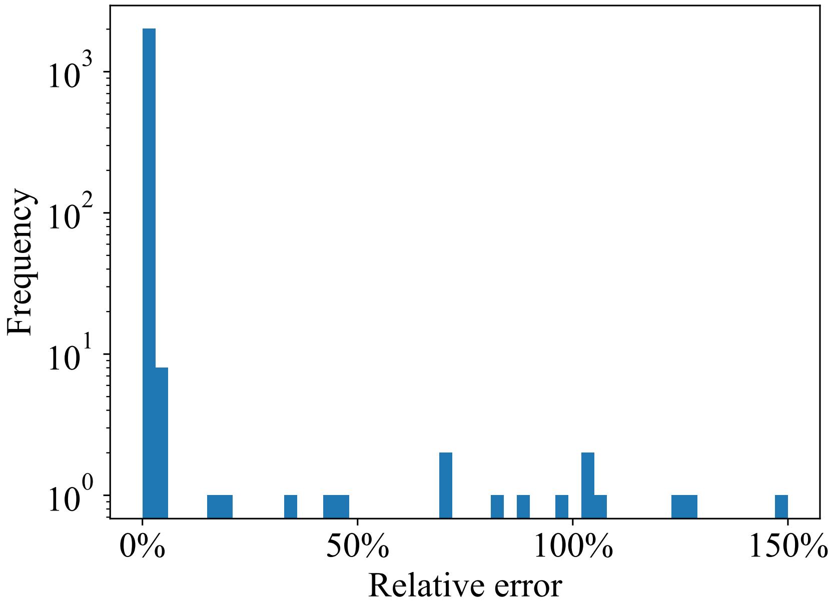
Figure 61 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 3 with relaxation factor ε equal to 0.75.



Source: the author.

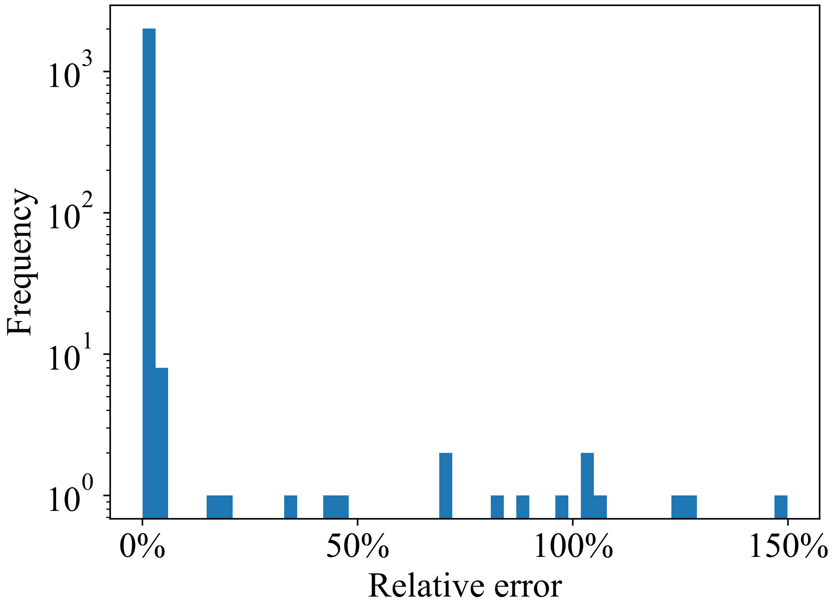
B.4 Scenario 4

Figure 62 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.01.



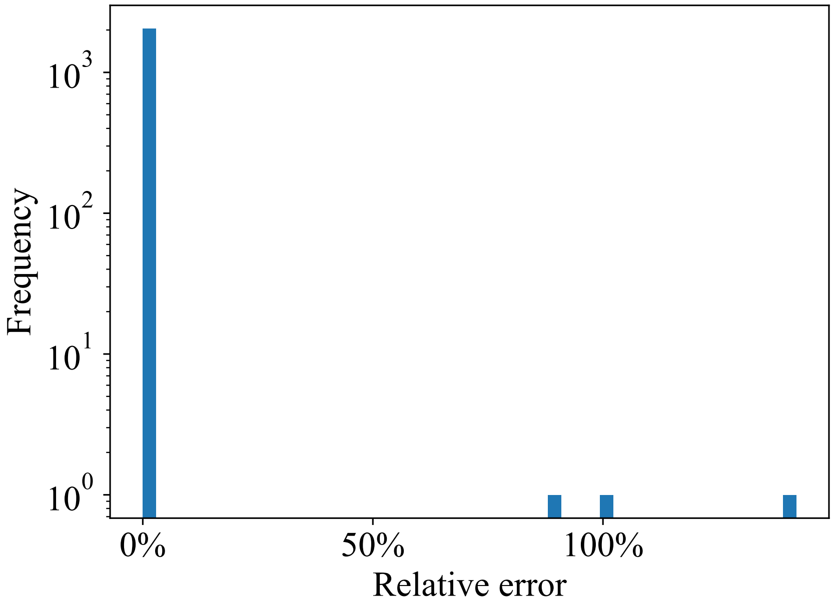
Source: the author.

Figure 63 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.01.



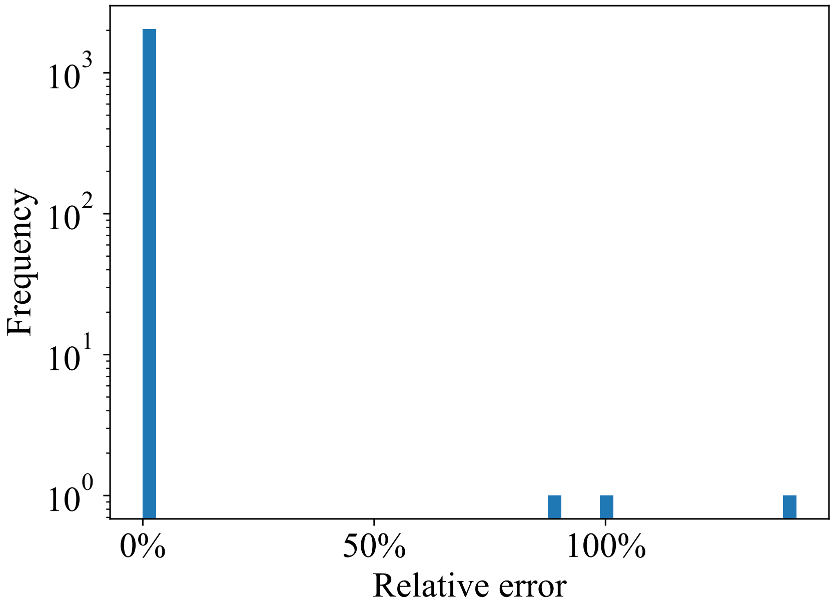
Source: the author.

Figure 64 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.05.



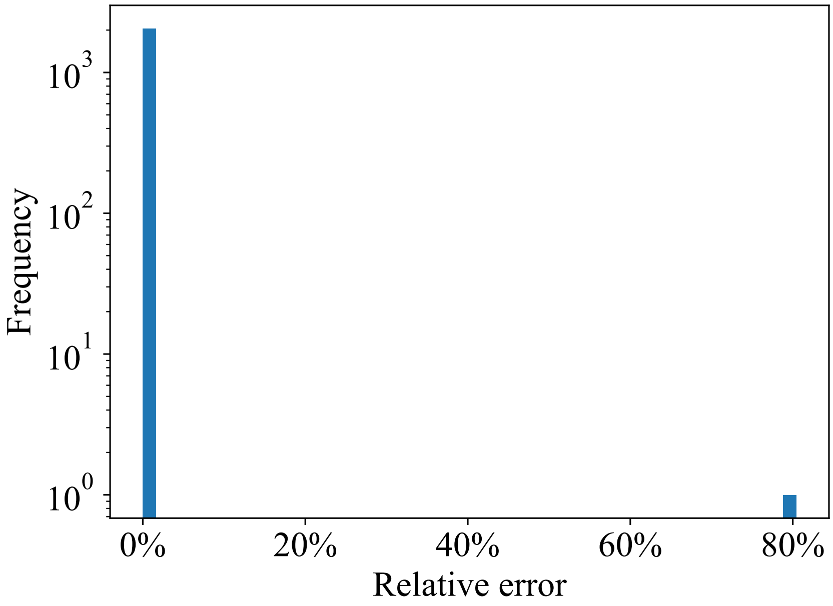
Source: the author.

Figure 65 – Histogram of relative errors $\delta_{ori\rightarrow rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.05.



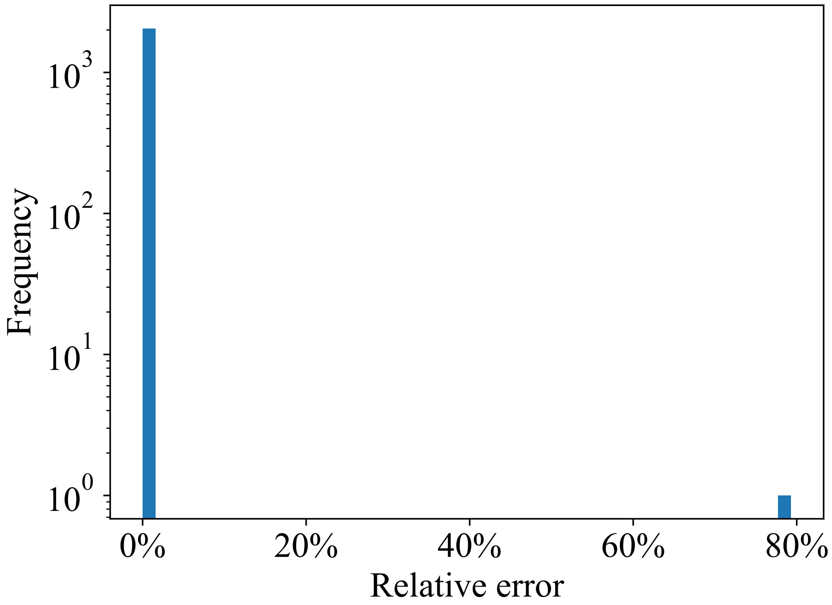
Source: the author.

Figure 66 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.10.



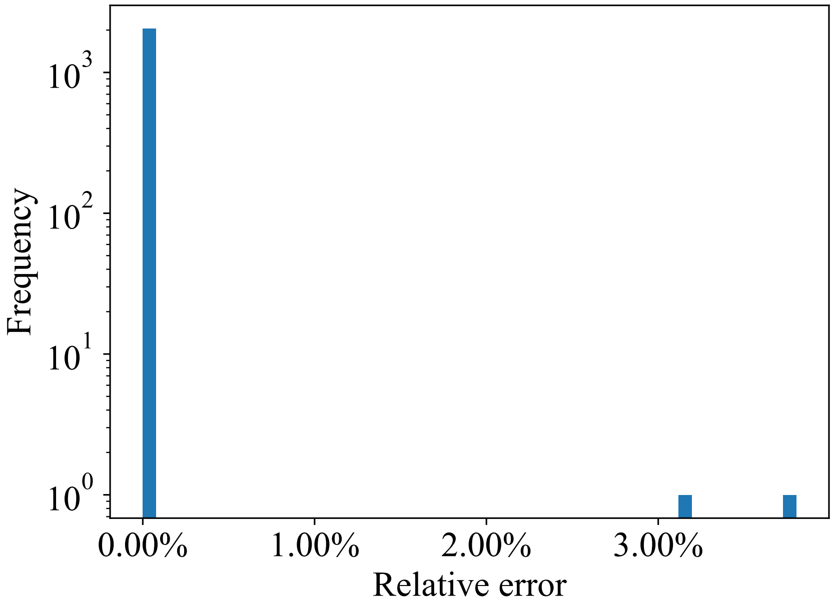
Source: the author.

Figure 67 – Histogram of relative errors $\delta_{ori \to rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.10.



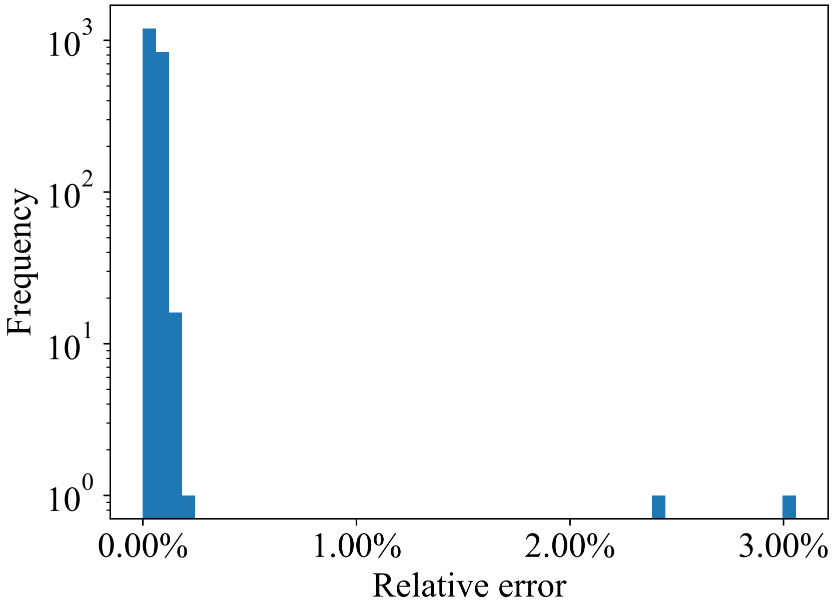
Source: the author.

Figure 68 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.25.



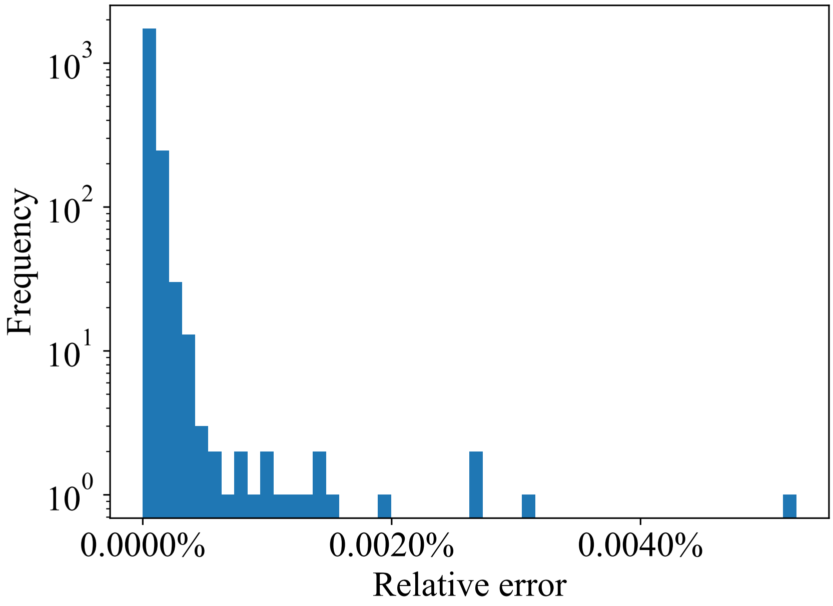
Source: the author.

Figure 69 – Histogram of relative errors $\delta_{ori \rightarrow rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.25.



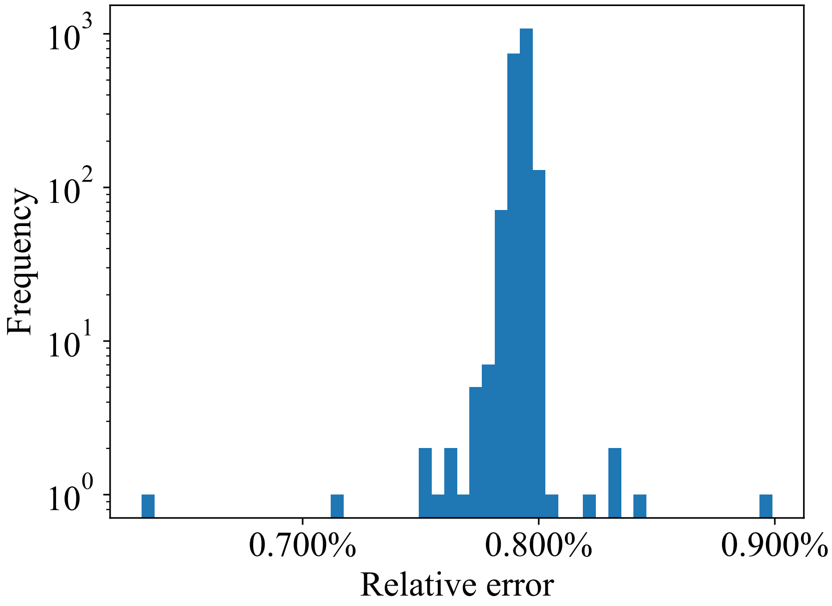
Source: the author.

Figure 70 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.50.



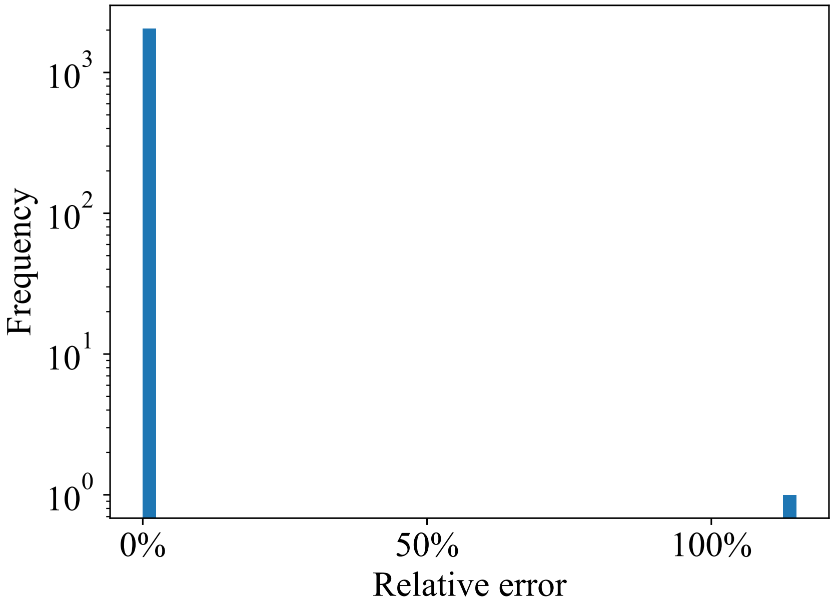
Source: the author.

Figure 71 – Histogram of relative errors $\delta_{ori\rightarrow rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.50.



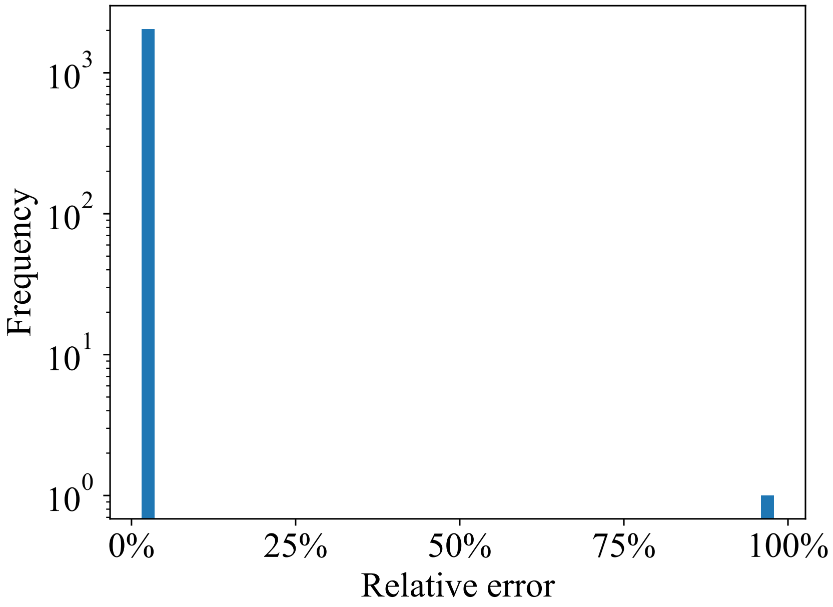
Source: the author.

Figure 72 – Histogram of relative errors $\delta_{rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.75.



Source: the author.

Figure 73 – Histogram of relative errors $\delta_{ori\rightarrow rlx,i}$ for Scenario 4 with relaxation factor ε equal to 0.75.



Source: the author.