



HAL
open science

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

► To cite this version:

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. *Mathematics and Computers in Simulation*, 2023, 10.1016/j.matcom.2023.07.010 . hal-04375453

HAL Id: hal-04375453

<https://hal.science/hal-04375453>

Submitted on 5 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Jose Almeida¹., Joao Soares^{1.*}, Fernando Lezama¹, Zita Vale¹, and Bruno Francois²

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 Almeida¹, Joao Soares^{1,*}, Fernando Lezama¹, Zita Vale¹, and Bruno Francois²

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)

$p^{\max\text{Red}}$	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
$p^{\max\text{Disch}}$	battery's maximum active discharging power for EVs and ESSs (MW)
$p^{\max\text{Ch}}$	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)
$p^{\max\text{Buy}}$	day-ahead maximum electricity market bid (MW)
$p^{\max\text{Sell}}$	day-ahead maximum electricity market offer (MW)
p^{load}	forecasted day-ahead active load power consumption (MW)
p^{DGnd}	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

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

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

45 Multiple metaheuristics applied to energy-related problems emerge as a re-
46 sult of moving from the standard metaheuristics such as the genetic algorithms
47 [14], particle swarm optimization (PSO) [15], or differential evolution (DE) [16],
48 towards more sophisticated and efficient approaches. Improved versions of these
49 algorithms (Hybrid-Adaptive DE (HyDE) [17], Vortex Search (VS) [18], Success-
50 History based Adaptive DE (SHADE) [19], etc.) are also being applied to
51 problems in the SG paradigm, including the risk-based ERM problem we are
52 modeling in this work, achieving acceptable results as shown in [20].

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

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

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

79 • comparison of new and complex EAs applied in the "Competition on
80 Evolutionary Computation in the Energy Domain: Risk-based Energy
81 Scheduling¹," with the algorithm used in [21].

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

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

91 2. Risk-based ERM methodology

92 This section presents the mathematical model for risk measurement consid-
93 ering the *CVaR* mechanism and also discusses the day-ahead scheduling taking
94 into account total scenario cost and problem restrictions.

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

105 2.1. Risk-based formulation

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

111 In this work, a set of scenarios is generated to deal with the uncertainty of
112 such resources as the demand, renewables, prices, and EV user uncertainty. As
113 such, we formulate the total cost of each generated scenario ($J_s^{\text{tot}C}$) as:

¹<http://www.gecad.isep.ipp.pt/ERM-competitions/2022-2/>

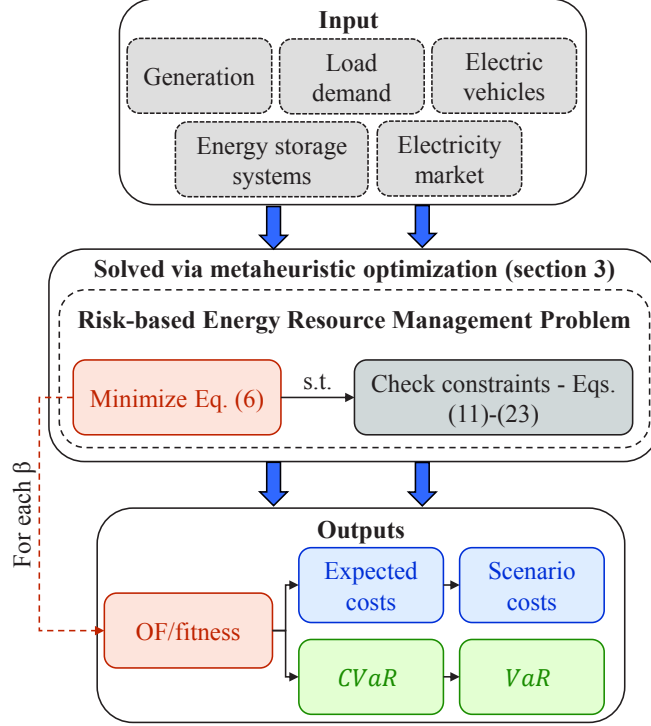


Figure 1: Flowchart of the proposed problem methodology.

$$f_s^{\text{totC}} = f_s^{\text{Cost}} + B_s \quad (1)$$

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

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

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

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

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:

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

where:

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

126 $z - \text{score}$ is computed in MATLAB with α equal to 95% using the **norminv()**
127 function.

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

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

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

132 2.2. Day-ahead scheduling formulation

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

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

135 where:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$E_{(v,t,s)}^{\text{stored}} = E_{(v,t-1,s)}^{\text{stored}} + \eta_{(v)}^{\text{ch}} \cdot p_{(v,t,s)}^{\text{EV}} \cdot \Delta t - \frac{1}{\eta_{(v)}^{\text{disch}}} \cdot p_{(v,t,s)}^{\text{EV}} \cdot \Delta t \quad \forall v, \forall t, \forall s \quad (19)$$

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

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

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

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

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

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

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

166 2.3. Uncertainty

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

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

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

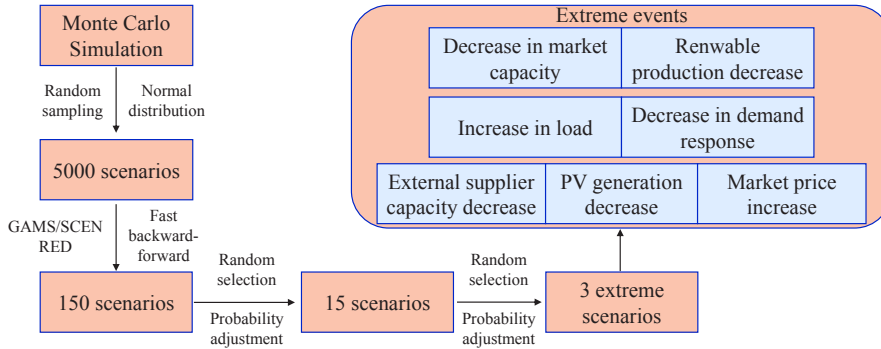


Figure 2: Scenario generation methodology.

196 3. Optimization

197 This section briefly describes each EA, solution encoding, and fitness evalu-
 198 ation process, which are typical for CI optimization.

199 3.1. Evolutionary algorithms

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

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

207 3.1.1. DE

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

215 In the iterative process of the algorithm, three random individuals are se-
 216 lected from the generated solution, and a mutation strategy is applied, gener-
 217 ating a donor vector. The binomial recombination is then applied, generating
 218 the trial vector $\vec{u}_{i,it}$. We then verify the boundary constraints, and if they are
 219 violated, the variables are updated to the minimum or maximum bounds ac-
 220 cordingly. Then the trial vector is evaluated, and elitism selection is applied.
 221 Finally, the best solution \vec{x}_{best} is updated with the minimum value of $\vec{x}_{i,it}$.

Algorithm 1 Standard Differential Evolution

```

1: Define algorithm parameters  $Pop, maxIt, F, Cr$ 
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:   for all  $Pop$  do
9:     Select three random individuals  $\vec{x}_{r1,it} \neq \vec{x}_{r2,it} \neq \vec{x}_{r3,it} \in \vec{x}_{i,it}$ 
10:    Apply mutation strategy  $\vec{m}_{i,it} = \vec{x}_{r1,it} + F(\vec{x}_{r2,it} - \vec{x}_{r3,it})$ 
11:    Apply binomial recombination (generate trial vector  $\vec{u}_{i,it}$ )
12:    Verify boundary control
13:    Evaluate  $\vec{u}_{i,it}$ 
14:    Apply elitism selection
15:    if  $\vec{u}_{i,it} < \vec{x}_{i,it}$  then
16:       $\vec{x}_{i,it+1} = \vec{u}_{i,it}$ 
17:      Update solution
18:    end if
19:  end for
20:  Store  $\vec{x}_{best} \leftarrow \min(\vec{x}_{i,it})$ 
21:   $it \leftarrow it + 1$ 
22: end while

```

222 *3.1.2. HyDE*

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

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

231 The mutation strategy applied is given by Eq. (24), where a random per-
 232 turbation factor ($\mathcal{N}(F_i^2, 1)$) is applied to the best solution found (x_{best}). The
 233 remaining steps of the algorithm are similar to the previous algorithm. Finally,
 234 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**
-

235 *3.1.3. ReSaDE*

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

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

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

Algorithm 3 Restart-assisted Self-adaptive Differential Evolution

```

1: Define algorithm parameters  $Pop$ ,  $maxIt$ ,  $F$ , and  $Cr$ 
2: Initialize  $maxUp$  and  $maxLow$ 
3: for all  $D$  do
4:   Perform variable soft-grouping ( $G_j$ )
5: end for
6: for all  $G_j$  do
7:   Initialize  $maxG_{Up}$  and  $maxG_{Low}$ 
8:   Generate one grouped solution as  $maxG_{Low}$  and the rest randomly between  $maxG_{Up}$  and  $maxG_{Low}$ 
9:   Evaluate grouped solution
10:  Store best grouped solution as  $x_{GBest}$ 
11:  Define iterations for self-adaptive DE ( $max_{it}SaDE$ )
12:   $it \leftarrow 1$ 
13:  while  $it \leq max_{it}SaDE$  do
14:    Run self-adaptive DE
15:    Perform elitist selection
16:  end while
17:  Store current best as  $x_{GBest}$ 
18: end for
19: Sort grouped variables according to  $x_{GBest}$ 
20: Select most effective group of variables
21: Define iterations for standard DE ( $max_{it}DE$ )
22: while  $it \leq max_{it}DE$  do
23:   Run DE as described in Algorithm 1
24: end while
25: Adjust trust region
26: Run self-adaptive DE with restart loops for remaining iterations
27: Store best solution as  $x_{best}$ 

```

252 3.1.4. RCEDUMDA

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

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

```

1: Define algorithm parameters  $Pop$ ,  $maxIt$ ,  $c$ ,  $m$ ,  $l$ ,  $s$ ,  $r$ ,  $\alpha$  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 individ-
   uals
13:    Scale according to  $\alpha$  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  $\vec{x}_{best} \leftarrow min(Pop)$ 

```

266 3.1.5. CLHC2RCEDUMDA

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

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

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

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

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

277 where λ represents the Lévy coefficient, and the Chaotic Lévy distribution ap-
 278 plied in this algorithm can be formulated as in Eq. (27), which uses the Gaussian
 279 map's randomly generated number in the Lévy distribution, the Chaotic equa-
 280 tion is employed to increase the variety and quality of the new population, which
 281 in turn enhances the algorithm's capacity to do a global search.

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

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}
-

282 *3.1.6. Particle Swarm Optimization*

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

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

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

$$\vec{v}_i^{min} = -\vec{v}_i^{max} \quad (29)$$

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

$$w = w_{max} - \frac{w_{max} - w_{min}}{maxIt} \cdot it \quad (30)$$

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

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

$$\vec{x}_{i,it+1} = \vec{x}_{i,it} + \vec{v}_{i,it+1} \quad (32)$$

301 where w is the inertia weight, c_p and c_g are the personal and global accelera-
 302 tion coefficients, and r_p and r_g are two random coefficients that vary between
 303 $[0,1]$. The personal and global best particle positions are described in $\vec{x}_{i,it}^p$ and
 304 $\vec{x}_{i,it}^g$, respectively. Following this process, a boundary control needs to be set
 305 for particle velocity and position, and then the newly generated particles are
 306 evaluated. Finally, the particle with the lowest fitness value is stored as the best
 307 individual.

308 3.1.7. Vortex Search

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

$$a_{it} = \frac{it}{maxIt} \quad (33)$$

Algorithm 6 Particle Swarm Optimization

```
1: Define algorithm parameters  $Pop, maxIt, w_{min}, w_{max}, c_p, c_g, v_f$ 
2: Initialize  $maxUp$  and  $maxLow$ 
3: Calculate  $v_i^{max}$  and  $v_i^{min}$  based on Eqs. (28)-(29)
4: Generate one solution as  $maxLow$  and the rest randomly between  $maxUp$ 
   and  $maxLow$ 
5: Generate initial velocity between  $v_i^{min}$  and  $v_i^{max}$ 
6: Evaluate initial solution
7: Store best individual as the  $\vec{x}_{best}$ 
8:  $it \leftarrow 1$ 
9: while  $it \leq maxIt$  do
10:   Update inertia weight via Eq. (30)
11:   for all  $Pop$  do
12:     Update particle velocity via Eq. (31)
13:     Update particle position via Eq. (32)
14:     Verify boundary control
15:     Evaluate new solution ( $\vec{x}_{i,it+1}$ )
16:   end for
17:   Store  $\vec{x}_{best} \leftarrow \min(\vec{x}_{i,it+1})$ 
18:    $it \leftarrow it + 1$ 
19: end while
```

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

$$\mu = \frac{maxUp - maxLow}{2} \quad (34)$$

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

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

3.1.8. Success-history based Adaptive Differential Evolution

321 SHADE is an algorithm that uses a parameter adaptation method based on
322 a historical record of effective parameter adjustments [19]. The process used is
323 described in algorithm 8, where initially algorithm parameters are defined. In
324 this case, and different from HyDE, for example, F and Cr are parameters that
325 will be recorded in memory for H entries, which are designed as M_{Cr} and M_F ,
326 where H represents the memory size. Additionally, an archive A is also set to
327 store problem solutions. After, the algorithm follows similar processes as the
328 precious algorithms described. Entering the iterative process, a random entry
329 (r_i) is selected, which determines the position of memory to update $Cr_{i,it}$ and
330

Algorithm 7 Vortex Search, adapted from [18]

- 1: Define algorithm parameters Pop , $maxIt$
 - 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: Sample a values within $[0,1]$ by using Eq. (33)
 - 9: Calcualte initial circle radius using Eqs. (34)-(35)
 - 10: Generate Pop candidate solutions ($\vec{x}_{i,it+1}$) using Gaussian distribution
 - 11: Verify boundary control
 - 12: Evaluate new solution ($\vec{x}_{i,it+1}$)
 - 13: Store $\vec{x}_{best} \leftarrow \min(\vec{x}_{i,it+1})$
 - 14: $it \leftarrow it + 1$
 - 15: **end while**
-

331 $F_{i,it}$. Following this process, a random value $p_{i,it}$ is generated, where $p_{min} =$
332 $Pop/2$ and the trial vector is generated according to:

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

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

3.2. Solution generation

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

350 The scheduling problem in question includes 13,680 variables per solution,
351 divided into 570 variables every period, with 21 variables forming the active
352 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]

```
1: Define algorithm parameters  $Pop$ ,  $maxIt$ ,  $M_F$ ,  $M_{Cr}$ 
2: Set memory size with  $H$  entries
3: Set archive  $A$ 
4: Initialize  $maxUp$  and  $maxLow$ 
5: Generate one solution as  $maxLow$  and the rest randomly between  $maxUp$ 
   and  $maxLow$ 
6: Evaluate initial solution
7: Store best individual as the  $\vec{x}_{best}$ 
8:  $it \leftarrow 1$ 
9:  $k \leftarrow 1$ 
10: while  $it \leq maxIt$  do
11:   Initialize  $S_{Cr}, S_F$ 
12:   for all  $Pop$  do
13:     Select randomly between  $[1, H]$  ( $r_i$ )
14:      $Cr_{i,it} = randn_i(M_{Cr,r_i}, 0.1)$ 
15:      $F_{i,it} = randc_i(M_{F,r_i}, 0.1)$ 
16:      $p_{i,it} = rand[p_{min}, 0.2]$ 
17:     Generate trial vector ( $\vec{u}_{i,it}$ ) using the mutation strategy in Eq. (36)
18:   end for
19:   Evaluate trial vector
20:   for all  $Pop$  do
21:     if  $f(\vec{u}_{i,it}) \leq f(\vec{x}_{i,it})$  then
22:        $\vec{x}_{i,it+1} = \vec{u}_{i,it}$ 
23:     end if
24:     if  $f(\vec{u}_{i,it}) < f(\vec{x}_{i,it})$  then
25:       Update solution in  $A$ 
26:       Update  $Cr_{i,it}$  and  $F_{i,it}$ 
27:     end if
28:   end for
29:   If archive size exceeds  $|A|$ , select random individuals to delete ( $|A| \leq$ 
    $|Pop|$ )
30:   if  $S_{Cr}, S_F \neq 0$  then
31:     Update  $M_{Cr,k}, M_{F,k}$  based on  $S_{Cr}, S_F$ 
32:      $k \leftarrow k + 1$ , where  $k < H$ , otherwise  $k \leftarrow 1$ 
33:   end if
34:    $it \leftarrow it + 1$ 
35: end while
```

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

355 3.3. Fitness evaluation

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

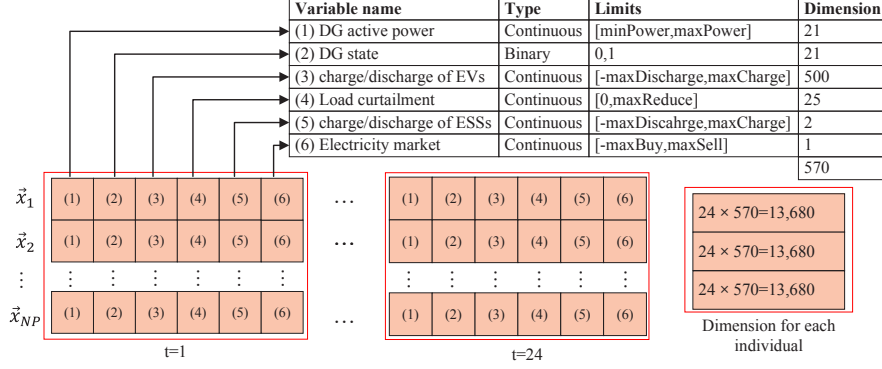


Figure 3: Proposed solution encoding.

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 $f_s^{\text{tot}C}$.

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

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

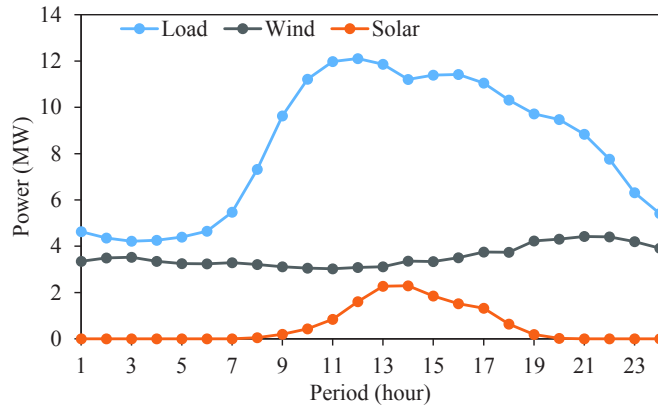


Figure 4: Average forecasted day-ahead load consumption, PV, and wind generation.

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

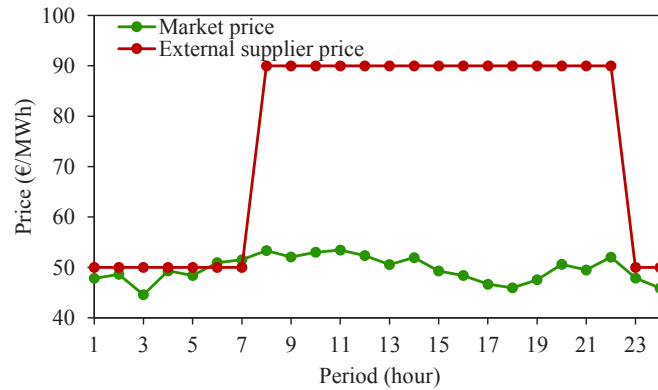


Figure 5: Average forecasted day-ahead electricity market and external supplier prices.

394 To address EV uncertainty, a tool in [31] was employed. With two distinct
 395 EV types—battery and plug-in hybrid—and the features and classes described in
 396 [32]. With the aid of this simulator, we can gather information about each EV's
 397 journey, including the maximum charge and discharge rates and the minimal
 398 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	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)	-	-	-	-	-	-	-	-	-	-	-	-

428 using MATLAB 2018a.

429 5. Numerical results

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

432 5.1. Overall risk-based scheduling results

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

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

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

460 risk-averse approach, a reduction of 11.43% was guaranteed from the risk-neutral
 461 method when taking risk into account. In an initial phase, the following EA,
 462 RCEDUMDA, also reduced the risk costs. Still, from a partial risk aversion
 463 to a full risk aversion, the opposite was verified, with an increase of 6.75€ in
 464 worst-scenario expenses, which is not alarming to the aggregator. Still, the
 465 opposite should occur when considering a higher value for β . The same case
 466 occurs for the last EA where from 0% to 50% of risk aversion, a decrease of
 467 13.59% in worst-scenario costs is evidenced, but when we increase this risk factor
 468 to 100% an increase of 1.25% is noticed, which represents a rise of 143.96€ in
 469 $f^{\text{exC}} + CVaR_{0.95}$ costs.

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

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

481 Since ReSaDE achieved the lowest cost results, we use this algorithm for the
 482 following simulations. To further evaluate the proposed method, more levels
 483 of risk aversion were considered for the ERM optimization taking the ReSaDE
 484 algorithm. Figure 6 shows the total scenario costs for five different levels of
 485 risk aversion, where the extreme events are given in scenarios 1, 7, and 11. In
 486 these scenarios, except in scenario 7, the costs are reduced the more the risk
 487 aversion increases, which is the effect of considering the risk tools like VaR
 488 and $CVaR$ given in Figure 7. As the expected cost increases, given that the
 489 remaining scenario costs also increase, the VaR and $CVaR$ costs decrease. In
 490 this situation, the most noticeable reduction was when β increases from 0 to
 491 25%, which reduces VaR in 19.87%, and $CVaR$ in 21.64%, since the worst
 492 scenario cost also reduced in 10.60%. The other reductions are less significant,
 493 which shows that even a small weight in the risk-aversion parameter significantly
 494 reduces the risk.

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

499 5.2. Algorithm performance

500 Regarding the performance of the tested EAs, Figure 8 and Figure 8 show the
 501 simulation time for the 20 runs and the corresponding convergence, respectively.
 502 Regarding optimization time, ReSaDE is the fastest algorithm, with an average
 503 of 13.86 minutes per run, followed by CUMDANCauchy, with an average time
 504 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^{exC} (€)	$f^{\text{exC}} +$ $CVaR_{0.95}$ (€)	B_s (€)	$\max(f_{tot}^s)$ (€)
DE	0	8,508.16	8,508.16	18,554.27	433.33	20,709.06
	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
HyDE	0	8,506.04	8,506.04	18,550.84	433.33	20,709.71
	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
ReSaDE	0	8,452.48	8,452.48	16,940.69	366.67	18,719.77
	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
RCEDUMDA	0	8,496.32	8,496.32	17,025.83	368.33	18,809.02
	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
CLHC2RCEDUMDA	0	8,505.84	8,505.84	18,123.81	415.00	20,174.11
	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
CUMDANCauchy[21]	0	8,508.16	8,508.16	18,554.27	433.33	20,709.06
	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
PSO	0	8,508.16	8,508.16	18,554.27	433.33	20,709.06
	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
VS	0	8,508.16	8,508.16	18,554.27	433.33	20,709.06
	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
SHADE	0	8,508.16	8,508.16	18,554.27	433.33	20,709.06
	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

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

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

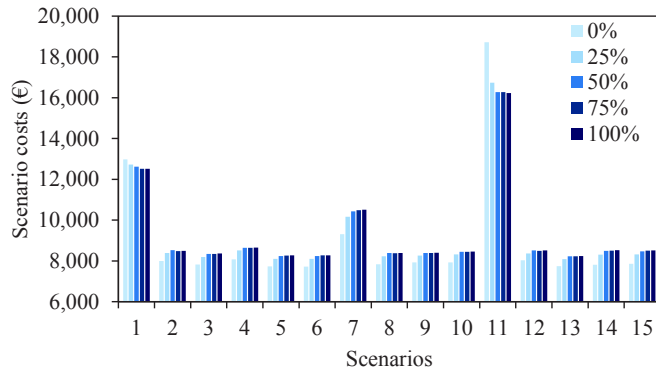


Figure 6: Average scenario costs for multiple levels of risk aversion.

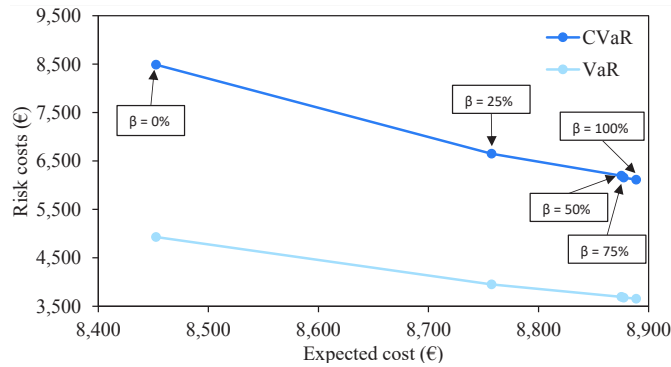


Figure 7: Average *VaR* and *CVaR* costs for multiple levels of risk aversion.

517 where the first iterations are for dimension clustering and a warm start using
 518 SaDE without the restart loops. The RCEDUMDA algorithm showed a fast
 519 convergence around the 300 iterations with slight improvements around the
 520 1,300 iterations. The last EA seems to have not yet converged from the number
 521 of function evaluations set for this problem, so more evaluations would allow this
 522 algorithm to improve. Still, the optimization time would increase even further,
 523 which may not be reasonable.

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

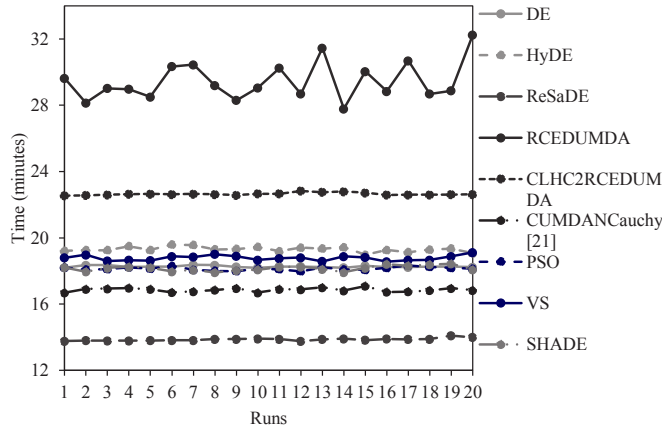


Figure 8: Time variation for the simulated runs.

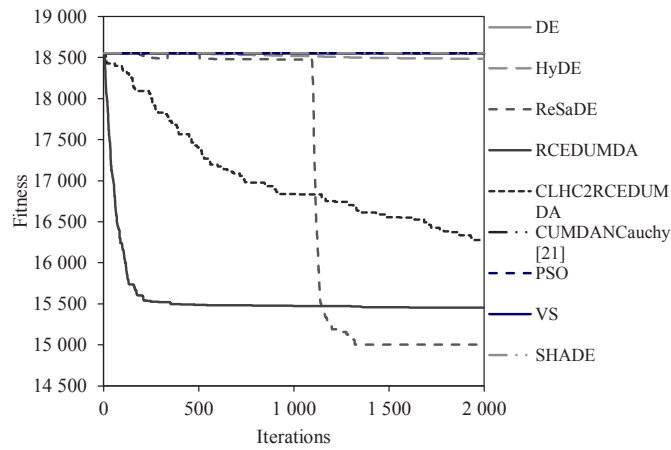


Figure 9: Average EA convergence for 20 runs for a full risk aversion.

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

537 6. Conclusions

538 A risk-based day-ahead energy resource scheduling was proposed in this
 539 paper. The risk measurement associated with extreme events was made through
 540 the *CVaR* tool for a given confidence level value. Due to the significant problem
 541 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	+

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

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

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

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

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

567 Acknowledgments

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

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

578 References

- 579 [1] Xin Wen, Dhaker Abbes, and Bruno Francois. Stochastic Optimization for
580 Security-Constrained Day-Ahead Operational Planning Under PV Produc-
581 tion Uncertainties: Reduction Analysis of Operating Economic Costs and
582 Carbon Emissions. *IEEE Access*, 9:97039–97052, 2021.
- 583 [2] Georgios Mavromatidis, Kristina Orehounig, and Jan Carmeliet. Design
584 of distributed energy systems under uncertainty: A two-stage stochastic
585 programming approach. *Applied Energy*, 222:932–950, July 2018.
- 586 [3] Ahmad Ghasemi, Houman Jamshidi Monfared, Abdollah Loni, and Mousa
587 Marzband. CVaR-based retail electricity pricing in day-ahead scheduling
588 of microgrids. *Energy*, 227:120529, July 2021.
- 589 [4] Xindong Liu, Mohammad Shahidehpour, Yijia Cao, Zuyi Li, and Wei Tian.
590 Risk Assessment in Extreme Events Considering the Reliability of Protec-
591 tion Systems. *IEEE Transactions on Smart Grid*, 6(2):1073–1081, March
592 2015.
- 593 [5] Xiaoyu Cao, Jianxue Wang, Jianhui Wang, and Bo Zeng. A Risk-Averse
594 Conic Model for Networked Microgrids Planning With Reconfiguration and
595 Reorganizations. *IEEE Transactions on Smart Grid*, 11(1):696–709, Jan-
596 uary 2020.
- 597 [6] James W. Taylor. Forecast combinations for value at risk and expected
598 shortfall. *International Journal of Forecasting*, 36(2):428–441, April 2020.
- 599 [7] Vijaya Dixit and Manoj Kumar Tiwari. Project portfolio selection and
600 scheduling optimization based on risk measure: a conditional value at risk
601 approach. *Annals of Operations Research*, 285(1-2):9–33, February 2020.
- 602 [8] Guanguan Li, Qiqiang Li, Yi Liu, Huimin Liu, Wen Song, and Ran Ding. A
603 cooperative Stackelberg game based energy management considering price
604 discrimination and risk assessment. *International Journal of Electrical
605 Power & Energy Systems*, 135:107461, February 2022.
- 606 [9] Josue Campos Do Prado and Ugonna Chikezie. A Decision Model for an
607 Electricity Retailer With Energy Storage and Virtual Bidding Under Daily
608 and Hourly CVaR Assessment. *IEEE Access*, 9:106181–106191, 2021.

- 609 [10] Wei Fan, Zhongfu Tan, Fanqi Li, Amin Zhang, Liwei Ju, Yuwei Wang, and
610 Gejirifu De. A two-stage optimal scheduling model of integrated energy
611 system based on CVaR theory implementing integrated demand response.
612 *Energy*, 263:125783, January 2023.
- 613 [11] José Almeida, Joao Soares, Bruno Canizes, Iván Razo-Zapata, and Zita
614 Vale. Day-ahead to intraday energy scheduling operation considering ex-
615 treme events using risk-based approaches. *Neurocomputing*, 543:126229,
616 July 2023.
- 617 [12] J. Soares, C. Lobo, M. Silva, Z. Vale, and H. Morais. Day-ahead distributed
618 energy resource scheduling using differential search algorithm. In *2015*
619 *18th International Conference on Intelligent System Application to Power*
620 *Systems (ISAP)*, pages 1–6, Porto, Portugal, September 2015. IEEE.
- 621 [13] Stefano Lilla, Camilo Orozco, Alberto Borghetti, Fabio Napolitano, and
622 Fabio Tossani. Day-Ahead Scheduling of a Local Energy Community: An
623 Alternating Direction Method of Multipliers Approach. *IEEE Transactions*
624 *on Power Systems*, 35(2):1132–1142, March 2020.
- 625 [14] Oliver Kramer. *Genetic Algorithm Essentials*, volume 679 of *Studies*
626 *in Computational Intelligence*. Springer International Publishing, Cham,
627 2017.
- 628 [15] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceed-*
629 *ings of ICNN'95 - International Conference on Neural Networks*, volume 4,
630 pages 1942–1948, Perth, WA, Australia, 1995. IEEE.
- 631 [16] Rainer Storn and Kenneth Price. Differential Evolution – A Simple and Ef-
632 ficient Heuristic for Global Optimization over Continuous Spaces. *Journal*
633 *of Global Optimization*, 11:341–359, 1997.
- 634 [17] Fernando Lezama, Joao Soares, Ricardo Faia, Tiago Pinto, and Zita Vale. A
635 New Hybrid-Adaptive Differential Evolution for a Smart Grid Application
636 Under Uncertainty. In *2018 IEEE Congress on Evolutionary Computation*
637 *(CEC)*, pages 1–8, Rio de Janeiro, July 2018. IEEE.
- 638 [18] Berat Doğan and Tamer Ölmez. A new metaheuristic for numerical function
639 optimization: Vortex Search algorithm. *Information Sciences*, 293:125–145,
640 February 2015.
- 641 [19] Ryoji Tanabe and Alex Fukunaga. Success-history based parameter adap-
642 tion for Differential Evolution. In *2013 IEEE Congress on Evolutionary*
643 *Computation*, pages 71–78, Cancun, Mexico, June 2013. IEEE.
- 644 [20] José Almeida, Fernando Lezama, João Soares, Zita Vale, and Bruno
645 Canizes. Preliminary results of advanced heuristic optimization in the
646 risk-based energy scheduling competition. In *Proceedings of the Genetic*
647 *and Evolutionary Computation Conference Companion*, pages 1812–1816,
648 Boston Massachusetts, July 2022. ACM.

- 649 [21] Jose Almeida, Joao Soares, Fernando Lezama, and Zita Vale. Robust En-
650 ergy Resource Management Incorporating Risk Analysis Using Conditional
651 Value-at-Risk. *IEEE Access*, 10:16063–16077, 2022.
- 652 [22] Karin Alvehag. *Impact of dependencies in risk assessment of power dis-*
653 *tribution systems*. PhD thesis, Electric Power Systems, School of Electri-
654 cal Engineering, Royal Institute of Technology, Stockholm, 2008. ISBN:
655 9789174151107 OCLC: 938765772.
- 656 [23] N. Growe-Kuska, H. Heitsch, and W. Romisch. Scenario reduction and
657 scenario tree construction for power management problems. In *2003 IEEE*
658 *Bologna Power Tech Conference Proceedings*, volume 3, pages 152–158,
659 Bologna, Italy, 2003. IEEE.
- 660 [24] Ansel Y. Rodríguez-González, Ramón Aranda, Miguel Á. Álvarez Car-
661 mona, Yoan Martínez-López, and Julio Madera-Quintana. Applying ring
662 cellular encode-decode UMDA to risk-based energy scheduling. In *Proceed-*
663 *ings of the Genetic and Evolutionary Computation Conference Companion*,
664 pages 1–2, Boston Massachusetts, July 2022. ACM.
- 665 [25] Ansel Y. Rodríguez-González, Samantha Barajas, Ramón Aranda, Yoan
666 Martínez-López, and Julio Madera-Quintana. Ring cellular encode-decode
667 UMDA: simple is effective. In *Proceedings of the Genetic and Evolutionary*
668 *Computation Conference Companion*, pages 1–2, Lille France, July 2021.
669 ACM.
- 670 [26] Fernando Lezama, João Soares, Ricardo Faia, and Zita Vale. Hybrid-
671 adaptive differential evolution with decay function (HyDE-DF) applied to
672 the 100-digit challenge competition on single objective numerical optimiza-
673 tion. In *Proceedings of the Genetic and Evolutionary Computation Confer-*
674 *ence Companion*, pages 7–8, Prague Czech Republic, July 2019. ACM.
- 675 [27] Weiming Liu, Yinda Zhou, Bin Li, and Ke Tang. Cooperative Co-evolution
676 with Soft Grouping for Large Scale Global Optimization. In *2019 IEEE*
677 *Congress on Evolutionary Computation (CEC)*, pages 318–325, Wellington,
678 New Zealand, June 2019. IEEE.
- 679 [28] Zhenyu Yang, Ke Tang, and Xin Yao. Self-adaptive differential evolu-
680 tion with neighborhood search. In *2008 IEEE Congress on Evolutionary*
681 *Computation (IEEE World Congress on Computational Intelligence)*, pages
682 1110–1116, Hong Kong, China, June 2008. IEEE.
- 683 [29] Betül Sultan Yıldız, Sumit Kumar, Nantiwat Pholdee, Sujin Bureerat,
684 Sadiq M. Sait, and Ali Riza Yildiz. A new chaotic Lévy flight distribu-
685 tion optimization algorithm for solving constrained engineering problems.
686 *Expert Systems*, 39(8), September 2022.
- 687 [30] Bruno Canizes, João Soares, Zita Vale, and Juan M. Corchado. Optimal
688 distribution grid operation using DLMP-based pricing for electric vehicle
689 charging infrastructure in a smart city. *Energies*, 12(4), 2019.

- 690 [31] João Soares, Bruno Canizes, Cristina Lobo, Zita Vale, and Hugo Morais.
691 Electric Vehicle Scenario Simulator Tool for Smart Grid Operators. *Ener-*
692 *gies*, 5(6):1881–1899, June 2012.
- 693 [32] Jose Almeida, Joao Soares, Bruno Canizes, Fernando Lezama, Moham-
694 mad Ali Ghazvini Fotouhi, and Zita Vale. Evolutionary Algorithms for
695 Energy Scheduling under uncertainty considering Multiple Aggregators. In
696 *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 225–232,
697 Kraków, Poland, June 2021. IEEE.