Comparison of evolutionary algorithms for solving risk-based energy resource management considering conditional value-at-risk analysis

José Almeida, Joao Soares, Fernando Lezama, Zita Vale, Bruno Francois

Comparison of Evolutionary Algorithms for Solving Risk-based Energy Resource Management Considering Conditional Value-at-Risk Analysis

Jose Almeida[1]., Joao Soares[1].*, Fernando Lezama[1], Zita Vale[1], and Bruno Francois[2]

1*GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, LASI - Intelligent Systems Associate LAboratory, Polytechnic of Porto, Porto, Portugal;*

2 *Univ. Lille, Arts et Metiers Institute of Technology, Centrale Lille, Junia, ULR 2697 - L2EP, F-59000 Lille, France*

**Correspondence: jan@isep.ipp.pt*

https://doi.org/10.1016/j.matcom.2023.07.010

Mathematics and Computers in Simulation
Available online 23 July 2023

Comparison of Evolutionary Algorithms for Solving Risk-based Energy Resource Management Considering Conditional Value-at-Risk Analysis

José Almeida[1·], Joao Soares[1·*], Fernando Lezama[1], Zita Vale[1], and Bruno Francois[2]

[1] *GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, LASI - Intelligent Systems Associate LAboratory, Polytechnic of Porto, Porto, Portugal;*
[2] *Univ. Lille, Arts et Metiers Institute of Technology, Centrale Lille, Junia, Lille, France;*
**Correspondence: jan@isep.ipp.pt.*

Abstract

Energy management systems must evolve due to the widespread use of distributed energy resources in modern society. In fact, with the current high penetration of renewables and other resources like electric vehicles, the challenge of managing energy resources becomes more difficult. Uncertainty and unpredictability from distributed resources open the door for unique undesirable situations, often known as extreme events. Despite the low likelihood of occurrence, such severe events represent a significant risk to an aggregator's resource management, for example. In this paper, we propose a day-ahead energy resource management model for an aggregator in a 13-bus distribution network with high penetration of distributed energy resources. In the proposed model, we consider a risk-based mechanism through the conditional value-at-risk method for risk measurement of these extreme events. Due to the complexity of the model, we also propose the use of evolutionary algorithms, a set of stochastic search algorithms, to find near-optimal solutions to the problem. Results show that implementing risk-averse strategies reduces the cost of the worst scenario and scheduling. From the tested algorithms, ReSaDE provides the solutions with the lowest cost, which is an improvement from previous work, and a reduction of around 13% in the worst-scenario costs comparing a risk-neutral approach to a risk-averse approach.

Keywords: Aggregator, Computational intelligence, Energy resource management, Evolutionary algorithms, Risk analysis, Smart grid.

NOTATION

Binaries:

x^{DG}	DG's state

Indices:

s	scenario
t	time step
i	distributed generator (DG)
e	energy storage system (ESS)
v	electric vehicle (EV)
l	load
m	wholesale electricity market
x	extreme scenarios

Parameters:

C^{DG}	generation cost of DG (€/MWh)
C^{ESS^-}	discharging cost of ESS (€/MWh)
C^{EV^-}	discharging cost of EV (€/MWh)
C^{Red}	reduction cost of load (€/MWh)
C^{imb^-}	energy not supplied cost (€/MWh)
C^{imb^+}	cost of excess DG generation (€/MWh)
MP	electricity market prices (€/MWh)
π_s	scenario probability
T	number of periods
Δt	period resolution
N_i, N_e, N_v	number of DGs/ESSs/EVs
N_l, N_m	number of loads/markets
N_s, N_x	number of scenarios/extreme scenarios
β	risk aversion factor
α	confidence level
p^{minGen}	DG's minimum active power generation (MW)
p^{maxGen}	DG's maximum active power generation (MW)

p^{maxRed}	Load's maximum active generation reduction (MW)
η^{ch}	battery's charging efficiency for EVs and ESSs
η^{disch}	battery's discharging efficiency for EVs and ESSs
p^{maxDisch}	battery's maximum active discharging power for EVs and ESSs (MW)
p^{maxCh}	battery's maximum active charging power for EVs and ESSs (MW)
E^{BatCap}	maximum battery capacity for EVs and ESSs (MWh)
E^{PMin}	minimum energy required for EVs and ESSs (MWh)
p^{maxBuy}	day-ahead maximum electricity market bid (MW)
p^{maxSell}	day-ahead maximum electricity market offer (MW)
p^{load}	forecasted day-ahead active load power consumption (MW)
$p^{\text{DG}_{\text{nd}}}$	forecasted day-ahead active renewable production (MW)

Sets and subsets:

Ω_{DG}	set of DG
$\Omega_{\text{DG}}^{\text{d}}$	subset of dispatchable DG
$\Omega_{\text{DG}}^{\text{nd}}$	subset of non-dispatchable DG

Variables:

p^{DG}	DG active power generation (MW)
p^{ESS}	ESS active discharging/charging power(MW)
p^{EV}	EV active discharging/charging power (MW)
p^{Red}	load active power reduction(MW)
p^{imb^-}	non-supplied demand active power (MW)
p^{imb^+}	DG active excess power (MW)
p^{EMarket}	active power transacted in the wholesale market (MW)
p^{Buy}	active power bid in the wholesale market (MW)
p^{Sell}	active power offered in the wholesale market (MW)
f_s^{totC}	total scenario costs (€)
f_s^{Cost}	scenario operational costs (€)

C^{EV}	costs associated with EV discharging (€)
C^{ESS}	costs associated with ESS discharging (€)
E^{stored}	energy stored in EV and ESS batteries (MWh)
f^{exC}	expected cost (€)
VaR_{α}	value-at-risk (€)
$CVaR_{\alpha}$	conditional value-at-risk (€)
OF	objective function costs (€)
B_s	bound violation penalty (€)

1. Introduction

Due to the stochastic nature of distributed energy resources (DERs), the uncertainty associated with their forecasting adds a significant level of complexity to operation problems [1]. When ignored, this uncertainty becomes a concern and may endanger the functioning of the energy chain [2]. One situation that is not commonly regarded in operation problems is the possibility of extreme events due to the variability of DERs. Even if certain occurrences are unlikely to occur, those can nonetheless have a large influence on the scheduling solution [3]. In this context, these situations can provoke, among other things, a significant rise in market pricing, a breakdown in the demand response (DR) services' communication system, an issue with the substation, or a sudden fall or rise in the amount of renewable energy produced [4]. Thus, this type of events increases the risk associated with different management problems. The risk may be assessed using tools like conditional value-at-risk ($CVaR$) and value-at-risk (VaR). Given a confidence level, the $CVaR$ mechanism simply enables finding a safer and more reliable solution than the VaR technique. In other words, $CVaR$ is helpful when the cost of the simulated scenarios exceeds a particular degree of confidence at a greater cost [5].

Both VaR and $CVaR$ risk measuring tools have seen their use majorly in the field of economics [6, 7], and their implementation has already been applied to problems in electrical power systems. In [8], a bi-level energy management system is presented in this research to assist the retail market in coordinating peer-to-peer energy trading across numerous prosumers. A stochastic programming technique using $CVaR$ is used to describe the retailer's predicted losses, taking into account the uncertainty of renewable energy. A two-stage stochastic optimization approach is presented in [9] to propose a short-term decision-making model for an electricity retailer with a battery energy storage system (BESS) and virtual bidding. The suggested approach incorporates two varieties of $CVaR$ to control the retailer's hourly and daily risks for multiple risk aversion levels. The authors of [10] propose an integrated energy system two-stage risk economic optimal model for the day ahead and intraday are developed. The

first stage aims to reduce the day-ahead operational costs, and the second stage the intraday costs. In the second stage, the $CVaR$ tool is used to enhance the objective function to evaluate the risk cost of numerous power, load, and pricing uncertainties.

The operational planning energy resource management (ERM) problem is a complex, large-scale optimization problem (i.e., with high dimensionality) due to the drastic increase in energy resources [11]. Investments in smart grid (SG) technologies, including SG communications and smart meters, are necessary to utilize DER properly. Mathematical approaches become less efficient and require large computational resources as the ERM problem's depth and complexity rise. As a result, the use of metaheuristics for ERM optimization began regularly being discussed in the literature. In fact, numerous efforts on day-ahead DER scheduling are presented in the literature [12, 13].

Multiple metaheuristics applied to energy-related problems emerge as a result of moving from the standard metaheuristics such as the genetic algorithms [14], particle swarm optimization (PSO) [15], or differential evolution (DE) [16], towards more sophisticated and efficient approaches. Improved versions of these algorithms (Hybrid-Adaptive DE (HyDE) [17], Vortex Search (VS) [18], Success-History based Adaptive DE (SHADE) [19], etc.) are also being applied to problems in the SG paradigm, including the risk-based ERM problem we are modeling in this work, achieving acceptable results as shown in [20].

In order to account for the uncertainties coming from renewable generation, load demand, electricity market pricing, and EV user behavior, this study presents a risk-based ERM model for the day ahead. The proposed methodology is based on [21]. However, the case study is different. Here, we consider three extreme scenarios, and we extend the previous work by adding multiple levels of risk-aversion, studying the variability of the VaR and $CVaR$ methods, which the previous work did not consider. We also utilize and compare different computational intelligence (CI) optimization approaches for the optimization problem outlined in this research, evaluating their performance statistically, which reference [21] failed to do. As such, the contributions of this work are summarized here:

- a day-ahead ERM formulation considering the uncertainty of load demand, renewable energy, wholesale electricity prices, and EV travel behavior.
- the integration of VaR and $CVaR$ economic risk measurement tools to address the financial risk associated with operating expenses due to technological uncertainties that might result in extreme events.
- the use of a parameter built into the formulation of the problem's objective function, to apply different levels of risk aversion for day-ahead optimization.
- implementation of CI optimization techniques through a solution-based design to deal with the computational cost of evaluating a high number of variables and probabilistic scenarios with uncertain parameters.

- initialization method to improve the performance of metaheuristics where one solution is set to the lower bounds closer to a local optima, which provides better initial results, an improvement over [21], in which all the solutions were randomly initialized.

- comparison of new and complex EAs applied in the "Competition on Evolutionary Computation in the Energy Domain: Risk-based Energy Scheduling[1]," with the algorithm used in [21].

The suggested methodologies are tested using real-world data from power and energy systems in a series of case studies, providing substantial numerical results.

The article is structured as follows: The mathematical formulation for the risk-based analysis and energy resource management is provided in Section 2. Section 3 describes the structure of the optimization approach, and Section 4 presents the case study used to test the suggested techniques. Section 5 displays the findings and results for risk-based strategies. Finally, Section 6 summarizes the significant conclusions of the planned research.

2. Risk-based ERM methodology

This section presents the mathematical model for risk measurement considering the $CVaR$ mechanism and also discusses the day-ahead scheduling taking into account total scenario cost and problem restrictions.

Figure 1 shows the flowchart of the proposed methodology. The model has as an input the total generation data (renewable and non-renewable), the load demand data, the EV and ESS requirements, and the day-ahead wholesale electricity market capacities and prices. The input data in the model was already generated and altered to include the extreme events for risk-based management (see subsection 2.3). Through the metaheuristic optimization process described in detail in section 3, the day-ahead ERM problem is solved, which is a cost minimization problem. For each value of risk-aversion (β), multiple outputs can be obtained regarding fitness costs and the corresponding terms further explained in the following section.

2.1. Risk-based formulation

Uncertain technologies include those related to renewable energy, load consumption, electricity market prices, and EV travel preferences. Extreme occurrences may arise as a result of the presence of this uncertainty. These events have a low probability of happening but a high impact on the solution, causing significant problems in the proper operation of the distribution network system.

In this work, a set of scenarios is generated to deal with the uncertainty of such resources as the demand, renewables, prices, and EV user uncertainty. As such, we formulate the total cost of each generated scenario (f_s^{totC}) as:

[1] http://www.gecad.isep.ipp.pt/ERM-competitions/2022-2/

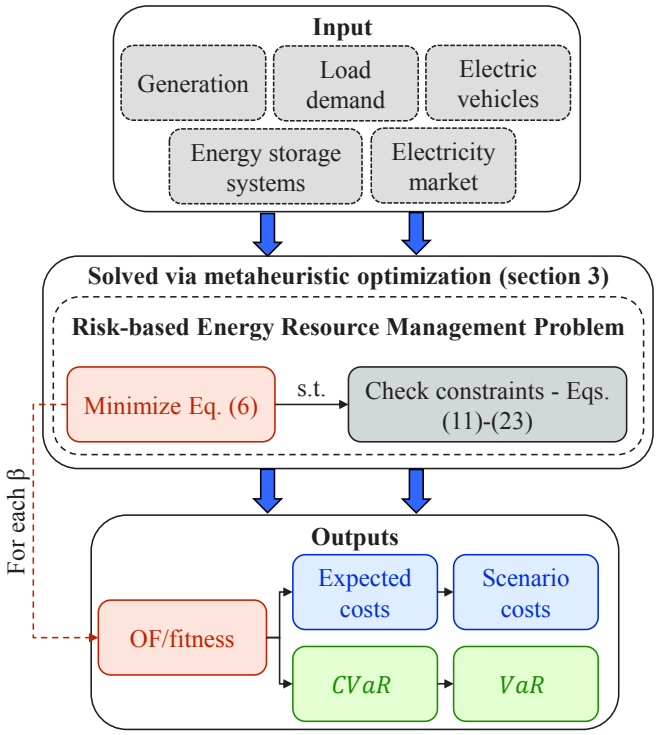

Figure 1: Flowchart of the proposed problem methodology.

$$f_s^{\text{totC}} = f_s^{\text{Cost}} + B_s \tag{1}$$

where B_s is a penalty added if any of the limit constraints is violated. We define the expected cost (f^{exC}) as:

$$f^{\text{exC}} = \sum_{s=1}^{N_s} \pi_s \times f_s^{\text{totC}} \tag{2}$$

The VaR and $CVaR$ mechanisms are introduced to evaluate the impact of extreme events. These methods estimate the financial risk associated with the operation costs. Only when the expected cost does not exceed the confidence level α, VaR can be used to assess risk. $CVaR$ is a better mechanism because it allows a more robust solution when the scenario costs exceed α. In this situation the value of α considered was 95%, a typical value for this parameter [22]. The value of VaR_α is calculated through the cumulative probability distribution function, after knowing the value of the expected cost (f^{exC}), which is calculated through a weighted sum of the total scenario costs (f_s^{totC}) and the scenario probability (π_s). With this information, $CVaR_\alpha$ can be calculated as follows:

$$CVaR_\alpha(f_s^{\text{totC}}) = VaR_\alpha(f_s^{\text{totC}}) + \frac{1}{1-\alpha}\sum_{s\in N_x}\pi_s \times (f_s^{\text{totC}} - f^{\text{exC}} - VaR_\alpha(f_s^{\text{totC}})) \tag{3}$$

where the parameter N_x is represented by the scenarios where the cost exceeds that of the expected cost in addition to the VaR_α given by:

$$f_s^{\text{totC}} \geq f^{\text{exC}} + VaR_\alpha(f_s^{\text{totC}}) \quad \forall s \in N_x \tag{4}$$

where:

$$VaR_\alpha(f_s^{\text{totC}}) = z - score(\alpha) \times std(f_s^{\text{totC}}) \tag{5}$$

$z - score$ is computed in MATLAB with α equal to 95% using the **norminv()** function.

The objective function (OF) for the risk-based that the aggregator minimizes can be given by:

$$min\ OF = f^{\text{exC}} + (\beta \times CVaR_\alpha) \tag{6}$$

where β is a risk aversion parameter that varies from 0 to 1. If this parameter is 0, the aggregator minimizes only the expected cost. In contrast, if it is 1, the aggregator has 100% risk aversion in the formulation and considers the total value of $CVaR_\alpha$.

2.2. Day-ahead scheduling formulation

The mathematical formulation of the day-ahead scheduling, taking into account the total operational costs of each scenario s (f_s^{Cost}), is given by:

$$f_s^{\text{Cost}} = \sum_{t=1}^{T} \cdot \left[\begin{array}{l} \sum\limits_{i\in\Omega_{\text{DG}}^{\text{d}}} p_{(i,t)}^{\text{DG}} \cdot C_{(i,t)}^{\text{DG}} + \\ \sum\limits_{i\in\Omega_{\text{DG}}^{\text{nd}}} p_{(i,t,s)}^{\text{DG}} \cdot C_{(i,t)}^{\text{DG}} + \\ \sum\limits_{e=1}^{N_e} C_{(e,t,s)}^{\text{ESS}} + \sum\limits_{v=1}^{N_v} C_{(v,t,s)}^{\text{EV}} + \\ \sum\limits_{l=1}^{N_l} (p_{(l,t,s)}^{\text{Red}} \cdot C_{(l,t)}^{\text{Red}} + p_{(l,t,s)}^{\text{imb}^-} \cdot C_{(l,t)}^{\text{imb}^-}) + \\ \sum\limits_{i=1}^{N_i} p_{(i,t,s)}^{\text{imb}^+} \cdot C_{(i,t)}^{\text{imb}^+} + \\ \sum\limits_{m=1}^{N_m} p_{(m,t)}^{\text{EMarket}} \cdot MP_{(m,t,s)} \end{array} \right] \cdot \Delta t \quad \forall s \tag{7}$$

where:

$$C_{(e,t,s)}^{\text{ESS}} = \begin{cases} p_{(e,t,s)}^{\text{ESS}} \cdot C_{(e,t)}^{\text{ESS}^-} & \text{if } p_{(e,t,s)}^{\text{ESS}} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$C_{(v,t,s)}^{\text{EV}} = \begin{cases} p_{(v,t,s)}^{\text{EV}} \cdot C_{(v,t)}^{\text{EV}^-} & \text{if } p_{(v,t,s)}^{\text{EV}} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$p_{(m,t)}^{\text{EMarket}} = \begin{cases} p_{(m,t)}^{\text{Buy}} & \text{if } p_{(m,t)}^{\text{EMarket}} < 0 \\ p_{(m,t)}^{\text{Sell}} & \text{if } p_{(m,t)}^{\text{EMarket}} > 0 \end{cases} \tag{10}$$

The OF is subject to multiple constraints. These constraints refer to: the power balancing constraint stipulates that the amount of generated power must equal the amount of consumed power at any given time t, as Eq. (11) shows:

$$\left[\begin{array}{l} \sum\limits_{i \in \Omega_{\text{DG}}^{\text{d}}} p_{(i,t)}^{\text{DG}} + \sum\limits_{i \in \Omega_{\text{DG}}^{\text{nd}}} p_{(i,t,s)}^{\text{DG}} + \\ \sum\limits_{l=1}^{N_l} (p_{(l,t,s)}^{\text{Red}} - p_{(l,t,s)}^{\text{load}}) + \\ \sum\limits_{e=1}^{N_e} p_{(e,t,s)}^{\text{ESS}} + \sum\limits_{v=1}^{N_v} p_{(v,t,s)}^{\text{EV}} + \sum\limits_{m=1}^{N_m} p_{(m,t)}^{\text{EMarket}} + \\ \sum\limits_{i \in \Omega_{\text{DG}}^{\text{nd}}} p_{(i,t,s)}^{\text{imb}^+} - \sum\limits_{l=1}^{N_l} p_{(l,t,s)}^{\text{imb}^-} \end{array} \right] = 0 \quad \forall s \tag{11}$$

The minimum and maximum power generation restrictions on dispatchable generation at each time t, and the forecasted renewable non-dispatchable generation contraint, given by Eqs. (12)-(13):

$$p_{(i,t)}^{\text{minGen}} \cdot x_{(i,t)}^{\text{DG}} \leq p_{(i,t)}^{\text{DG}} \leq p_{(i,t)}^{\text{maxGen}} \cdot x_{(i,t)}^{\text{DG}} \quad \forall i \in \Omega_{\text{DG}}^{\text{d}}, \forall t \tag{12}$$

$$p_{(i,t,s)}^{\text{DG}} = p_{(i,t,s)}^{\text{DG}_{\text{nd}}} \cdot x_{(i,t)}^{\text{DG}_{\text{nd}}} \quad \forall i \in \Omega_{\text{DG}}^{\text{nd}}, \forall t \tag{13}$$

Eq. (13) represents the DR limitation imposed by the maximum amount of load reduction l in period t:

$$p_{(l,t,s)}^{\text{Red}} \leq P_{(l,t)}^{\text{maxRed}} \quad \forall l, \forall t, \forall s \tag{14}$$

The battery balance of each energy storage system (ESS) is described by Eq. (14):

$$E_{(e,t,s)}^{\text{stored}} = E_{(e,t-1,s)}^{\text{stored}} + \eta_{(e)}^{\text{ch}} \cdot p_{(e,t,s)}^{\text{ESS}} \cdot \Delta t - \frac{1}{\eta_{(e)}^{\text{disch}}} \cdot p_{(e,t,s)}^{\text{ESS}} \cdot \Delta t \qquad \forall e, \forall t, \forall s \tag{15}$$

The maximum charge and discharge restrictions for each ESS, the battery capacity limit, and the minimal amount of energy that must be guaranteed at the end of period t are given by Eqs. (16)-(18):

$$-p_{(e,t)}^{\text{maxDisch}} \leq p_{(e,t,s)}^{\text{ESS}} \leq p_{(e,t)}^{\text{maxCh}} \quad \forall e, \forall t, \forall s \tag{16}$$

$$E_{(e,t,s)}^{\text{stored}} \leq E_{(e)}^{\text{BatCap}} \quad \forall e, \forall t, \forall s \tag{17}$$

$$E_{(e,t,s)}^{\text{stored}} \geq E_{(e,t)}^{\text{PMin}} \quad \forall e, \forall t, \forall s \tag{18}$$

Similar to the ESS, the balance of each EV battery can be formulated as in Eq. (19), since the set of EVs is viewed as a group of loads that stand in for virtual batteries. However, EVs have several restrictions and requirements that ESSs do not. For instance, EVs have unique journey requirements depending on user choices and are stationed at designated network points. These requirements are connected to the uncertainties surrounding EV travel behavior as well. While these criteria are developed as an input to the problem, the restrictions on EVs remain the same.

$$E^{\text{stored}}_{(v,t,s)} = E^{\text{stored}}_{(v,t-1,s)} + \eta^{\text{ch}}_{(v)} \cdot p^{\text{EV}}_{(v,t,s)} \cdot \Delta t - \frac{1}{\eta^{\text{disch}}_{(v)}} \cdot p^{\text{EV}}_{(v,t,s)} \cdot \Delta t \qquad \forall v, \forall t, \forall s \quad (19)$$

The maximum charge and discharge restrictions for each EV, the battery capacity limit and minimal amount of energy that must be guaranteed at the end of period t are given by Eqs. (20)-(22):

$$-p^{\text{maxDisch}}_{(v,t)} \leq p^{\text{EV}}_{(v,t,s)} \leq p^{\text{maxCh}}_{(v,t)} \quad \forall v, \forall t, \forall s \quad (20)$$

$$E^{\text{stored}}_{(v,t,s)} \leq E^{\text{BatCap}}_{(v)} \quad \forall v, \forall t, \forall s \quad (21)$$

$$E^{\text{stored}}_{(v,t,s)} \geq E^{\text{PMin}}_{(v,t)} \quad \forall v, \forall t, \forall s \quad (22)$$

The offer and bidding limits in the electricity market, can be expressed by Eq. (23) as follows:

$$-p^{\text{maxBuy}}_{(m,t)} \leq p^{\text{EMarket}}_{(m,t)} \leq p^{\text{maxSell}}_{(m,t)} \quad \forall m, \forall t \quad (23)$$

To avoid the use of binary variables, for the EV and ESS state of charging and discharging, variables P^{ESS}, P^{EV} take a negative value when the EVs and ESSs are discharging and a positive value when they are charging, guaranteeing a non-simultaneity. The same approach is used for the market offer and bid status through the P^{EMarket} variable, where bidding in the wholesale electricity market is given by a negative value, and positive values give market offerings.

2.3. Uncertainty

In the model under consideration, the aggregator must cope with uncertainty resulting from various factors, such as the unpredictable driving and charging behaviors of EV customers, changes in market pricing, and unpredictable renewable energy supply, for example. The aggregator cannot assure the success of the decision-making process because the precise result of these resources is practically impossible to foresee (because of the unpredictability of these factors). As a result, the suggested solution uses a scenario-based optimization strategy to consider the uncertainties related to the given resources. The initial set of scenarios is generated via Monte Carlo Simulation (MCS), as Figure 2 shows, to forecast probable results.

A large set of scenarios is initially created (5,000 scenarios) through random sampling using the Gaussian probability distribution function. But to reduce computational effort, this set of scenarios is reduced using a fast backward-forward method in [23]. This reduction is achieved by grouping scenarios with similar characteristics while excluding those with a low probability of occurring. Consequently, a scenario subset that corresponds to a probability measure is created close to the initial distribution. Reducing the problem's magnitude is the scenario reduction's main goal, corresponding to faster processing times.

Additionally, another reduction by a random scenario selection is processed to this first reduction, so the computation effort and time are reduced even further. It is important to note that, as a result of this reduction, it is hard to prevent some imprecision in the final scenarios, even while the statistical features of the original data set are preserved. The resulting scenario subset is then altered by incorporating three different extreme scenarios. These scenarios, compared to the previously computed, have a low probability of occurrence. Still, if they occur, their impact on the final solution can be substantial and impose extreme expenses on the aggregator. In this work, we have manually generated these extreme events based on problems that might occur in the day-ahead operation and present a risk for the aggregator, as Figure 2 shows.

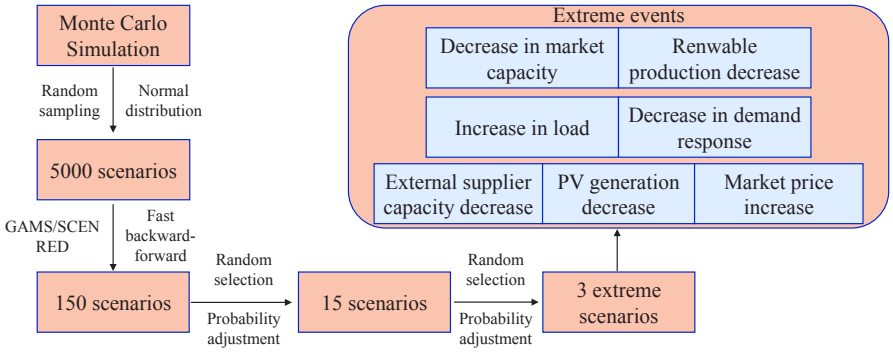

Figure 2: Scenario generation methodology.

3. Optimization

This section briefly describes each EA, solution encoding, and fitness evaluation process, which are typical for CI optimization.

3.1. Evolutionary algorithms

Multiple state-of-the-art EAs are used in this problem together with DE, namely: Hybrid-Adaptive Differential Evolution (HyDE) and the three first EAs classified in this years' competition, namely Restart-assisted Self-adaptive DE (ReSaDE), Ring Cellular Encode-Decode UMDA (RCEDUMDA) [24] and

Chaotic Levy Hybrid RCEDUMDA (CLHC2RCEDUMDA) based on [25]. These following algorithms are proposed so we can compare with CUMDAN-Cauchy, the algorithm utilized in [21].

3.1.1. DE

The DE algorithm with the mutation strategy *"DE/rand/1/bin"* was applied to the proposed optimization problem with binomial crossover. The implementation of the algorithm is described in Algorithm 1. Initially, the algorithm, after defining the necessary parameters, generates one solution with the lower bounds and the remaining solutions are generated randomly between the upper and lower bounds, with represents the target vector ($\vec{x}_i$). This target vector is evaluated, and the best solution is stored as $\vec{x}_{best}$.

In the iterative process of the algorithm, three random individuals are selected from the generated solution, and a mutation strategy is applied, generating a donor vector. The binomial recombination is then applied, generating the trial vector $\vec{u}_{i,it}$. We then verify the boundary constraints, and if they are violated, the variables are updated to the minimum or maximum bounds accordingly. Then the trial vector is evaluated, and elitism selection is applied. Finally, the best solution $\vec{x}_{best}$ is updated with the minimum value of $\vec{x}_{i,it}$.

Algorithm 1 Standard Differential Evolution

Define algorithm parameters Pop, $maxIt$, F, Cr
Initialize $maxUp$ and $maxLow$
Generate one solution as $maxLow$ and the rest randomly between $maxUp$ and $maxLow$ ($\vec{x}_i$)
Evaluate initial solution
Store best individual as the $\vec{x}_{best}$
$it \leftarrow 1$
while $it \leq maxIt$ **do**
    for all Pop **do**
        Select three random individuals $\vec{x}_{r1,it} \neq \vec{x}_{r2,it} \neq \vec{x}_{r3,it} \in \vec{x}_{i,it}$
        Apply mutation strategy $\vec{m}_{i,it} = \vec{x}_{r1,it} + F(\vec{x}_{r2,it} - \vec{x}_{r3,it})$
        Apply binomial recombination (generate trial vector $\vec{u}_{i,it}$)
        Verify boundary control
        Evaluate $\vec{u}_{i,it}$
        Apply elitism selection
        if $\vec{u}_{i,it} < \vec{x}_{i,it}$ **then**
            $\vec{x}_{i,it+1} = \vec{u}_{i,it}$
            Update solution
        end if
    end for
    Store $\vec{x}_{best} \leftarrow min(\vec{x}_{i,it})$
    $it \leftarrow it + 1$
end while

3.1.2. HyDE

Self-adaptive DE versions do not require parameter adjustment and frequently exhibit adequate performance for many types of problems. The HyDE algorithm uses the mutation operator *"DE/target-to-perturbed_best/1"* given by Eq. (24). Initially, the process made is similar to the standard DE (Algorithm 2), where the inputs are defined. Different from the standard DE, HyDE in the iterative process creates three different scaling factors (F_i^1, F_i^2, F_i^3), where they are updated at the end of the iterative process, following a self-adaptive mechanism (step 17).

$$\vec{m}_{i,it} = \vec{x}_{i,it} + F_i^1[(\vec{x}_{best} \cdot \mathcal{N}(F_i^2, 1) - \vec{x}_{i,it})] + F_i^3[\vec{x}_{r1,it} - \vec{x}_{r2,it}] \tag{24}$$

The mutation strategy applied is given by Eq. (24), where a random perturbation factor ($\mathcal{N}(F_i^2, 1)$) is applied to the best solution found (x_{best}). The remaining steps of the algorithm are similar to the previous algorithm. Finally, the best global solution is stored as x_{best}.

Algorithm 2 Hybrid-Adaptive Differential Evolution, adapted from [26]

1: Define algorithm parameters Pop, $maxIt$, F_i^1, F_i^2, F_i^3 and Cr_i
2: Initialize $maxUp$ and $maxLow$
3: Generate one solution as $maxLow$ and the rest randomly between $maxUp$ and $maxLow$ ($\vec{x}_i$)
4: Evaluate initial solution
5: Store best individual as the $\vec{x}_{best}$
6: $it \leftarrow 1$
7: **while** $it \leq maxIt$ **do**
8: Create F_i^1, F_i^2, F_i^3 and Cr_i $\forall i \in \vec{x}_{i,it}$
9: **for** all Pop **do**
10: Select two random individuals $\vec{x}_{r1,it} \neq \vec{x}_{r2,it} \in \vec{x}_{i,it}$
11: Apply mutation strategy in Eq. (24)
12: Apply binomial recombination (same as Algorithm 1)
13: Verify boundary control
14: Evaluate new solution (same as Algorithm 1)
15: Apply elitism selection (same as Algorithm 1)
16: **end for**
17: Update F_i^1, F_i^2, F_i^3 and Cr_i $\forall i \in \vec{x}_{i,it}$
18: Store $\vec{x}_{best} \leftarrow min(\vec{x}_{i,it})$
19: $it \leftarrow it + 1$
20: **end while**

3.1.3. ReSaDE

The ReSaDE algorithm, similar to HyDE, is a self-adaptive version of DE, and the process is described by Algorithm 3. This algorithm initially performs a soft group of variables according to [27]. Then, new upper and lower bounds are initialized for each group based on the grouped variables, the population is

generated and evaluated, and the best-grouped solution is stored as x_{GBest}. In the iterative process, the algorithm runs for each group, and a given number of iterations, a modified self-adaptive DE (SaDE) based on [28] with no restart mechanisms.

Then, the current best group of variables is stored, and after iteratively going through each group, the groups are sorted, and the best group is selected. The algorithm then proceeds to run the standard DE for a given number of iterations as a "warm-start," so the trust region of the algorithm is adjusted in the search space. Finally, the SaDE is again run, but in this case, with a couple o restart loops if the algorithm gets stuck in the local optima. That is if there is a stagnation in the fitness value for a given number of iterations. As a result, the best solution is stored at the end of the procedure as x_{best}.

Algorithm 3 Restart-assisted Self-adaptive Differential Evolution

Define algorithm parameters Pop, $maxIt$, F, and Cr
Initialize $maxUp$ and $maxLow$
for all D **do**
    Perform variable soft-grouping (G_j)
end for
for all G_j **do**
    Initialize $maxG_{Up}$ and $maxG_{Low}$
    Generate one grouped solution as $maxG_{Low}$ and the rest randomly between $maxG_{Up}$ and $maxG_{Low}$
    Evaluate grouped solution
    Store best grouped solution as x_{GBest}
    Define iterations for self-adaptive DE (max_itSaDE)
    $it \leftarrow 1$
    while $it \leq max_itSaDE$ **do**
        Run self-adaptive DE
        Perform elitist selection
    end while
    Store current best as x_{GBest}
end for
Sort grouped variables according to x_{GBest}
Select most effective group of variables
Define iterations for standard DE (max_itDE)
while $it \leq max_itDE$ **do**
    Run DE as described in Algorithm 1
end while
Adjust trust region
Run self-adaptive DE with restart loops for remaining iterations
Store best solution as x_{best}

3.1.4. RCEDUMDA

A cellular estimation of distribution algorithm is known as RCEDUMDA (Algorithm 4) [25]. In this algorithm, a varied but encouraging sampling of the search space is the initial population that contains one of the solutions initialized with the variable's lower limits. It divides the global population into several tiny sub-populations using a ring structure. Additionally, it divides the continuous data into categorical variables (codes) during each neighborhood's reproductive cycle using an encoding technique, reducing the search space. Then the encoded solution is estimated and scaled (steps 12 and 13), and a new solution is generated. After this process, a decode needs to occur to transform the categorical variables into continuous variables, and this solution is inserted in an $auxPop$, which then replaces the current (Pop). Elitism selection is then performed, including the best individuals. Finally, the best solution in Pop is stored as the global solution x_{best}.

Algorithm 4 Ring Cellular Encode-Decode UMDA, adapted from [25]

1: Define algorithm parameters Pop, $maxIt$, c, m, l, s, r, α and k
2: Initialize $maxUp$ and $maxLow$
3: Generate one solution as $maxLow$ and the rest randomly between $maxUp$ and $maxLow$
4: Evaluate initial solution
5: Store best individual as the $\vec{x}_{best}$
6: $it \leftarrow 1$
7: **while** $it \leq maxIt$ **do**
8: Select globally l elitist individuals
9: **for** all $cell \in Pop$ **do**
10: Get the m best individuals in $neighborhood(cell, r)$
11: Encode solution (continuous variables to categorical variables)
12: Estimate through distribution $\prod_{i=1}^{l} p(x_i)$ the best encoded individuals
13: Scale according to α the best encoded individuals
14: Generate c new individuals according to scaled individuals
15: Decode new solution (categorical variables to continuous variables)
16: Insert decoded solution in the same cell of $auxPop$
17: **end for**
18: Replace Pop with the $auxPop$
19: Perform elitist selection
20: $it \leftarrow it + 1$
21: **end while**
22: Store $\vec{x}_{best} \leftarrow min(Pop)$

3.1.5. CLHC2RCEDUMDA

The CLHC2RCEDUMDA is a modified version of the HC2RCEDUMDA using chaotic Lévy flight distribution [29]. Here the algorithm initializes one

individual with the variables' lower bounds. The HC2RCEDUMDA algorithm uses discrete hill climbing to reduce the search space by encoding and decoding variables using a discrete step to go through the number of codes for a given variable. After, the algorithm uses the RCEDUMDA procedure described previously in Algorithm 4. Finally, the Levy distribution is applied in the continuous hill climbing, where the step used is calculated using Eq. (25):

$$step = \frac{rand(1, D) \times \sigma}{|rand(1, D)|^{\frac{1}{\beta}}} \tag{25}$$

where D represents the problem dimension, that is, the number of variables and σ is given by the following equation:

$$\sigma = \left[\frac{\Gamma(1 + 2\lambda) sin(\Pi\lambda)}{\Gamma(\frac{(1+\lambda)}{2}) 2\lambda^{(\lambda-3)}} \right]^{\frac{1}{\lambda}} \tag{26}$$

where λ represents the Lévy coefficient, and the Chaotic Lévy distribution applied in this algorithm can be formulated as in Eq. (27), which uses the Gaussian map's randomly generated number in the Lévy distribution, the Chaotic equation is employed to increase the variety and quality of the new population, which in turn enhances the algorithm's capacity to do a global search.

$$\begin{aligned} CLrand &= rand(1, D) \\ CLpos &= \frac{(\frac{1}{CLrand}) - (floor(\frac{1}{CLrand}))}{2} \\ CLpdf &= unifrnd(0.2, 0.2, 1) \times step \times CLpos \end{aligned} \tag{27}$$

Algorithm 5 Chaotic Levy Hybrid Ring Cellular Encode-Decode UMDA

1: Define algorithm parameters Pop, $maxIt$, c, m, l, s, r, α and k
2: Initialize $maxUp$ and $maxLow$
3: Generate one solution as $maxLow$ and the rest randomly between $maxUp$ and $maxLow$
4: Evaluate initial solution
5: Store best individual as the $\vec{x}_{best}$
6: Discrete hill climbing (encoding and decoding based on $\vec{x}_{best}$)
7: Return best
8: Apply RCEDUMDA algorithm (Algorithm 4)
9: Apply continuous hill climbing (Lévy flight distribution)
10: Store best solution as $\vec{x}_{best}$

3.1.6. Particle Swarm Optimization

PSO is a population-based optimization technique that draws inspiration from the social behavior of fish schooling and bird flocking [15]. The swarm (a group of particles) travels around a search space for the best solution, and algorithm 6 describes the process used in this work. The parameters are defined

initially and as described in the previous algorithms, and the variable bounds are initiated for particle positions. Then for PSO, the velocity minimum and maximum values also need to be set according to:

$$\vec{v}_i^{max} = v_f \cdot (maxUp - maxLow) \tag{28}$$

$$\vec{v}_i^{min} = -\vec{v}_i^{max} \tag{29}$$

where v_f is a velocity factor used to regulate the particle velocity, and $maxUp$ and $maxLow$ are the upper and lower variable bounds. Also, like in the previous algorithms, one of the solutions is set to the lower bounds, and for the PSO, the initial particle velocity is initialized. After the initial set of solutions is evaluated, and the best fitness is stored. Entering the iterative process, the algorithm for each iteration updates the inertia weight through a dumping ratio given by:

$$w = w_{max} - \frac{w_{max} - w_{min}}{maxIt} \cdot it \tag{30}$$

w_{max} and w_{min} are the maximum and minimum limits set for the inertia, it is the current algorithm iteration, and $maxIt$ is the maximum number of iterations. After, for all the population size, the particle velocity and particle position are updated, as the following equations describe:

$$\vec{v}_{i,it+1} = w\vec{v}_{i,it} + c_p r_p(\vec{x}^p_{i,it} - \vec{x}_{i,it}) + c_g r_g(\vec{x}^g_{i,it} - \vec{x}_{i,it}) \tag{31}$$

$$\vec{x}_{i,it+1} = \vec{x}_{i,it} + \vec{v}_{i,it+1} \tag{32}$$

where w is the inertia weight, c_p and c_g are the personal and global acceleration coefficients, and r_p and r_g are two random coefficients that vary between [0,1]. The personal and global best particle positions are described in $\vec{x}^p_{i,it}$ and $\vec{x}^g_{i,it}$, respectively. Following this process, a boundary control needs to be set for particle velocity and position, and then the newly generated particles are evaluated. Finally, the particle with the lowest fitness value is stored as the best individual.

3.1.7. Vortex Search

The VS method is a single-solution based metaheuristic for resolving bound-constrained global optimization problems [18]. In the case of VS, only the Pop and $maxIt$ parameters need to be set as algorithm 7 describes. Before entering the iterative process, the remaining process is similar to the previous algorithms. When entering the iterative process, a_{it} sample values need to be generated between [0,1] to ensure search space coverage, which is given by the following equation:

$$a_{it} = \frac{it}{maxIt} \tag{33}$$

Algorithm 6 Particle Swarm Optimization

Define algorithm parameters Pop, $maxIt$, w_{min}, w_{max}, c_p, c_g, v_f
Initialize $maxUp$ and $maxLow$
Calculate v_i^{max} and v_i^{min} based on Eqs. (28)-(29)
Generate one solution as $maxLow$ and the rest randomly between $maxUp$ and $maxLow$
Generate initial velocity between v_i^{min} and v_i^{max}
Evaluate initial solution
Store best individual as the $\vec{x}_{best}$
$it \leftarrow 1$
while $it \leq maxIt$ **do**
    Update inertia weight via Eq. (30)
    for all Pop **do**
        Update particle velocity via Eq. (31)
        Update particle position via Eq. (32)
        Verify boundary control
        Evaluate new solution ($\vec{x}_{i,it+1}$)
    end for
    Store $\vec{x}_{best} \leftarrow min(\vec{x}_{i,it+1})$
    $it \leftarrow it + 1$
end while

After the initial circle radius needs to be set so candidate solutions can be generated and is demonstrated by the following:

$$\mu = \frac{maxUp - maxLow}{2} \tag{34}$$

$$r_{it} = \mu \cdot \frac{1}{0.1} \cdot gammaincinv(0.1, a_{it}) \tag{35}$$

Following this process, a set of candidate solutions is generated using a Gaussian probability distribution around the best solution. The final processes of the algorithm are also similar to the previously demonstrated metaheuristics.

3.1.8. Success-history based Adaptive Differential Evolution

SHADE is an algorithm that uses a parameter adaptation method based on a historical record of effective parameter adjustments [19]. The process used is described in algorithm 8, where initially algorithm parameters are defined. In this case, and different from HyDE, for example, F and Cr are parameters that will be recorded in memory for H entries, which are designed as M_{Cr} and M_F, where H represents the memory size. Additionally, an archive A is also set to store problem solutions. After, the algorithm follows similar processes as the precious algorithms described. Entering the iterative process, a random entry (r_i) is selected, which determines the position of memory to update $Cr_{i,it}$ and

Algorithm 7 Vortex Search, adapted from [18]

Define algorithm parameters Pop, $maxIt$
Initialize $maxUp$ and $maxLow$
Generate one solution as $maxLow$ and the rest randomly between $maxUp$ and $maxLow$
Evaluate initial solution
Store best individual as the $\vec{x}_{best}$
$it \leftarrow 1$
while $it \leq maxIt$ **do**
    Sample a values within [0,1] by using Eq. (33)
    Calcualte initial circle radius using Eqs. (34)-(35)
    Generate Pop candidate solutions ($\vec{x}_{i,it+1}$) using Gaussian distribution
    Verify boundary control
    Evaluate new solution ($\vec{x}_{i,it+1}$)
    Store $\vec{x}_{best} \leftarrow min(\vec{x}_{i,it+1})$
    $it \leftarrow it + 1$
end while

$F_{i,it}$. Following this process, a random value $p_{i,it}$ is generated, where $p_{min} = Pop/2$ and the trial vector is generated according to:

$$\vec{m}_{i,it} = \vec{x}_{i,it} + F_i \cdot (\vec{x}_{pbest,it} - \vec{x}_{i,it}) + F_i \cdot (\vec{x}_{r1,it} - \vec{x}_{r2,it}) \tag{36}$$

where $\vec{x}_{pbest,it}$ is an individual randomly selected according to $p_{i,it}$. The trial vector is then evaluated and the best solutions are updated and stored in the archive A, where the archive size does not exceed the total population size, otherwise randomly selected individuals need to be eliminated, and the memory is not updated when all members of generation it fail to provide a trial vector that is better than the parent solution.

3.2. Solution generation

Each of the suggested EAs first produces a population of solutions with one individual as the lower variables bounds and the remaining randomly within the given variable boundaries, as shown in Figure 3, and as specified for the guidelines of the competition. In this situation, we initialize one solution with the lower bounds so we guarantee a better initial result for the EA which tends to a better overall result. For each of the 24 periods, each collection of variables is successively repeated. Observe that the remaining variables are continuous and vary according to the set constraints, except for the generators' status, which is represented by a binary variable (0 - not connected to the grid, 1 - connected to the grid).

The scheduling problem in question includes 13,680 variables per solution, divided into 570 variables every period, with 21 variables forming the active power and status of the generators (N_i). A total of 500 EVs (N_v) were con-

Algorithm 8 Success-History Based Adaptive DE, adapted from [19]

Define algorithm parameters Pop, $maxIt$, M_F, M_{Cr}
Set memory size with H entries
Set archive A
Initialize $maxUp$ and $maxLow$
Generate one solution as $maxLow$ and the rest randomly between $maxUp$ and $maxLow$
Evaluate initial solution
Store best individual as the $\vec{x}_{best}$
$it \leftarrow 1$
$k \leftarrow 1$
while $it \leq maxIt$ **do**
    Initialize S_{Cr}, S_F
    for all Pop **do**
        Select randomly between [1,H] (r_i)
        $Cr_{i,it} = randn_i(M_{Cr,r_i}, 0.1)$
        $F_{i,it} = randc_i(M_{F,r_i}, 0.1)$
        $p_{i,it} = rand[p_{min}, 0.2]$
        Generate trial vector ($\vec{u}_{i,it}$) using the mutation strategy in Eq. (36)
    end for
    Evaluate trial vector
    for all Pop **do**
        if $f(\vec{u}_{i,it}) \leq f(\vec{x}_{i,it})$ **then**
            $\vec{x}_{i,it+1} = \vec{u}_{i,it}$
        end if
        if $f(\vec{u}_{i,it}) < f(\vec{x}_{i,it})$ **then**
            Update solution in A
            Update $Cr_{i,it}$ and $F_{i,it}$
        end if
    end for
    If archive size exceeds $|A|$, select random individuals to delete ($|A| \leq |Pop|$)
    if $S_{Cr}, S_F \neq 0$ **then**
        Update $M_{Cr,k}, M_{F,k}$ based on S_{Cr}, S_F
        $k \leftarrow k + 1$, where $k < H$, otherwise $k \leftarrow 1$
    end if
    $it \leftarrow it + 1$
end while

sidered, with 25 curtailable load types (N_l), two different ESSs (N_e), and one market (N_m).

3.3. Fitness evaluation

The risk-based scheduling methodology's optimization approach seeks to reduce the OF cost in Eq. (6). The database comprising all 15 created scenarios,

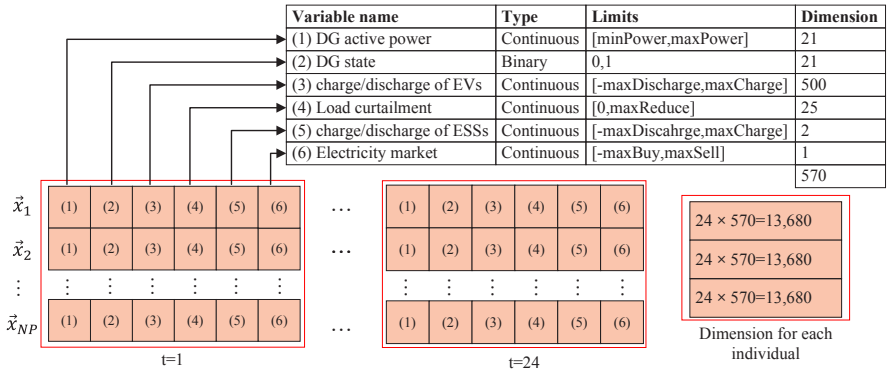

Figure 3: Proposed solution encoding.

including the extreme events generated as previously detailed in Section 2.3, is loaded to begin the fitness function evaluation done by each EA. The value of the risk-aversion-controlling variable (β) is likewise set. Then, for each scenario, variable bounds are updated according to the scenario data because load consumption and renewable generation are scenario-dependent, and if not corrected, costs wrongly associated with variable bound violations (B_s) are added to the f_s^{totC}.

The total costs of each scenario are also calculated according to the mathematical formulation in Eq. (7). The expected price, VaR, and $CVaR$ (Eqs.(3) and (5)) are calculated for risk assessment of the scheduling problem for all individuals, using the obtained total cost values of each scenario. After calculating these risk assessment variables, the aggregator starts a decision-making process based on the risk aversion factor. The aggregator chooses the optimum approach based on the OF's value.

4. Case study

The 13-bus medium voltage distribution network (DN) of a mock-up smart city from the BISITE laboratory in Salamanca, Spain [30] is used to create the case study. There are two wind farms and thirteen PV parks (15 renewable DG units), a 30 MVA substation in bus 1, and four 1 MVar capacitor banks (which are set to zero in this problem because reactive power is not considered). In terms of consumption, this DN consists of 25 different loads, including homes, offices, and some service buildings (hospital, fire station, and shopping mall). In the simulations, 500 EVs accounted for high EV adoption.

Regarding the scenarios created to deal with the uncertainty related to the considered technologies such as load consumption, renewable generation, and electricity prices, as mentioned in Section 2.3, Figure 4 shows the average forecasted load demand profiles from the fifteen scenarios created. The highest consumption values were registered between hours eleven and thirteen. The

figure also shows the average forecasted wind, and PV generation, with much lower values when compared to the load due to the extreme cases considered where renewable generation was decreased.

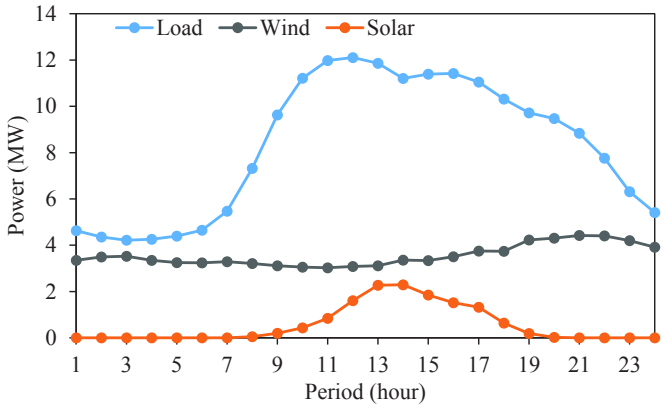

Figure 4: Average forecasted day-ahead load consumption, PV, and wind generation.

The forecasted wholesale electricity market prices and external supplier costs are shown in Figure 5. In general, the electricity market costs are lower than the external supplier costs, except in hours six and seven. So, the extreme event where the market costs substantially increase does not majorly affect the overall behavior of the market prices.

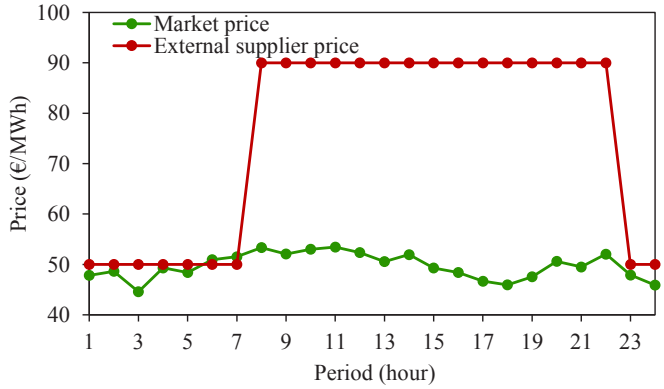

Figure 5: Average forecasted day-ahead electricity market and external supplier prices.

To address EV uncertainty, a tool in [31] was employed. With two distinct EV types–battery and plug-in hybrid–and the features and classes described in [32]. With the aid of this simulator, we can gather information about each EV's journey, including the maximum charge and discharge rates and the minimal amount of charging necessary for the EV to complete its journey within the next

hour (or hours). The remaining factors that serve as input for the optimization are described in [31].

Table 1 shows the energy resource information associated with the aggregator according to the considered prices, capacities, and forecast values for the given technologies units. The aggregator manages multiple EVs, ESSs, and different loads, power purchased from an external supplier, and energy purchased/sold on the open market.

Table 1: Energy resource information.

Energy resources		Prices (m.u./MWh)	Capacity (MW)	Forecast (MW)	Units
		min–max	min–max	min-max	
Photovoltaic		29–29		0.00–0.81	13
Wind		31–31		0.30–3.07	2
External Supplier		50–90	0.00–30.00		1
Storage units	Charge	110–110	0.00–1.25		2
	Discharge	90–90	0.00–1.25		
EVs	Charge	0–0	0.01-0.05		500
	Discharge	90–90	0.01–0.05		
DR		100–100	0.00–1.21		25
Load		0–0		0.01–2.38	25
Electricity market		29.85–104.61	0.00–10.00		1

Regarding EA parameterization, Table 2 shows the different parameters and values chosen for each algorithm. Considering the number of OF evaluations, we set the population size (Pop) and the maximum number ($maxIt$) of iterations for all algorithms to 10 and 2,000, respectively, resulting in a total of 20,000 OF evaluations.

The parameters of crossover probability (Cr) and scaling factor (F) are required for the DE, HyDE, ReSaDE and SHADE algorithms. But for SHADE these are historical memory values (M_{Cr} and M_F). Note that HyDE, ReSaDE and SHADE are self-adaptive algorithms, and the presented values are just for the initiation process. The RCEDUMDA and CLHC2RCEDUMDA algorithms consider the parameters related to the number of cells (c), that is, the number of subpopulations, the size of cells (m), number of elitist individuals (l). The number of selected individuals (s), the neighborhood ratio (r) used for neighborhood generation, and finally, the occurrence factor (α) and the number of codes (k) used for encoding/decoding of variables. CUMDANCauhy only considers the number of subpopulations and the number of selected individuals parameters. In the case of VS, only Pop and $maxIt$ need to be set. Regarding PSO, multiple parameters were set, namely a minimum and a maximum value for the inertia weight damping ratio (w_{min}, w_{max}), the personal and global learning coefficients (c_p, c_g), and the velocity factor (v_f).

The simulations were performed on a machine with a 6-core Intel Xeon E5-1650 CPU operating at 3.20 GHz with Windows 10 Pro, and 10 GB of RAM

Table 2: Metaheuristic parameterization.

EA	F	Cr	c	m	l	s	r	α	k	w_{min}	w_{max}	c_p	c_g	v_f
DE	0.30	0.50	-	-	-	-	-	-	-	-	-	-	-	-
HyDE	0.30	0.50	-	-	-	-	-	-	-	-	-	-	-	-
ReSaDE	0.90	0.14	-	-	-	-	-	-	-	-	-	-	-	-
RCEDUMDA	-	-	5	2	3	2	1	0.009	3	-	-	-	-	-
CLHC2RCEDUMDA	-	-	4	2	3	3	4	0.009	7	-	-	-	-	-
CUMDANCauchy[21]	-	-	8	-	-	2	-	-	-	-	-	-	-	-
PSO	-	-	-	-	-	-	-	-	-	0.4	0.9	1.5	2	0.1
VS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SHADE	(0.50)	(0.50)	-	-	-	-	-	-	-	-	-	-	-	-

using MATLAB 2018a.

5. Numerical results

This section includes the numerical findings for the risk-based strategies when multiple levels of risk-aversion are implemented.

5.1. Overall risk-based scheduling results

Table 3 shows the average results for twenty independent runs obtained for the risk-based day-ahead scheduling for all proposed EAs considering three levels of risk-aversion for the OF value, expected cost (f^{exC}), the costs of considering the risk evaluation ($f^{\text{exC}} + CVaR_{0.95}$), the bounds violations (B_s) in the fitness and the worst-scenario cost ($max(f^s_{tot})$). In the first, the aggregator does not consider risk in his scheduling decision because β, in this case, is zero, so it is regarded as a risk-neutral strategy. In the second, we call it a partial risk-aversion with β equal to 50%. In the third case, full risk aversion is considered.

We can conclude from Table 3 that as we increase the risk-aversion, the total day-ahead costs increase for the aggregator since the aggregator is guaranteeing a higher safety against the probability of an extreme event occurring. That is, the aggregator prevents himself from possible risk situations. By preventing himself, the value of the risk tools $CVaR$ and VaR reduces because, through these mechanisms, the cost of worst scenarios is diminished.

Regarding the EAs used, DE with one solution set to the lower bounds did not show an improvement from 0 to 50% of risk-aversion, so it got stuck in local minima. But regarding a full risk aversion approach, the algorithm achieved a slight reduction of 2.28€in worst scenario costs. The HyDE algorithm reduces the costs of the worst scenario by around 0.22% from a risk-neutral situation to a partial risk-averse situation and 0.09% from partial to full risk-aversion. The OF value increased in both cases, 37.02% and 20.93% since a higher risk aversion is being considered, guaranteeing a more robust approach. Because when considering the total risk costs ($CVaR$), the costs are reduced. The algorithm that achieved the best results was ReSaDE, which also corresponds to the winning algorithm of the competition. From a risk-neutral strategy to a partial risk aversion, a reduction of 13.08% in expenses for the worst scenario corresponding to an 11.03% decrease in $f^{\text{exC}} + CVaR_{0.95}$. If we consider a full

risk-averse approach, a reduction of 11.43% was guaranteed from the risk-neutral method when taking risk into account. In an initial phase, the following EA, RCEDUMDA, also reduced the risk costs. Still, from a partial risk aversion to a full risk aversion, the opposite was verified, with an increase of 6.75€in worst-scenario expenses, which is not alarming to the aggregator. Still, the opposite should occur when considering a higher value for β. The same case occurs for the last EA where from 0% to 50% of risk aversion, a decrease of 13.59% in worst-scenario costs is evidenced, but when we increase this risk factor to 100% an increase of 1.25% is noticed, which represents a rise of 143.96€in $f^{\text{exC}} + CVaR_{0.95}$ costs.

When we compare CUMDANCauchy to the other algorithms, all but DE, for the first two risk aversion levels, achieved better results. Even DE for a full risk aversion achieved a slight reduction of 2.38€in $f^{\text{exC}} + CVaR_{0.95}$, that is, in risk costs. Most significant is the difference between ReSaDE, which, when compared, differs from 10.09% and 24.23% in worst-scenario costs risk-neutral and risk-averse approaches.

PSO, VS, and SHADE, when initialized with one solution to the lower bounds, did not achieve any kind of variation from the lower bounds' solution for each level of risk aversion, as the table shows. All these three algorithms present the same cost values, similar to what occurred with CUMDANCauchy, which showed poor performance when compared to the remaining.

Since ReSaDE achieved the lowest cost results, we use this algorithm for the following simulations. To further evaluate the proposed method, more levels of risk aversion were considered for the ERM optimization taking the ReSaDE algorithm. Figure 6 shows the total scenario costs for five different levels of risk aversion, where the extreme events are given in scenarios 1, 7, and 11. In these scenarios, except in scenario 7, the costs are reduced the more the risk aversion increases, which is the effect of considering the risk tools like VaR and $CVaR$ given in Figure 7. As the expected cost increases, given that the remaining scenario costs also increase, the VaR and $CVaR$ costs decrease. In this situation, the most noticeable reduction was when β increases from 0 to 25%, which reduces VaR in 19.87%, and $CVaR$ in 21.64%, since the worst scenario cost also reduced in 10.60%. The other reductions are less significant, which shows that even a small weight in the risk-aversion parameter significantly reduces the risk.

Taking the proposed approach, when compared to HyDE from [20], which involves the problem and case study, ReSaDE achieved a reduction of 65.81% in OF costs for a full risk-aversion, translated in a reduction of 67.31% in worst scenario costs.

5.2. Algorithm performance

Regarding the performance of the tested EAs, Figure 8 and Figure 8 show the simulation time for the 20 runs and the corresponding convergence, respectively. Regarding optimization time, ReSaDE is the fastest algorithm, with an average of 13.86 minutes per run, followed by CUMDANCaucy, with an average time of 16.86 minutes. PSO, SHADE, DE, VS, and HyDE presented similar times

Table 3: Average risk-based results for the tested metaheuristics for 20 runs.

EA	β	OF (€)	$f^{\mathbf{exC}}$ (€)	$f^{\mathbf{exC}} + CVaR_{0.95}$ (€)	B_s (€)	$max(f^s_{tot})$ (€)
	0	8,508.16	8,508.16	18,554.27	433.33	20,709.06
DE	0.5	13,531.22	8,508.16	18,554.27	433.33	20,709.06
	1	18,551.89	8,508.29	18,551.89	433.33	20,706.78
	0	8,506.04	8,506.04	18,550.84	433.33	20,709.71
HyDE	0.5	13,506.61	8,509.07	18,504.15	433.33	20,664.24
	1	18,484.91	8,538.11	18,484.91	433.33	20,645.95
	0	8,452.48	8,452.48	16,940.69	366.67	18,719.77
ReSaDE	0.5	11,973.93	8,875.30	**15,072.55**	270.00	**16,270.54**
	1	15,003.98	8,888.71	**15,003.98**	270.00	**16,233.30**
	0	8,496.32	8,496.32	17,025.83	368.33	18,809.02
RCEDUMDA	0.5	12,339.57	9,250.21	15,428.93	333.33	16,510.34
	1	15,453.50	9,359.00	15,453.50	330.00	**16,517.09**
	0	8,505.84	8,505.84	18,123.81	415.00	20,174.11
CLHC2RCEDUMDA	0.5	12,595.81	9,056.97	16,134.66	386.67	17,432.58
	1	16,278.62	9,072.98	16,278.62	386.67	**17,650.91**
	0	8,508.16	8,508.16	18,554.27	433.33	20,709.06
CUMDANCauchy[21]	0.5	13,531.22	8,508.16	18,554.27	433.33	20,709.06
	1	18,554.27	8,508.16	18,554.27	433.33	20,709.06
	0	8,508.16	8,508.16	18,554.27	433.33	20,709.06
PSO	0.5	13,531.22	8,508.16	18,554.27	433.33	20,709.06
	1	18,554.27	8,508.16	18,554.27	433.33	20,709.06
	0	8,508.16	8,508.16	18,554.27	433.33	20,709.06
VS	0.5	13,531.22	8,508.16	18,554.27	433.33	20,709.06
	1	18,554.27	8,508.16	18,554.27	433.33	20,709.06
	0	8,508.16	8,508.16	18,554.27	433.33	20,709.06
SHADE	0.5	13,531.22	8,508.16	18,554.27	433.33	20,709.06
	1	18,554.27	8,508.16	18,554.27	433.33	20,709.06

with averages of 18.14 minutes, 18.16 minutes, 18.26 minutes, 18.78 minutes, and 19.31 minutes. RCEDUMDA and CLHC2RCEDUMDA are the slowest algorithms, with an average time of 29.46 minutes and 22.64 minutes. Even though these two algorithms presented good results regarding optimization time, they are the worst performers.

From the convergence graph in Figure 9, it is possible to conclude that the DE, and HyDE algorithms fell into local minima, fastly converging to this value. Most important CUMDANCauchy, PSO, VS, and SHADE did not present a better fitness value than the one found with one solution initialized with the lower bounds, that is, the fitness value remained a static value. As expected, the ReSaDE achieved the lowest value and converged around iteration 1,300. From the figure, it is possible to observe the initial phase of this algorithm,

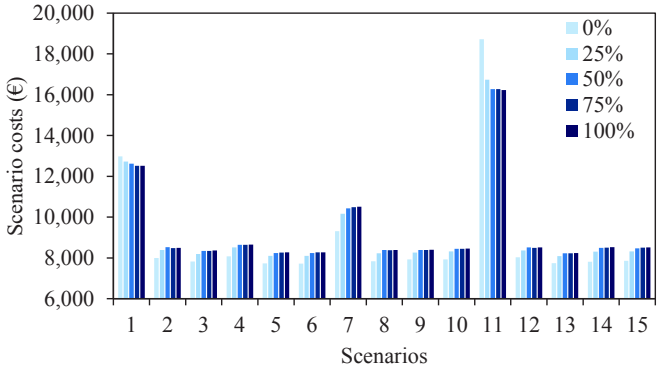

Figure 6: Average scenario costs for multiple levels of risk aversion.

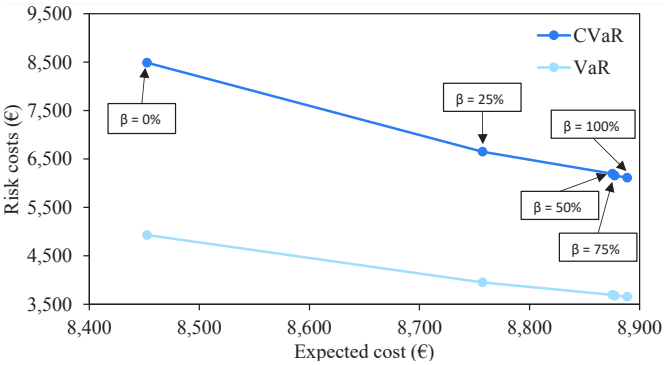

Figure 7: Average VaR and $CVaR$ costs for multiple levels of risk aversion.

where the first iterations are for dimension clustering and a warm start using SaDE without the restart loops. The RCEDUMDA algorithm showed a fast convergence around the 300 iterations with slight improvements around the 1,300 iterations. The last EA seems to have not yet converged from the number of function evaluations set for this problem, so more evaluations would allow this algorithm to improve. Still, the optimization time would increase even further, which may not be reasonable.

A Wilcoxon test was performed for the full risk-averse results with a significance threshold of 5%, as Table 4 presents. ReSaDE was used as the primary algorithm for comparison in the statistical test since it produced the lowest cost results, as previously shown in Table 3. The table shows the R+, R-, p-value, and L-sign results. The performance of the ReSaDE algorithm in relation to the other EAs is given by the R+ and R-, which are the total of positive and negative values. As expected, ReSaDE outperformed the various algorithms. The p-values demonstrate the significance of the discrepancy since they are higher than 5%. ReSaDE shows a significant disparity compared to the remaining, ex-

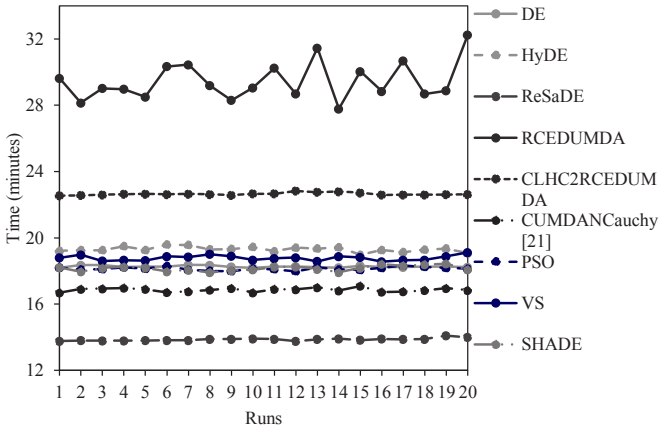

Figure 8: Time variation for the simulated runs.

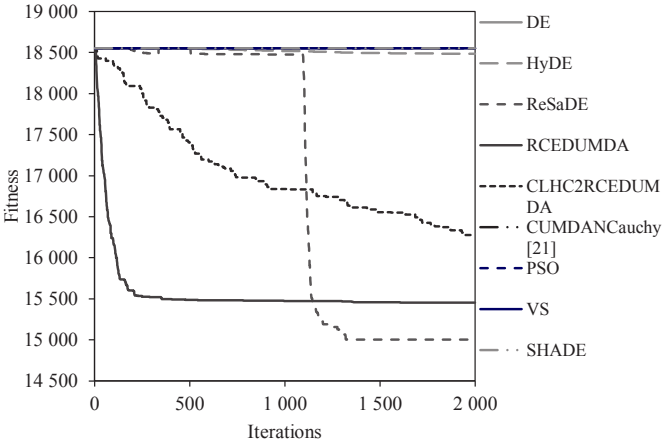

Figure 9: Average EA convergence for 20 runs for a full risk aversion.

cept RCEDUMDA, where this discrepancy is minor. The L-sign, which is used to denote statistical performance (+ means better, - means worse, and = means equal performance), shows that ReSaDE had the best statistical performance for the specified problem.

6. Conclusions

A risk-based day-ahead energy resource scheduling was proposed in this paper. The risk measurement associated with extreme events was made through the $CVaR$ tool for a given confidence level value. Due to the significant problem dimension and complexity, multiple EAs were used to solve this optimization

Table 4: Pair-wise Wilcoxon statistical test.

ReSaDE vs.	R+	R-	p-value	L-sign
DE	210	0	1.91E-06	+
HyDE	210	0	1.91E-06	+
RCEDUMDA	206	4	1.34E-05	+
CLHC2RCEDUMDA	210	0	1.91E-06	+
CUMDANCauchy[21]	210	0	1.91E-06	+
PSO	210	0	1.91E-06	+
VS	210	0	1.91E-06	+
SHADE	210	0	1.91E-06	+

problem, some taken from this year's competition on evolutionary computation in the energy domain.

For most proposed algorithms, as the risk-aversion weight increases, the worst scenario's cost decreases. That is, the risk decreases because the aggregator considers the existence of extreme events when making the scheduling decision. The winning algorithm of the competition, ReSaDE achieved the best results for all risk-aversion levels applied. This reduction resulted in smaller expenses for the aggregator when he is considering the occurrence of risk events (around 11% for $f^{\text{exC}} + CVaR_{0.95}$), which is given by the $CVaR$ tool.

One interesting note is as the risk-aversion factor increases, the expected cost also increases because the cost of the other scenarios apart from the extreme ones also increases in their majority, given that the reduction is mainly verified in the extreme scenarios, where the costs cause a high impact in the scheduling solution.

To further evaluate the performance of the applied EAs, a pair-wise Wilcoxon statistical test was used. ReSaDE showed that it outperformed all the other algorithms, even in optimization time, where it was the fastest, proving the achieved results. Compared to the author's previous work, we proposed more efficient algorithms that could achieve better solutions and perform better than CUMDANCauchy, a previously utilized algorithm in [21].

Regarding the authors' previous work, the work that we propose in this manuscript improves the cost results of the centralized day-ahead ERM problem. By initializing the proposed algorithms with one of the solutions set to the lower bounds we improved greatly costs and risk results from our previous work as demonstrated.

Acknowledgments

The present work has received funding from European Regional Development Fund through COMPETE 2020 - Operational Programme for Competitiveness and Internationalisation through the P2020 Project TIoCPS (ANI—P2020 POCI-01-0247-FEDER-046182), and has been developed under the EUREKA - ITEA3 Project TIoCPS (ITEA-18008); by National Funds

through the FCT Portuguese Foundation for Science and Technology, under Project PTDC/EEI-EEE/28983/2017 (CENERGETIC); we also acknowledge the work facilities and equipment provided by GECAD research center (UIDB/00760/2020) to the project team. João Soares is also supported by the grant CEECIND/02814/2017.

References

[1] Xin Wen, Dhaker Abbes, and Bruno Francois. Stochastic Optimization for Security-Constrained Day-Ahead Operational Planning Under PV Production Uncertainties: Reduction Analysis of Operating Economic Costs and Carbon Emissions. *IEEE Access*, 9:97039–97052, 2021.

[2] Georgios Mavromatidis, Kristina Orehounig, and Jan Carmeliet. Design of distributed energy systems under uncertainty: A two-stage stochastic programming approach. *Applied Energy*, 222:932–950, July 2018.

[3] Ahmad Ghasemi, Houman Jamshidi Monfared, Abdolah Loni, and Mousa Marzband. CVaR-based retail electricity pricing in day-ahead scheduling of microgrids. *Energy*, 227:120529, July 2021.

[4] Xindong Liu, Mohammad Shahidehpour, Yijia Cao, Zuyi Li, and Wei Tian. Risk Assessment in Extreme Events Considering the Reliability of Protection Systems. *IEEE Transactions on Smart Grid*, 6(2):1073–1081, March 2015.

[5] Xiaoyu Cao, Jianxue Wang, Jianhui Wang, and Bo Zeng. A Risk-Averse Conic Model for Networked Microgrids Planning With Reconfiguration and Reorganizations. *IEEE Transactions on Smart Grid*, 11(1):696–709, January 2020.

[6] James W. Taylor. Forecast combinations for value at risk and expected shortfall. *International Journal of Forecasting*, 36(2):428–441, April 2020.

[7] Vijaya Dixit and Manoj Kumar Tiwari. Project portfolio selection and scheduling optimization based on risk measure: a conditional value at risk approach. *Annals of Operations Research*, 285(1-2):9–33, February 2020.

[8] Guanguan Li, Qiqiang Li, Yi Liu, Huimin Liu, Wen Song, and Ran Ding. A cooperative Stackelberg game based energy management considering price discrimination and risk assessment. *International Journal of Electrical Power & Energy Systems*, 135:107461, February 2022.

[9] Josue Campos Do Prado and Ugonna Chikezie. A Decision Model for an Electricity Retailer With Energy Storage and Virtual Bidding Under Daily and Hourly CVaR Assessment. *IEEE Access*, 9:106181–106191, 2021.

[10] Wei Fan, Zhongfu Tan, Fanqi Li, Amin Zhang, Liwei Ju, Yuwei Wang, and Gejirifu De. A two-stage optimal scheduling model of integrated energy system based on CVaR theory implementing integrated demand response. *Energy*, 263:125783, January 2023.

[11] José Almeida, Joao Soares, Bruno Canizes, Iván Razo-Zapata, and Zita Vale. Day-ahead to intraday energy scheduling operation considering extreme events using risk-based approaches. *Neurocomputing*, 543:126229, July 2023.

[12] J. Soares, C. Lobo, M. Silva, Z. Vale, and H. Morais. Day-ahead distributed energy resource scheduling using differential search algorithm. In *2015 18th International Conference on Intelligent System Application to Power Systems (ISAP)*, pages 1–6, Porto, Portugal, September 2015. IEEE.

[13] Stefano Lilla, Camilo Orozco, Alberto Borghetti, Fabio Napolitano, and Fabio Tossani. Day-Ahead Scheduling of a Local Energy Community: An Alternating Direction Method of Multipliers Approach. *IEEE Transactions on Power Systems*, 35(2):1132–1142, March 2020.

[14] Oliver Kramer. *Genetic Algorithm Essentials*, volume 679 of *Studies in Computational Intelligence*. Springer International Publishing, Cham, 2017.

[15] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, Perth, WA, Australia, 1995. IEEE.

[16] Rainer Storn and Kenneth Price. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[17] Fernando Lezama, Joao Soares, Ricardo Faia, Tiago Pinto, and Zita Vale. A New Hybrid-Adaptive Differential Evolution for a Smart Grid Application Under Uncertainty. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, Rio de Janeiro, July 2018. IEEE.

[18] Berat Doğan and Tamer Ölmez. A new metaheuristic for numerical function optimization: Vortex Search algorithm. *Information Sciences*, 293:125–145, February 2015.

[19] Ryoji Tanabe and Alex Fukunaga. Success-history based parameter adaptation for Differential Evolution. In *2013 IEEE Congress on Evolutionary Computation*, pages 71–78, Cancun, Mexico, June 2013. IEEE.

[20] José Almeida, Fernando Lezama, João Soares, Zita Vale, and Bruno Canizes. Preliminary results of advanced heuristic optimization in the risk-based energy scheduling competition. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1812–1816, Boston Massachusetts, July 2022. ACM.

[21] Jose Almeida, Joao Soares, Fernando Lezama, and Zita Vale. Robust Energy Resource Management Incorporating Risk Analysis Using Conditional Value-at-Risk. *IEEE Access*, 10:16063–16077, 2022.

[22] Karin Alvehag. *Impact of dependencies in risk assessment of power distribution systems*. PhD thesis, Electric Power Systems, School of Electrical Engineering, Royal Institute of Technology, Stockholm, 2008. ISBN: 9789174151107 OCLC: 938765772.

[23] N. Growe-Kuska, H. Heitsch, and W. Romisch. Scenario reduction and scenario tree construction for power management problems. In *2003 IEEE Bologna Power Tech Conference Proceedings,*, volume 3, pages 152–158, Bologna, Italy, 2003. IEEE.

[24] Ansel Y. Rodríguez-González, Ramón Aranda, Miguel Á. Álvarez Carmona, Yoan Martínez-López, and Julio Madera-Quintana. Applying ring cellular encode-decode UMDA to risk-based energy scheduling. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1–2, Boston Massachusetts, July 2022. ACM.

[25] Ansel Y. Rodríguez-González, Samantha Barajas, Ramón Aranda, Yoan Martínez-López, and Julio Madera-Quintana. Ring cellular encode-decode UMDA: simple is effective. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1–2, Lille France, July 2021. ACM.

[26] Fernando Lezama, João Soares, Ricardo Faia, and Zita Vale. Hybrid-adaptive differential evolution with decay function (HyDE-DF) applied to the 100-digit challenge competition on single objective numerical optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 7–8, Prague Czech Republic, July 2019. ACM.

[27] Weiming Liu, Yinda Zhou, Bin Li, and Ke Tang. Cooperative Co-evolution with Soft Grouping for Large Scale Global Optimization. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 318–325, Wellington, New Zealand, June 2019. IEEE.

[28] Zhenyu Yang, Ke Tang, and Xin Yao. Self-adaptive differential evolution with neighborhood search. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 1110–1116, Hong Kong, China, June 2008. IEEE.

[29] Betül Sultan Yıldız, Sumit Kumar, Nantiwat Pholdee, Sujin Bureerat, Sadiq M. Sait, and Ali Riza Yildiz. A new chaotic Lévy flight distribution optimization algorithm for solving constrained engineering problems. *Expert Systems*, 39(8), September 2022.

[30] Bruno Canizes, João Soares, Zita Vale, and Juan M. Corchado. Optimal distribution grid operation using DLMP-based pricing for electric vehicle charging infrastructure in a smart city. *Energies*, 12(4), 2019.

[31] João Soares, Bruno Canizes, Cristina Lobo, Zita Vale, and Hugo Morais. Electric Vehicle Scenario Simulator Tool for Smart Grid Operators. *Energies*, 5(6):1881–1899, June 2012.

[32] Jose Almeida, Joao Soares, Bruno Canizes, Fernando Lezama, Mohammad Ali Ghazvini Fotouhi, and Zita Vale. Evolutionary Algorithms for Energy Scheduling under uncertainty considering Multiple Aggregators. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 225–232, Kraków, Poland, June 2021. IEEE.