Optimal Microgrid Sizing using Gradient-based Algorithms with Automatic Differentiation

Evelise de Godoy Antunes, Pierre Haessig, Chaoyun Wang, Roberto Chouhy Leborgne

Optimal Microgrid Sizing using Gradient-based Algorithms with Automatic Differentiation

Evelise de Godoy Antunes*†, Pierre Haessig†, Chaoyun Wang† and Roberto Chouhy Leborgne* * Federal University of Rio Grande do Sul (UFRGS), Porto Alegre/RS, Brazil
{evelise.antunes, roberto.leborgne}@ufrgs.br † IETR - CentraleSupélec, Cesson-Sévigné, France
pierre.haessig@centralesupelec.fr

***Abstract*—Microgrid sizing optimization is often formulated as a black-box optimization problem. This allows modeling the microgrid with a realistic temporal simulation of the energy flows between components. Such models are usually optimized with gradient-free methods, because no analytical expression for gradient is available. However, the development of new Automatic Differentiation (AD) packages allows the efficient and exact computation of the gradient of black-box models. Thus, this work proposes to solve the optimal microgrid sizing using gradient-based algorithms with AD packages. However, physical realism of the model makes the objective function discontinuous which hinders the optimization convergence. After an appropriate smoothing, the objective is still nonconvex, but convergence is achieved for more that 90 % of the starting points. This suggest that a multi-start gradient-based algorithm can improve the state-of-the-art sizing methodologies.**

***Index Terms*—Automatic differentiation, gradient-based optimization, microgrid, optimal sizing**

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001, by Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq) and by the grant "Accélérer le dimensionnement des systèmes énergétiques avec la différentiation automatique" from GdR SEEDS (CNRS, France).

I. Introduction

Microgrid sizing is an optimization problem whose objective is to find the optimal values of the sizing variables, which are the capacities of the power and energy storage resources. Optimality is expressed with at least one objective function and often extra objectives or constraints which are built from performance indices related to microgrid costs, environmental impact, load serving or penetration of renewable sources. All these factors must be evaluated on the project lifecycle to include the maintenance and replacements of components. The microgrid sizing problem has been studied for some decades and several classical approaches have been consolidated into convenient software packages such as HOMER [1] or DER-CAM [2]. This means that the simplest microgrid design tasks can be considered as solved problems (i.e. optimized in seconds to minutes with HOMER). By "simple task" we mean optimizing a few components using a 1-year long hourly simulation of power balance on a single bus. Still, there is an interest in finding ever more performant sizing methods in order to tackle more complex cases. Complexity comes when optimizing more than a handful of components, or components with several parameters (e.g the orientation of PV panels which is generally considered as fixed), but also when the sizing optimization is formulated as a multistage stochastic problem to mitigate long term uncertainty (like fuel price over 25 years) as in Fioriti *et al.* [3]. This latter work, which is very interesting from a theoretical point of view, has a limited applicability due to the reported solving time of 70 hours. Finally, the closed-source nature of programs like HOMER prevent reusing and expanding its existing parts to adapt to new design settings. Accelerating the pace of progress in the domain calls for more open source tools.

A. Optimization approaches

There are many approaches to solve the sizing optimization problem, but they can be divided into mathematical programming (MP) and black-box optimization (BB)[1]. The MP approach consists of formulating the microgrid sizing problem algebraically, e.g., with a Mixed Integer Linear Program (MILP) model. In the BB approach, the microgrid behavior is described inside a simulator that receives as inputs the sizing variables and returns as outputs the performance indices which are used as objective and constraints functions of the optimization problem. DER-CAM [2] is a good representative of the MP approach while HOMER [1] is perhaps the most famous simulation-based, i.e. BB, sizing tool.

MP is generally formulated with modeling languages dedicated to mathetical optimization, e.g. AMPL or GAMS, or dedicated libraries, e.g. YALMIP, JuMP, Pyomo. One of the MP advantages is that the problem can be passed to reliable optimization solvers, often with fast and guaranteed convergence properties. Nevertheless, it is necessary to make several simplifications in the models of power sources and energy storage systems, e.g. linearizations, to stay in the scope of the dedicated language/library and to get the best convergence. Also, the daily operation is often optimized in an anticipative manner, i.e., disregarding the uncertainty of hourly inputs such as load or solar production, because the optimizer has access to the entire time series at once. All this can lead to a sizing that does not meet the requirements of a microgrid in practice.

The BB approach allows models which can be way more physically realistic. Indeed, it uses a temporal simulation code written with the full freedom of numerical computing

[1]It is a black-box optimization from the point of view of the optimization algorithm, although the cost function can be fully open source.

languages. Also, implementing a non-anticipative operational control is easy, e.g. the Load Following strategy of HOMER [1], [4]. However, the optimization must be solved by *gradient-free* methods, for example Nelder-Mead or evolutionary algorithms [5], which could have poor convergence speed. Indeed, the objective and constraint functions are generally too complex to allow deriving their gradient by hand and using a finite difference approximation is disregarded as too slow.

However, it is possible to accelerate black-box design problems by using *gradient-based* optimization algorithms thanks to Automatic Differentiation (AD) software packages [6]. AD tools can compute the numerically exact gradient of computer codes. While AD tools have been around for some decades [7], there is a recent increase in developping high performance AD packages, mainly because of the increase in the research on "scientific machine learning" [8]. The promise of gradient-based microgrid sizing optimization using AD is to obtain fast convergence with the physically detailed models of the classical BB gradient-free approaches. Still, the non-convexity and sometimes the discontinuity of the objective function (see §III) implies that this promise should be carefully checked in practice. This work proposes to: 1) implement an open source microgrid simulator, with a simplified HOMER-like model, using the high performance Julia language [9]; 2) use recent Julia's AD and optimization packages to implement a gradient-based microgrid sizing tool; and, 3) analyze the convergence of this method in terms of speed and reliability on an examplary microgrid sizing problem.

Section II presents the proposed AD-based sizing method while section III details the microgrid model. Finally, convergence is assessed in section IV.

II. Methodology

To implement a black-box optimization approach for microgrid sizing using gradient-based algorithms with automatic differentiation, three elements are needed:

1) a microgrid simulator;
2) an Automatic Differentiation package; and,
3) a gradient-based optimization algorithm.

The microgrid simulator is responsible for receiving the microgrid technical characteristics, as load demand, climatic data and components parameters, and returning economic and energetic indicators, accordingly to an hourly power dispatch strategy. The detailed description of the models implemented in the simulator is presented in §III.

The Automatic Differentiation package is used to evaluate efficiently the gradients of the indicators provided by the microgrid simulator. For this purpose, the AD package needs to be have access to the internal coding of the simulator. Hence, it is important to choose compatible tools.

The gradient-based optimization algorithm solves an optimization problem that is generic and formally described as

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) && \text{(1a)} \\ s.t. \quad & \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} && \text{(1b)} \\ & \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0} && \text{(1c)} \\ & \boldsymbol{x} \in \Omega \end{aligned}$$

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 microgrid simulator and the AD package interact with the optimization algorithm sending the values and gradients of the objective and constraints functions. The Fig. 1 illustrates the interaction among these three elements.

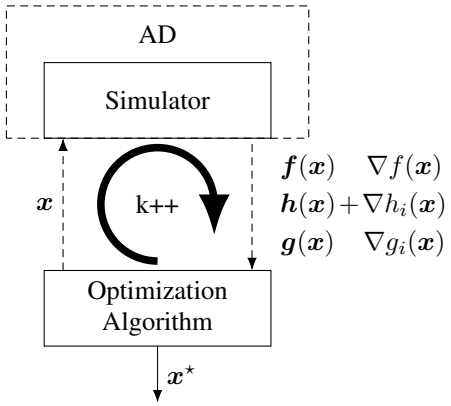

Fig. 1. Relation between optimizer and simulator.

A. Optimization convergence assessment

The functions of (1) need to be smooth, i.e., the functions need to be differentiable, to use a gradient-based algorithm. However, a more realistic microgrid model introduces discontinuities in the indicators that are used as functions in the optimization problem.

For that reason, relaxations in the modeling are usually necessary to smooth the functions. This smoothing creates a trade-off between the optimization convergence and model accuracy. Thereby, the optimization convergence and also the accuracy of the obtained results must be tested.

The process to evaluate the optimization convergence is shown in the Fig. 2, where $f(x)$ is the objective function for the original model, $f_{rlx}(x)$ is the objective function for the relaxed model, $\boldsymbol{x}^*$ is the optimal value found with the optimization of $f(x)$ and $\boldsymbol{x}^*_{rlx}$ is the optimal value found with the optimization of $f_{rlx}(x)$. Many initial points are chosen to perform the optimizations. For each one of this points, the optimization with the original model is executed, as well as the optimization of the model with relaxations implemented. The results of both process are saved for a subsequent analysis. The optimal points $\boldsymbol{x}^*_{rlx}$ found with the relaxed model objective function are reinserted in the microgrid simulator to evaluate the indicators values for the original modeling.

B. Employed tools

This work proposes using the Julia language's packages to study the feasibility of the presented methodology. The chosen AD package is the *ForwardDiff.jl*, because it usually presents better speed and accuracy than non-AD algorithms [10].

The gradient-based optimization algorithm used is a sequential quadratic programming one, more specifically the SLSQP from the NLopt Julia module [11].

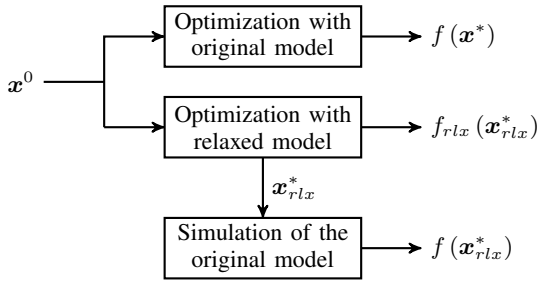

Fig. 2. Methodology for optimization convergence assessment.

The microgrid simulator is completely developed in the Julia language [12], which allows a proper integration with the AD and optimization Julia packages.

III. Microgrid simulator

This section presents the components' models and some of the economic and energetic indicators that compose the open source simulator used in this work.

The total simulation steps that happens during one year is given by $T = 365 \cdot \frac{24}{\Delta t}$, where Δt is the time-step in hours. At each one of this steps, the PV and DG production, BT operation and evaluation of its limits are accounted. Also, many operation variables and indicators, for further use in the economic and operation analysis.

A. Power dispatch strategy

In the microgrid simulator the power dispatch to meet the load $P_{load}(t)$ is defined by a rule-based strategy with order

1) Photovoltaic
2) Battery
3) Diesel generator

prioritizing the nondispatchable and less pollutants sources.

The power dispatch is combined with the load-following battery charging strategy. In this approach, the battery is only charged by the photovoltaic source.

The power dispatch strategy define the $P_{BT}(t)$ and the $P_{DG}(t)$ needed by the microgrid.

B. Photovoltaic model

The PV production for each time instant ($P_{PV}(t)$), in kW, is evaluated by $P_{PV}(t) = P^{rtd}_{PV} \cdot P^{1k}_{PV}(t)$, where P^{rtd}_{PV} is the rated capacity of the PV array, in kW, $P^{1k}_{PV}(t)$ 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^{rep}_{PV} depends on its lifetime ℓ_{PV} and also the project lifetime ℓ_{proj}, and is calculated by

$$N^{rep}_{PV} = \left\lceil \frac{\ell_{proj}}{\ell_{PV}} \right\rceil - 1. \tag{2}$$

With the number of replacements, it's possible to calculate the PV's remaining life as $\ell^{rem}_{PV} = \ell_{PV} - (\ell_{proj} - \ell_{PV} \cdot N^{rep}_{PV})$.

C. Battery model

At each time step, the battery energy E_{BT} is updated according to $E_{BT}(t+1) = E_{BT}(t) - (P_{BT}(t) + \alpha\,|P_{BT}(t)|)\,\Delta t$, 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 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 energy implied superior bound is

$$P^{e,max}_{BT}(t) = \frac{E_{BT}(t) - E^{min}_{BT}}{(1+\alpha)\,\Delta t} \tag{3}$$

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

$$P^{e,min}_{BT}(t) = -\frac{E^{max}_{BT} - E_{BT}(t)}{(1-\alpha)\,\Delta t}, \tag{4}$$

where E^{max}_{BT} 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 $P^{discharge}_{BT}(t) = \min\left(P^{max}_{BT}, P^{e,max}_{BT}(t)\right)$ and the charge limit is $P^{charge}_{BT}(t) = \max\left(P^{min}_{BT}, P^{e,min}_{BT}(t)\right)$, where P^{max}_{BT} and P^{min}_{BT} are respectively the discharge and charge power limits.

For the battery type chosen, P^{min}_{BT} is numerically equal to $-E_{BT,rtd}$ and P^{max}_{BT} is numerically equal to $E_{BT,rtd}$. The maximum available energy E^{max}_{BT} is equal to the rated energy $E_{BT,rtd}$ and the minimum available energy E^{min}_{BT}, 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^{years}_{BT}, of use and also in the energy that cycles through it. The throughput lifetime is

$$\ell^{thrpt}_{BT} = \frac{2 \cdot E^{rtd}_{BT} \cdot N_{cycles}}{Q^{thrpt}_{BT}}, \tag{5}$$

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

$$Q^{thrpt}_{BT} = \sum_{t=1}^{T} |P_{BT}(t)| \cdot \Delta t. \tag{6}$$

The battery lifetime is the most restrictive of the two types, thus it's calculated as $\ell_{BT} = \min\left(\ell^{thrpt}_{BT}, \ell^{years}_{BT}\right)$.

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

$$N^{rep}_{BT} = \left\lceil \frac{\ell_{proj}}{\ell_{BT}} \right\rceil - 1 \tag{7}$$

and the remaining life is $\ell^{rem}_{BT} = \ell_{BT} - (\ell_{proj} - \ell_{BT} \cdot N^{rep}_{BT})$.

D. Diesel generator model

The DG fuel consumption in liters at each time instant ($F(t)$) is evaluated by $F_{DG}(t) = \left(F_0 \cdot P^{rtd}_{DG} + F_1 \cdot P_{DG}(t)\right) \cdot \Delta t$ where the F_0 is the fuel curve intercept coefficient in $l/hr/kW$, F_1 is the

fuel curve slope in $l/hr/kW$, P_{DG}^{rtd} is the rated capacity of the generator in kW and $P_{DG}(t)$ is the DG electrical output in kW for the time t.

The total fuel consumption in one year is

$$F_{DG}^{tot,year} = \sum_{t=1}^{T} F_{DG}(t)\,. \tag{8}$$

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

$$h_{DG}(t) = \begin{cases} 0, & P_{DG}(t) = 0 \\ \Delta t, & 0 < P_{DG}(t) \leq P_{DG}^{rtd} \end{cases} \tag{9}$$

The DG total operation hours in one year is

$$h_{DG}^{tot} = \sum_{t=1}^{T} h_{DG}(t) \tag{10}$$

and during all the project lifetime is $h_{DG}^{tot,proj} = h_{DG}^{tot} \cdot \ell_{proj}$.

However, the DG operation hours expressed in (9) adds a discontinuity in the microgrid operation model, interfering in the gradient-based algorithms convergence. Thus, a relaxation is implemented as

$$h_{DG}^{rlx}(t) = \begin{cases} \dfrac{\Delta t}{\varepsilon}\dfrac{P_{DG}(t)}{P_{DG}^{rtd}}, & P_{DG}(t) \leq \varepsilon P_{DG}^{rtd} \\ \Delta t, & \varepsilon P_{DG}^{rtd} < P_{DG}(t) \leq P_{DG}^{rtd} \end{cases} \tag{11}$$

and the total operation hours in one year can be calculated by replacing $h_{DG}(t)$ with $h_{DG}^{rlx}(t)$ in (10).

The Fig. 3 shows the curves for the DG operations hours models expressed in (9) and (11).

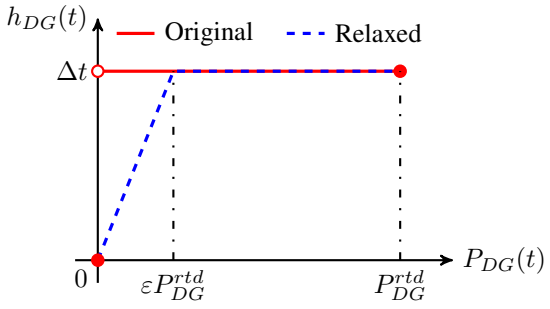

Fig. 3. Diesel generator consumption hours.

The DG number of replacements during the project is

$$N_{DG}^{rep} = \left\lceil \frac{h_{DG}^{tot,proj}}{\ell_{DG}} \right\rceil - 1 \tag{12}$$

where ℓ_{DG} is the DG lifetime in hours of operation, and the remaining life in operation hours is $\ell_{DG}^{rem} = \ell_{DG} - \left(h_{DG}^{tot,proj} - \ell_{DG} \cdot N_{DG}^{rep}\right)$.

E. Economic model

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, $c \in \mathscr{C} = \{PV, BT, DG\}$, and

$$\phi(c) = \begin{cases} P_{PV}^{rtd}, & c = PV \\ E_{BT}^{rtd}, & c = BT \\ P_{DG}^{rtd}, & c = DG \end{cases} \tag{13}$$

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

$$f_{di} = \frac{1}{(1+d)^i}, i \in \{1, 2, \ldots, \ell_{proj}\}. \tag{14}$$

We can define the summation of these factors as $\sigma_{f_d} = \sum_{y=1}^{\ell_{proj}} f_{dy}$.

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 $\mathscr{Y}_c = \{y \cdot \ell_c \mid y \in \{1, 2, \ldots, N_c^{rep}\}\}$, and with this information, the replacement factors $f_{c,i}^{rep}$ for each component are calculated using

$$f_{c,i}^{rep} = \frac{1}{(1+d)^i},\ i \in \mathscr{Y}_c. \tag{15}$$

With the factors calculated in (14) and (15), 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

$$C_c^{inv,tot} = C_c^{inv} \cdot \phi(c). \tag{16}$$

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

$$C_{PV}^{OM,tot} = C_{PV}^{OM} \cdot P_{PV}^{rtd} \cdot \sigma_{f_d} \tag{17a}$$

$$C_{BT}^{OM,tot} = C_{BT}^{OM} \cdot E_{BT}^{rtd} \cdot \sigma_{f_d} \tag{17b}$$

$$C_{DG}^{OM,tot} = C_{DG}^{OM} \cdot P_{DG}^{rtd} \cdot h_{DG}^{tot} \cdot \sigma_{f_d} + C_{DG}^{fuel,tot} \tag{17c}$$

where the total fuel cost is

$$C_{DG}^{fuel,tot} = C_{DG}^{fuel} \cdot F_{DG}^{tot,year} \cdot \sigma_{f_d}. \tag{18}$$

The present replacement cost is

$$C_c^{rep,tot} = C_c^{rep} \cdot \phi(c) \cdot \sum_{y \in \mathscr{Y}_c} f_{c,y}^{rep}. \tag{19}$$

If the components have a remaining life different of zero, a salvage value needs to be calculated. The proportional unitary salvage value is $S_c = C_c^{rep} \cdot \frac{\ell_c^{rem}}{\ell_c}$ and the total salvage value is $S_c^{tot} = S_c \cdot \phi(c) \cdot f_{d\ell_{proj}}$.

The total component present cost is

$$C_c^{tot} = C_c^{inv,tot} + C_c^{OM,tot} + C_c^{rep,tot} - S_c^{tot}. \tag{20}$$

Finally, the microgrid NPC is calculated by the summation

$$NPC = \sum_{c \in \mathscr{C}} C_c^{tot}. \tag{21}$$

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

$$LCOE = \frac{NPC}{E_{serv,lifetime}} \cdot (\sigma_{f_d})^{-1}, \tag{22}$$

where the $E_{serv,lifetime}$ is the total energy served during the project lifetime.

F. Energetic indicators

The shedding rate (SR) is the fraction of demand not supplied and it is calculated by

$$SR\,(\%) = 100 \cdot \frac{\sum_{t=1}^{T} P_{shed}(t)}{\sum_{t=1}^{T} P_{load}(t)} \tag{23}$$

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

G. Test case

The microgrid used for the methodology validation is composed of a photovoltaic system, a battery energy system and a diesel generator. The location chosen was the Ushant Island and the hourly load data are from 2016 [13]. The solar data $P_{PV}^{1k}(t)$ was obtained from the PVGIS-SARAH database [14], [15], for a PV panel with 40° slope, 0° azimuth and a loss of 14%.

The project and components parameters, technical and costs, are presented in Table I. In the carried studies, the unitary replacement costs, C_c^{rep}, are equal to the unitary capital costs, C_c^{inv}, provided by [16], [17]. In addition, the battery is considered to be discharged at the start of the project.

TABLE I
PROJECT AND COMPONENTS PARAMETERS.

Parameter	Value	Parameter	Value
ℓ_{proj}	25 yr	C_{PV}^{inv}	\$ 1200.00 $/kW$
d	5% apr	C_{PV}^{OM}	\$ 20.00 $/kW/yr$
Δt	1 h	ℓ_{PV}	25 yr
E_{BT}^{min}	0 kWh	F_0	0
α	5%	F_1	0.24 $L/h/kW_{output}$
C_{BT}^{inv}	\$ 350.00 $/kWh$	C_{DG}^{inv}	\$ 400.00 $/kW$
C_{BT}^{OM}	\$ 10.00 $/kWh/yr$	C_{DG}^{OM}	\$ 0.02 $/kW/h_{oper}$
ℓ_{BT}^{years}	15 yr	C_{DG}^{fuel}	\$ 1.00 $/l$
N_{cycles}	3000	ℓ_{DG}	15,000 h_{oper}

IV. CONVERGENCE AND PERFORMANCE ASSESSMENT

To validate the methodology, we need to evaluate the convergence by analyzing the optimization results. Using the optimization problem structure shown in (1), the optimal microgrid sizing problem has the vector of optimization variables $\boldsymbol{x} = \begin{bmatrix} P_{PV}^{rtd} & E_{BT}^{rtd} & P_{DG}^{rtd} \end{bmatrix}^T$, $NPC\,(\boldsymbol{x})$ is the objective function, and its considered one inequality constraint, $g\,(\boldsymbol{x}) = SR\,(\boldsymbol{x}) - SR^{max}$, where SR^{max} is the maximum shedding rate in percentage. The optimization variables are bounded between 10^{-8} and 10,000 kW(h).

For the convergence assessment two cases were studied. First, since we introduced a relaxation parameter ε, we look at how it influences the convergence and how it biases the result to choose a reasonable value. Second, since we suspect that the maximum shedding rate SR^{max}, which is a input from the system designer, can also affect the convergence, we test a wide array of values from 0 to 5 % of shedding rate.

In both cases, the assessment is done by running the optimization with many starting points, as described in §II-A. For a most exhaustive approach, we take starting points on a regular grid in the 3D parameter space of $\boldsymbol{x}$ (min: 0, max: 10 MW(h) for P_{PV}^{rtd} and E_{BT}^{rtd}, 2 MW for P_{DG}^{rtd}, step: 500 kW(h), which makes $21 \times 21 \times 5 = 2205$ starting points).

Further, the calculation time was also studied to analyze if this methodology could be faster than the gradient-free ones.

A. Choosing the amount of relaxation

The relaxation parameter ε is introduced to make the optimization problem continuous to ease the convergence, at the expense of biasing, i.e. underestimating, some costs, which in turn can displace the optimal sizing $\boldsymbol{x}^*$. These two aspects affect respectively the two stages of our proposed sizing method (see §II-A) and we study them separately.

1) Effect on convergence: Despite making the function continuous, the relaxation does not make it smooth or convex. Thus, the convergence of the optimization can only be assessed empirically. For choosing relaxation parameter ε, we run the gridded multistart optimization for a several amounts of relaxation between none and full (ε from 0 to 1). We conduct this analysis for one constraint level $SR^{max} = 0.01\,\%$.

For each run, we collect 2205 optimized results and we analyze the distribution of the objective and constraint functions, that are presented in Fig. 4 for $\varepsilon = 0.1$. It shows that the objective and constraint values are tightly clustered around best and maximum value respectively. However, there are some strong outliers (heavy distribution tails), that fall either in the "lower cost, unfeasible" or "higher cost, much below constraint" categories. To quantify outliers, we define tolerance thresholds: 105 % of SR^{max} for the shedding rate and 101 % of the best LCOE for the objective (tighter tolerance because the LCOE varies much less). For $\varepsilon = 0.1$, there are 0.45 % points above the objective tolerance and 1.59 % points above the constraint tolerance. Because the two case are almost always exclusive, the total amount of unacceptable solutions is the sum of both, resulting in 2.04 %.

On Fig. 5, we show that the rejection rate of optimization results generally decreases with ε. It is 100 % for $\varepsilon = 0$ (absence of convergence without relaxation) and rapidly falls to about 2 % for ε as small as 0.05. Surprisingly, the convergence decreases (up to 5 % rejection) when the relaxation becomes almost total ($\varepsilon \to 1$, which makes h_{DG} smooth, see Fig. 3). This requires further investigation.

2) Effect on biasing: The optimal values obtained with the relaxed objective function are used to recomputed the original model indicators, as shown in Fig. 2. The LCOE for the original and relaxed models for the tested ε are shown in Fig. 6, where the biasing effect of the relaxation can be observed.

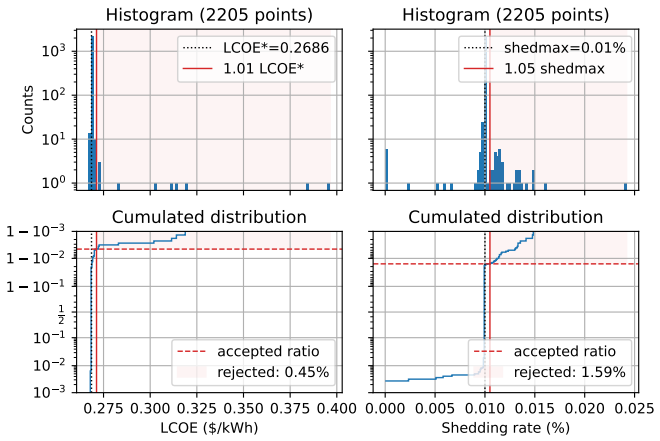

Fig. 4. Distribution of the multi-start optimization results for one test case ($\varepsilon = 0.1$, $SR^{max} = 0.01\,\%$). Distribution of the objective on the left and of the constraint on the right. Distributions are shown as log-scaled histograms on top and as cumulated distributions on the bottom, with a "logit" scale to highlight frequencies close to 0 and 1. Vertical red lines are tolerance thresholds for counting results as acceptable.

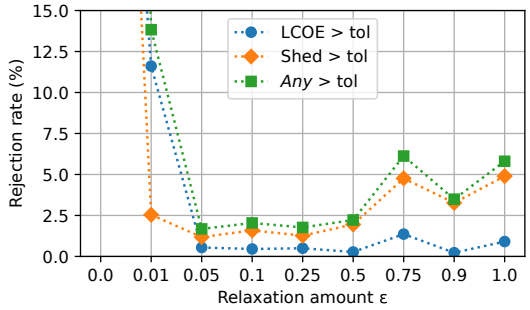

Fig. 5. Effect of the amount of relaxation ε on the rejection rate of optimization results (case $SR^{max} = 0.01\,\%$). Values for $\varepsilon = 0$ are out of bounds because all points are rejected.

As ε increases, the difference between the original and relaxed LCOE also increases. This happens because the smoothing, due to the relaxation, in the objective function conduct the optimal point away from the original optimum.

Therefore, the higher the relaxation, the worse the result found for the original model. Using this conclusion with the results presented in the previous section, a relaxation parameter ε amid 0.05 and 0.1 offers the best compromise between the effects on convergence and on biasing. For the rest of this work, we use $\varepsilon = 0.1$.

B. Robustness against the maximum shedding rate

The optimization was made for the SR^{max} equal to 0%, 0.01%, 0.10%, 0.30%, 1.00%, 3.00% and 5.00%. The Pareto front for these SR^{max} is presented in Fig. 7. This figure illustrate the trade-off between cost and quality of service, i.e. the LCOE decreases for higher SR^{max}, showing the coherence of the obtained results.

For the studied range of SR^{max}, the rejection rates for the LCOE, shedding rate or the sum of both, do not exceed 5%. The rejections for each SR^{max} are presented in the Fig. 8.

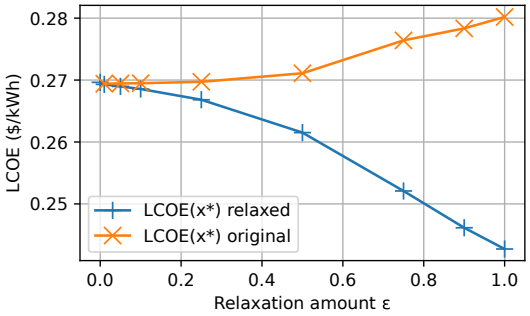

Fig. 6. Effect of the relaxation on biasing the optimal sizing (case $SR^{max} = 0.01\,\%$). The best sizing (among all multistarts) for a given amount of relaxation ε is evaluated in the unrelaxed simulator. For $\varepsilon \geq 0.25$, the relaxed optimal sizing is too far from the actual unrelaxed optimum, so that it yields a too large unrelaxed cost.

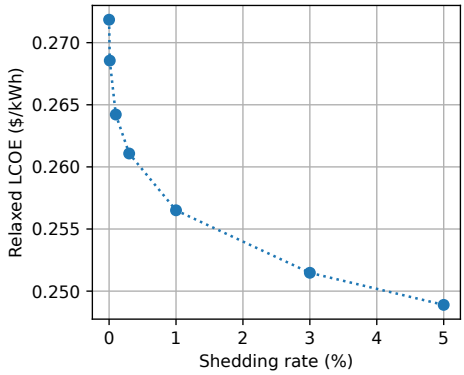

Fig. 7. Pareto front of the relaxed problem (best of all multi starts, $\varepsilon = 0.1$).

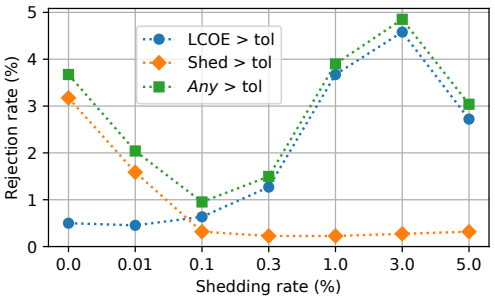

Fig. 8. Rejection rate of optimization results of the relaxed problem ($\varepsilon = 0.1$), for varying level of shedding rate constraint SR^{max}.

Even with the methodology presenting acceptable results for this SR^{max} range, we realize that the underlying structure of the optimization problem changes since the cause of rejection varies with the shedding limit. For small levels of maximum shedding rate, i.e. $SR^{max} < 0.1\%$, the rejection is mainly caused by the violation of the shedding rate tolerance. While for large shedding rate, the tolerance of LCOE is the primarily cause of rejection.

These results suggest that using a multi-start gradient-based algorithm may be suitable for solving the optimal microgrid sizing with the proposed methodology.

C. Preliminary results of computational performance

The focus of this work is more the assessment of the accuracy (e.g. empirical convergence) of the proposed microgrid sizing method than getting the shortest possible computation time. Still, the primary motivation to use AD-based optimization is indeed the promise of a shorter computational speed, compared to gradient-free methods. So we analyze the running time of our method keeping in mind that there may be room for improvement. Also, the reference problem considered here has only 3 design parameters $\boldsymbol{x}$. The computer used for these experiment is a notebook with an Intel Core i7-9750H CPU.

The running time for the simulator alone is 15 ms on average. Computing the gradient of either the objective or the constraint is only slightly longer at 22 ms, thanks to the well thought implementation of ForwardDiff.jl. Now, due to the way we have interfaced our simulator to the NLopt, each iteration of SLSQP calls separately the objective, the constraint and their gradient, which calls the simulator 4 times and makes up for ≈ 74 ms per iteration. However, there is a better interfacing to compute all these values with one ForwardDiff call, bringing this number down to 22 ms per iteration [18].

We measured the optimization running time with one particular representative starting point which converge in 43 iterations and this took on average 2.9 s. This is coherent with the simulator timings, e.g. ≈ 3.2 s. With the better interfacing mentioned above, this should go down to ≈ 0.95 s.

Optimizing a microgrid with 3 variables, for a 1 year long hourly simulation, in about 1 s sounds very promising, but we cannot claim this number due to the convergence difficulties studied here. A multi-start is needed and we need to check how many starts are needed. Still, the results from previous section suggest that only a modest number of starts is sufficient.

Also, a comparison with state-of-the-art gradien-free solver is needed. However, only few of them have *actual* support of nonlinear constraint, e.g. NOMAD.

V. Conclusions

This article proposed a new approach to solve the microgrid sizing problem. The presented methodology associates the black-box optimization with gradient-based algorithms, and the integration between these two elements was achieved through the use of Automatic Differentiation.

Although the BB method allows the use of more complex models, some relaxations need to be implemented to reduce model discontinuities and enable the use of gradient-based optimization algorithms. The effect of a relaxation was assessed from the point of view of convergence and model accuracy.

The results showed that a gradient-based optimization algorithm can find an optimal microgrid sizing with a reasonable tolerance level for the majority of the initial points scattered in the search space. The highest rejection was lower than 5%. Nevertheless, this rejection is decreased to approximately 2% with a tuning of the relaxation parameter ε.

Therefore, the application of this methodology in more complex microgrids is worthwhile. It is proposed as future work the optimal microgrid sizing and resource allocation considering the power flow equations.

References

[1] T. Lambert, P. Gilman, and P. Lilienthal, "Micropower system modeling with HOMER," in *Integration of Alternative Sources of Energy*, F. A. Farret and M. G. Simões, Eds. John Wiley & Sons, 2005.

[2] C. Marnay, G. Venkataramanan, M. Stadler, A. S. Siddiqui, R. Firestone, and B. Chandran, "Optimal Technology Selection and Operation of Commercial-Building Microgrids," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 975–982, aug 2008.

[3] D. Fioriti, D. Poli, P. Duenas-Martinez, and I. Perez-Arriaga, "Multi-year stochastic planning of off-grid microgrids subject to significant load growth uncertainty: overcoming single-year methodologies," *Electric Power Systems Research*, vol. 194, p. 107053, 2021.

[4] C. D. Barley and C. B. Winn, "Optimal dispatch strategy in remote hybrid power systems," *Solar Energy*, vol. 58, no. 4, pp. 165–179, 1996.

[5] Y. Thiaux, J. Seigneurbieux, B. Multon, and H. Ben Ahmed, "Load profile impact on the gross energy requirement of stand-alone photovoltaic systems," *Renewable Energy*, vol. 35, no. 3, pp. 602–613, Mar. 2010.

[6] P. Enciu, F. Wurtz, L. Gerbaud, and B. Delinchant, "Automatic differentiation for electromagnetic models used in optimization," *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, vol. 28, no. 5, pp. 1313–1326, sep 2009.

[7] J.-F. M. Barthelemy and L. E. Hall, "Automatic differentiation as a tool in engineering design," *Structural Optimization*, vol. 9, no. 2, pp. 76–82, apr 1995.

[8] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman, "Universal Differential Equations for Scientific Machine Learning," Jan. 2020. [Online]. Available: https://arxiv.org/abs/2001.04385

[9] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A Fresh Approach to Numerical Computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, jan 2017.

[10] J. Revels, M. Lubin, and T. Papamarkou, "Forward-Mode Automatic Differentiation in Julia," *arXiv:1607.07892 [cs.MS]*, 2016. [Online]. Available: https://arxiv.org/abs/1607.07892

[11] S. G. Johnson *et al.*, "The NLopt module for Julia," https://github.com/JuliaOpt/NLopt.jl.

[12] E. de Godoy Antunes, N. Sadou, and P. Haessig, "Microgrid.jl," https://github.com/evelisea/Microgrids.jl.

[13] EDF, "Open Data de EDF sur les îles du Ponant," https://opendata-iles-ponant.edf.fr/, Accessed: 2018.

[14] European Commission Joint Research Centre, "Photovoltaic Geographical Information System (PVGIS)," https://re.jrc.ec.europa.eu/pvg_tools/.

[15] T. Huld, R. Müller, and A. Gambardella, "A new solar radiation database for estimating PV performance in europe and africa," *Solar Energy*, vol. 86, no. 6, pp. 1803–1815, jun 2012.

[16] ADEME, "Coûts des énergies renouvelables et de récupération en France. Données 2019," https://www.ademe.fr/couts-energies-renouvelables-recuperation-France, jan 2020, Accessed: 2021-05-31.

[17] Lazard, "Levelized Cost Of Energy, Levelized Cost Of Storage, and Levelized Cost Of Hydrogen," https://www.lazard.com/perspective/lcoe2020, oct 2020, Accessed: 2021-05-31.

[18] P. Haessig, "Example of NLopt+ForwardDiff to only call the "simulator" once," https://discourse.julialang.org/t/nlopt-same-complex-clalculations-in-objetive-function-and-nonlinear-constraints/41677/15.