Online Learning-Based Adaptive Hybrid Benders Decomposition for Risk-Averse Optimal Sizing

Saif Ahmad[a] Seifeddine BEN ELGHALI[a] Hafiz Ahmed[b]

***Abstract*—Optimal sizing problem (OSP) under uncertainty is often formulated as two-stage stochastic programming (2SP). However, this formulation typically introduces significant computational and RAM (memory) bottlenecks as the scenario size grows. Benders Decomposition (BD) is a popular approach to tackle this problem, but it suffers from slow convergence to the exact solution. To address this problem, we propose an accelerated online learning-based adaptive hybrid BD algorithm. The proposed method avoids getting stuck in the infeasible region during early iterations by explicitly embedding a carefully selected scenario in the master problem, while online learning based on logistic regression (LR) leverages a light machine learning (ML) algorithm to avoid solving the entire scenario set at every iteration. The OSP is formulated to select a behind-the-meter battery energy storage system in a multi-site energy community with existing renewables. The use case explores an interesting middle-ground between deterministic methods and training-heavy ML models, showcasing the potential of ML-assisted decision-making. Numerical results on scenario sets of up to 500 illustrate the effectiveness and scalability of the proposed approach.**

***Index Terms*—Machine Learning, two-stage stochastic problem, optimal sizing, battery energy storage system, Benders decomposition**

I. Introduction

The year 2026 is turning out to be one of the warmest years to date, another record that underscores the severity of global warming and the urgency of expediting the transition to renewables. Decarbonization of the urban energy mix via smart aggregation, such as energy communities (ECs), is a promising concept that prioritizes collective self-consumption of renewables and can alleviate distribution network stress during peak times. This work focuses on the economic sizing of battery energy storage systems (BESS) in ECs with preinstalled PV on one or more sites, a recurring scenario in which declining PV module costs and high feed-in tariffs drive initial investments, while recent regulatory changes demand a shift towards BESS installation to remain profitable.

Decision-making under uncertainty, such as optimal BESS sizing, can often be formulated as a stochastic optimization problem. In this work, we focus on two-stage stochastic programs (2SPs), which involve first-stage (here-and-now) decisions made before uncertainty is realized and second-stage (recourse) decisions made under the assumption of perfect scenario foresight [1]. We consider an illustrative example for sizing a behind-the-meter battery energy storage system (BESS) in energy communities (ECs) with multiple sites that exchange energy via the local distribution network. In this case, BESS sizing is the first-stage decision, while the power dispatch is optimized after the uncertainty in renewable generation and demand is realized.

2SPs are usually solved by approximating the uncertainty with a discrete set of possible realizations, called scenarios. A monolithic 2SP formulation with copies of the recourse variables per scenario becomes computationally intractable as the scenario size grows [1]. Benders decomposition (BD) is a powerful tool that overcomes this problem by partitioning the model into more manageable chunks and solving them sequentially and in parallel whenever possible [2]. In the BD framework, a large monolithic 2SP model is split into a master problem containing first-stage variables and subproblems corresponding to scenarios. Although the model split helps with computation, solving the master problem and then the subproblems (SPs) iteratively to reach convergence drastically slows the process.

In recent years, much effort has been devoted to developing acceleration methods that reduce the computation time for BD, such as those targeting the relaxed master problem (RMP) [3]–[5], while others aim to speed up the SP evaluation or cut generation [6]–[9]. Managing RMP growth via cut filtering [3], [4] or aggregation [5] is an effective approach that reduces total run time, especially when the RMP is computationally heavy. On the other hand, batch-wise scenario selection in each iteration [5], [7] and modeling the recourse problem/cut coefficients using neural networks (NNs) [1], [6], [10] are powerful methods to reduce scenario computation time. Although NNs have exhibited promising speed-ups in BD algorithm, they typically rely on nontrivial offline training runs, and the model performance may degrade in the event of structural changes to the problem. Batchwise computation of a selected set of scenarios allows the algorithm to run faster by avoiding the need to solve the entire set at each iteration.

This research was partially funded by the following projects: (a.) CETPartnership, the Clean Energy Transition Partnership under the 2023 joint call for research proposals, co-funded by the European Commission (GA N°101069750) and with the funding organizations detailed on https://cetpartnership.eu/funding-agencies-and-call-modules and (b.) 'Multidisciplinary Approaches and Software Technologies for Engagement, Recruitment and Participation in Innovative Energy Communities in Europe (MASTERPIECE)' funded by the European Union under the H2020 Innovation Framework Programme, project number 101096836

[a] Affiliated with Aix Marseille Université, LIS Lab CNRS UMR 7020, Campus Saint-Jérôme, 13013 Marseille, France.

[b] Affiliated with the School of Electrical and Electronic Engineering, The University of Sheffield, S1 3JD Sheffield, U.K. and Autonex Systems Limited, Coventry CV1 2NT, U.K.

email: saif.ahmad,seifeddine.benelghali@{lis-lab.fr}, hafiz.h.ahmed@ieee.org

A number of stochastic and random scenario selection-based BD algorithms with verified convergence guarantees have been proposed in [11], [12]. However, random scenario selection may result in worse performance than a deterministic baseline [8]. A comprehensive review of the acceleration methods used in BD framework is presented in [13]. Although the modern version of BD is usually implemented via a callback method, made possible by the RMP having integer variables, a bulk of these approaches readily translates to the more conventional approach, where the RMP and SPs are solved iteratively. We follow the latter implementation as our model is an LP.

In this work, we introduce an online learning-based adaptive hybrid BD (OLAH-BD) algorithm for solving a risk-averse OSP formulated as a 2SP. The contribution is threefold: **a.)** we propose a hybrid Benders master where one high-stress scenario is embedded explicitly in the RMP, while the remaining scenarios are mapped to SPs as usual, and solved batch-wise in parallel on cpu threads **b.)** we introduce a regime-based online scenario selector: the algorithm begins with full-sweep stabilization and training, then switches to exploitation using a logistic regression (LR) model and feasibility prioritizing logic, and **c.)** we combine adaptive scenario selection with optimality-cut filtering such that tail relevant scenarios are prioritised. Periodic full sweeps recover missed information and provide exact convergence certificates while additionally allowing online retraining of the LR model on a more relevant dataset when required. The online retraining step is especially useful, as it enables the algorithm to detect regime changes in the algorithm's behavior and recalibrate model weights based on new information, while simultaneously avoiding the computational burden of pretraining heavy models.

Remaining sections in the paper are organized as follows: Section II revisits the risk-averse 2SP framework, while the main contributions and proposed algorithm are discussed in Section III. Section IV presents the numerical study on an energy community focusing on collective self-consumption, followed by conclusion and outlook in Section V.

II. Problem Formulation

We briefly revisit the two-stage stochastic program (2SP) framework to build intuition for the risk-averse formulation. A 2SP separates the decision vector into here-and-now (first-stage) variables, selected before uncertainty is realized, and recourse (second-stage) variables, optimized for the observed scenario. Let $x \in \mathcal{X},\ y_s \in \mathcal{Y}_s$ denote the first and second-stage variables, respectively, where $s \in S$ denotes the observed scenarios, then the scenario recourse value is given by

$$\begin{aligned} Q_s(x) := \min_{y_s \in \mathcal{Y}_s} \quad & q_s^\top y_s \\ \text{s.t.} \quad & M y_s = m_s + W_s x, \\ & N y_s \le n_s + R_s x, \end{aligned} \tag{1}$$

and the risk-neutral model minimizes

$$\min_{x \in \mathcal{X}} \ c^\top x + \sum_{s \in \mathcal{S}} \pi_s Q_s(x), \tag{2}$$

where π_s is the probability of scenario s, $\{q_s, m_s, n_s, W_s, R_s\}$ are realisation of the random variables in each scenario while $\{M, N\}$ are considered to be deterministic for most cases. The risk-neutral formulation gives equal importance to all scenarios in the recourse objective function. However, the user might be exposed to higher recourse costs in some scenarios than others, and it might be essential to protect the operator under these extreme/tail scenarios, which is known as the risk-aware formulation.

In the risk-averse formulation, the expected value of the recourse function in (2) is replaced by a risk measure, which in our case is the conditional value-at-risk (CVaR) [14]. CVaR penalizes the tail of the recourse-cost distribution by considering the expectation over the worst-scenario set defined by a confidence parameter α. For a discrete scenario set $s \in S$, the CVaR based 2SP is defined as

$$\min_{x,\eta,\xi} \quad c^\top x + \eta + \frac{1}{1-\alpha} \sum_{s \in \mathcal{S}} \pi_s \xi_s \tag{3a}$$

$$\text{s.t.} \quad x \in \mathcal{X}, \tag{3b}$$

$$\xi_s \ge Q_s(x) - \eta, \qquad \forall s \in \mathcal{S}, \tag{3c}$$

$$\xi_s \ge 0, \qquad \forall s \in \mathcal{S}. \tag{3d}$$

Here η is the value-at-risk threshold and ξ_s represents the positive excess cost above η. It is to be noted that only scenarios satisfying $Q_S(x) > \eta$, i.e., those in the upper tail above the value-at-risk threshold, contribute directly to the CVaR objective at a given first-stage point, while the recourse feasibility must hold for all scenarios. This problem structure is the primary motivation for building a solution that gives equal weight to all infeasible scenarios but prioritizes optimal scenarios that are more likely to be in or near the CVaR tail.

In the numerical study carried out in Section IV, $Q_s(x)$ represents the CVAR of the optimal annual dispatch cost under PV-load scenarios $s \in S$. The first-stage vector x contains BESS sizing and subscription decisions, while y_s contains the scenario-dependent directed power-flow variables. These application-specific variables and cost coefficients are defined in Section IV while the full recourse model is reported in the Appendix.

III. Online Learning-Based Adaptive Hybrid Benders

We present the online learning-based adaptive hybrid BD (OLAH-BD) for problem (3) in this section. The method combines three different elements to accelerate the BD algorithm without sacrificing accuracy, while having a tunable memory footprint: an explicit stress scenario embedded in the relaxed master problem (RMP) to avoid getting stuck in the infeasible region early on, an online trained logistic-regression (LR) scenario selector to avoid solving all the scenario problems (SPs) in each iteration, and periodic full-sweep for upper bound (UB) certification as well as for model retraining if needed. We discuss these elements in detail in the following subsections.

A. Hybrid Benders Decomposition

Vanilla BD (VBD) implementation of 2SP has a clean split where the first-stage (including CVaR, if applicable) variables and costs form the RMP, while the recourse counterparts form the SPs. In the hybrid BD, we select a stressful scenario to explicitly embed in the RMP as

$$s^e = \arg\max_{s\in\mathcal{S}} \psi_s, \tag{4}$$

where ψ_s is an aggregate score based on normalized annual deficit energy, peak deficit, and worst monthly deficit. The remaining scenarios form the decomposed set $\mathcal{D} := \mathcal{S} \setminus \{s^e\}$. Embedding scenario s^e automatically yields a much tighter master formulation with a realistic selection of x even during early iterations.

At each iteration k, the hybrid master contains the first-stage variables, the explicit recourse variables of scenario s^e, the CVaR variables, and one approximation variable θ_s for each decomposed scenario $s \in \mathcal{D}$. The hybrid master problem is formulated as

$$\min \quad c^\top x + \eta + \frac{1}{1-\alpha}\left(\pi_e \xi_e + \sum_{s\in\mathcal{D}} \pi_s \xi_s\right) \tag{5a}$$

$$\text{s.t.} \quad x \in X, \qquad y_e \in \mathcal{Y}_e(x), \tag{5b}$$

$$\xi_e \geq q_e^\top y_e - \eta, \qquad \xi_e \geq 0, \tag{5c}$$

$$\xi_s \geq \theta_s - \eta, \qquad \xi_s \geq 0, \qquad \forall s \in \mathcal{D}, \tag{5d}$$

$$\theta_s \geq a_{s\ell}^\top x + b_{s\ell}, \qquad \forall \ell \in \mathcal{K}_s^{\text{opt}},\ s \in \mathcal{D}, \tag{5e}$$

$$f_{s\ell}^\top x + h_{s\ell} \leq 0, \qquad \forall \ell \in \mathcal{K}_s^{\text{feas}},\ s \in \mathcal{D}. \tag{5f}$$

Here $\mathcal{K}_s^{\text{opt}}$ and $\mathcal{K}_s^{\text{feas}}$ are the sets of optimality and feasibility cuts generated for scenario s. The coefficients $(a_{s\ell}, b_{s\ell})$ are obtained from dual-optimal solutions of feasible recourse LPs and define lower approximations of $Q_s(x)$. The coefficients $(f_{s\ell}, h_{s\ell})$ are obtained from Farkas certificates and remove first-stage points for which the recourse LP is infeasible.

B. Learning Scenario Usefulness Online

Solving the master gives (x^k, η^k, θ^k) for the current iteration. Define the SP residual as

$$v_s^k = \left[Q_s(x^k) - \theta_s^k\right]_+, \tag{6}$$

and the CVaR-weighted residual as

$$r_s^k = \frac{\pi_s}{1-\alpha}\,\mathcal{B}_s^k\, v_s^k,\ \text{ where } \mathcal{B}_s^k := \begin{cases} 1, & Q_s(x^k) > \eta^k, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The LR selector uses a set of five features for each scenario. For the first feature $\widehat{v}_s^k$ defined as pre-solve residual proxy, let $\ell_s(k) < k$ be the most recent iteration at which scenario s was solved. We define

$$\widehat{v}_s^k = \left[Q_s(x^{\ell_s(k)}) - \theta_s^k\right]_+, \tag{8}$$

where $Q_s(x^{\ell_s(k)})$ is the last observed recourse value of scenario s. This proxy is available before solving scenario s at iteration k and is therefore suitable for adaptive selection. The resulting feature vector is given by

$$\phi_s^k = \left[\gamma(\widehat{v}_s^k),\ \gamma(r_{s,\text{last}}),\ \tau_s^k,\ \gamma(\bar{r}_s^k),\ \gamma(\bar{f}_s^k)\right]^\top, \tag{9}$$

where $\gamma(v) := \ln(1+v)$. The logarithmic scaling reduces the influence of large residual values, while the EMA terms preserve recent tail, risk, and feasibility history.

An LR model assigns a usefulness score

$$p_s^k = \sigma\Big(w^\top \phi_s^k + b\Big), \qquad \sigma(z) = \frac{1}{1+\exp(-z)}, \tag{10}$$

where the score p_s^k is used for ranking scenarios before subproblem evaluation. During adaptive iterations, $Q_s(x^k)$, v_s^k, and r_s^k are known only for the scenarios that are actually solved. Therefore, the LR model uses the historical feature vector ϕ_s^k to estimate which unsolved scenarios are likely to be useful.

Training labels are generated during full sweeps, when all decomposed scenarios are solved. Let

$$m_\alpha = \lceil (1-\alpha)|\mathcal{D}| \rceil \tag{11}$$

be the empirical CVaR-tail size. A scenario is labeled useful if it generates a feasibility cut or belongs to the m_α scenarios with the largest r_s^k. The tail-relevant set at iteration k is defined as

$$\mathcal{T}_\alpha^k = \text{Top}_{m_\alpha}\left\{r_s^k :\ s \in \mathcal{D}\right\}, \tag{12}$$

where Top_{m_α} returns the m_α scenarios with the largest CVaR-weighted residuals. Let F_s^k be a binary indicator equal to one if scenario s generates a feasibility cut at iteration k. The learning label is then

$$\ell_s^k = \begin{cases} 1, & s \in \mathcal{T}_\alpha^k \text{ or } F_s^k = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Thus, a scenario is labeled useful if it is either CVaR-tail relevant or feasibility-critical.

The training set is updated on full sweeps as

$$\mathcal{H} \leftarrow \mathcal{H} \cup \left\{(\phi_s^k, \ell_s^k) : s \in \mathcal{D}\right\}. \tag{14}$$

The LR model is then trained or refreshed on $\mathcal{H}$, if needed. In subsequent adaptive iterations, scenarios are ranked by p_s^k, and the highest ranked scenarios are combined with feasibility, stress, stale, and random safeguards to form the active set. This design keeps the learning task lightweight in the sense that the model learns only to prioritize scenario solves, while full sweeps remain responsible for validation and exact certification.

C. Adaptive active set, cut filtering and full sweeps

At most iterations, the algorithm solves only a subset $\mathcal{A}_k \subset \mathcal{D}$ of the decomposed scenarios. Let K_k denote the base adaptive batch size at iteration k. The active set is constructed as

$$\mathcal{A}_k = \mathcal{M}_k \cup \mathcal{A}_k^{\text{LR}} \cup \mathcal{A}_k^{\text{stress}} \cup \mathcal{A}_k^{\text{stale}} \cup \mathcal{A}_k^{\text{rand}}, \tag{15}$$

where $\mathcal{M}_k$ is the mandatory set of recently infeasible scenarios. After inserting $\mathcal{M}_k$, the remaining scenario slots $K_k - |\mathcal{M}_k|$ are split into LR exploitation and exploration blocks. Approximately 70% of the remaining slots are assigned to the highest-ranked scenarios according to the LR score in (10), 10% to the highest-stress scenarios selected based on (4), 10% to stale scenarios with the largest time since last evaluation given as $a_s^k = k - \ell_s(k)$., and the remaining slots to uniformly sampled scenarios. All blocks are selected without replacement after removing scenarios already included in $\mathcal{M}_k$. If rounding or unavailable candidates leave unused slots, the remaining positions are filled using the LR/heuristic ranking.

Feasibility cuts are never filtered. If a solved scenario is infeasible, its feasibility cut is added to the master and the scenario is placed in $\mathcal{M}_{k+1}$. Optimality cuts are filtered based on their potential relevance to the tail of the distribution. A solved feasible scenario first has to produce a significant master-underestimation residual, i.e., $Q_s(x^k) - \theta_s^k$ must exceed the cut tolerance. This reduces master growth while retaining cuts that are most relevant to the CVaR objective and to recourse feasibility.

At full sweep, all scenarios in $\mathcal{D}$ are evaluated and the exact upper bound at x^k is computed as

$$\overline{z}(x^k) = c^\mathsf{T} x^k + \eta^k + \frac{1}{1-\alpha} \sum_{s \in \mathcal{S}} \pi_s \left[Q_s(x^k) - \eta^k \right]_+ . \quad (16)$$

The algorithm can certify convergence only when the full set of scenarios is evaluated, due to the lack of an upper bound. Thus, learning affects only which subproblems are solved during adaptive iterations while full sweeps recover missed cuts, update the learning data, and provide the final certificate.

The algorithm follows a regime-based schedule. Iteration $k = 1$ is a stabilization full sweep. Iterations $k = 2, 3$ are also full sweeps and are used to collect the initial training data for the LR selector. From iteration $k = 4$ onward, adaptive LR-guided selection is used. Periodic full sweeps are performed every five iterations for validation and certification.

The nominal batch size is selected as approximately twice the CVaR-tail size,

$$K_0 \approx 2m_\alpha, \qquad m_\alpha = \lceil (1-\alpha)|\mathcal{D}| \rceil . \quad (17)$$

For $\alpha = 0.90$, this corresponds to roughly 20% of the decomposed scenario set. In the reported runs, $K_0 = 20$ for $|\mathcal{D}| = 99$ and $K_0 = 100$ for $|\mathcal{D}| = 499$.

The batch size is adapted online. If a full sweep indicates poor selector quality, defined by tail recall below 0.70, risk coverage below 0.60, missed feasibility cuts, or a large number of missed cuts, the LR model is retrained and the next batch is expanded as

$$K_{k+1} = \min \left\{ K_{\max}, \max \left(K_k + 10, \lceil 1.5 K_k \rceil \right) \right\} . \quad (18)$$

If the fraction of infeasible solved subproblems is at least 0.80, the next batch is expanded more aggressively,

$$K_{k+1} = \min \left\{ K_{\max}, 2K_k \right\} . \quad (19)$$

TABLE I: Case-study and model parameters

Group	Parameter	Value
Data	Sites / horizon	4 sites / 8760 h, $\Delta t = 1$ h
	Load source	RTE France annual load, scaled
	Annual load	690.4 MWh
	Load shares	$[0.30, 0.30, 0.20, 0.20]$
	PV source	CAMS GHI + Open-Meteo temp.
	PV location / area	Site 1 / 2000 m^2
	Scenario model	VAR–KDE residual sampling
	Scenario probabilities	Uniform, $\pi_s = 1/|\mathcal{S}|$
BESS	c^E, c^P	40 EUR/kWh-y, 20 EUR/kW-y
	η^{ch}, η^{dis}	0.95, 0.96
	$\rho^{\min}$, $\rho^{\max}$	0.10, 0.90
	ρ_i^0	$[0.757, 0.217, 0.630, 0.166]$
	$[\underline{h}, \overline{h}]$	$[1, 10]$ h
	c^{deg}	0.005 EUR/kWh throughput
Grid/ACC	λ_t^{buy}	HP/HC: 0.2 / 0.15 EUR/kWh
	λ^{exp}	0.05 EUR/kWh
	τ_t^{sh}	0.0123 EUR/kWh
	c^{sh}, c^{ret}	12.12, 31.66 EUR/kW-y
	$\overline{G}_i^{\mathrm{sh}}$, $\overline{G}_i^{\mathrm{ret}}$	100 kW
	c^{sl}	5.0 EUR/kWh unserved load
Bounds	$\overline{P}_i$	$\max\{\max r, \max d, 1\}$ kW
	$\overline{E}_i$	$10\overline{P}_i$ kWh

If the infeasible fraction is at most 0.30 for two consecutive adaptive iterations, the batch size is reduced toward its nominal value,

$$K_{k+1} = \max \left\{ K_0, \lceil 0.7 K_k \rceil \right\} . \quad (20)$$

This makes the learning policy conservative when feasibility or tail selection is poor, and computationally lean once the selector stabilizes.

IV. Numerical Simulation

A. Data and brief model description

The proposed method is evaluated on the BtM BESS sizing problem described in Appendix B. The case study is a four-site residential energy community over a one-year hourly horizon. The base load profile is obtained from annual RTE load data for France, converted to hourly resolution, scaled to an annual community consumption of 690.4 MWh (roughly 100 households of mixed sizes), and distributed across the four sites using fixed load shares. PV generation is assigned to site 1 only and is computed for Marseille, France, using CAMS irradiance and Open-Meteo temperature data with a PV averaged power model. The joint PV-load scenarios are generated using the VAR–KDE residual model in Appendix A.

The main data, technical limits, and economic coefficients used in the simulation are reported in Table I. These values instantiate the symbols appearing in Appendix B.

B. Scenario cases and algorithm settings

Two scenario sizes, S100 and S500, are considered to illustrate the scalability of the proposed approach, each having 1 explicit scenario in the master. Table II summarizes the decomposition, learning, and certification settings.

TABLE II: Scenario and algorithmic settings

Setting	S100	S500
$\lvert\mathcal{S}\rvert$	100	500
Explicit scenarios	1	1
$\lvert\mathcal{D}\rvert$	99	499
α	0.90	0.90
m_α	10	50
K_0	20	100
$K_0/\lvert\mathcal{D}\rvert$	20.2%	20.0%
$K_{\max}$	99	250
$K_{\max}/\lvert\mathcal{D}\rvert$	100%	50.1%
Warm-up full sweeps	3	3
Certification period	5 iterations	5 iterations
Active-set split	70% LR, 10% stale, 10% stress, rest random	
Retrain thresholds	tail recall < 0.70, risk coverage < 0.60	
Batch expansion	validation failure or infeasibility fraction ≥ 0.80	
Batch shrinkage	infeasibility fraction ≤ 0.30 for 2 iterations	
Cut tolerances	10^{-1} optimality, 10^{-4} feasibility	
Termination tolerance	10^{-3} certified relative gap	
Solver	Gurobi 13.0.1	
Parallelization	10 SP workers, 1 SP thread, 5 master threads	

TABLE III: Performance of the proposed method

Metric	S100, K20	S500, K100
Decomposed scenarios	99	499
CVaR tail size	10	50
Certified iteration	25	25
Certified LB (EUR)	77,533.97	78,818.85
Certified UB (EUR)	77,534.02	78,850.72
Certified gap	0.000061%	0.0404%
Baseline cost (EUR)	89,385.72	90,272.54
Savings vs. baseline	13.26%	12.65%
PV-site $E_1^\star$ (kWh)	65.85	57.43
PV-site $P_1^\star$ (kW)	13.86	13.00
BESS duration $E_1^\star/P_1^\star$ (h)	4.75	4.42
Scenario evaluations	1,176	6,022
Full-BD equivalent	2,475	12,475
Subproblem reduction	52.5%	51.7%
Candidate opt. cuts	1,132	6,022
Filtered opt. cuts	338 (29.9%)	1,658(27.5%)
Feasibility cuts	26	0
Runtime RMP	244 s	604 s
Runtime SPs	267 s	1372 s
Peak RAM use	8220 MB	9360 MB

The nominal batch size is selected to be approximately twice the CVaR tail size, corresponding to about 20% of the decomposed scenario set for both cases (corresponding to $\alpha = 0.9$). From iteration 4 onward, adaptive LR-guided selection is used between certification sweeps. Batch size regulation is triggered when full-sweep diagnostics show poor tail capture or when many solved subproblems are infeasible, whereas shrinkage is allowed only after persistent low infeasibility.

C. Performance Measure

Table III reports the certified results and the resulting BESS sizing decisions, with both cases terminating at iteration 25. The S100 case reaches a certified gap of 0.000061%, while the S500 reaches a certified gap of 0.0404%. The optimal storage capacity is concentrated at the PV site in both cases, which is to be expected given the potential benefit of self-consumption without being exposed to network charges, with approximately 65.85 kWh / 13.86 kW for S100 and 57.43 kWh / 13.00 kW for S500. The corresponding storage durations are about 4.75 h and 4.42 h, respectively.

The adaptive strategy also preserves its computational benefit when the scenario set scales from 100 to 500. The S100 case solves 1176 scenario subproblems instead of the 2475 evaluations required by full multi-cut Benders up to the same iteration, corresponding to a 52.5% reduction, whereas the S500 case solves 6022 scenario subproblems instead of 12475, corresponding to a 51.7% reduction.

Fig. 1a shows the lower-bound progression and the certified upper-bound values. The upper bound is plotted only at full sweeps due to the dependency on evaluating all decomposed scenarios. The initially loose lower bound is tightened rapidly during the warm-up full sweeps, after which both cases converge toward their certified upper bounds. Fig. 1b reports the number of added and filtered optimality cuts. The 500-scenario case naturally generates more candidate cuts, but the filtering mechanism discards a substantial fraction of them, especially at full sweeps, thereby limiting master growth.

D. Selector diagnostics and batch control

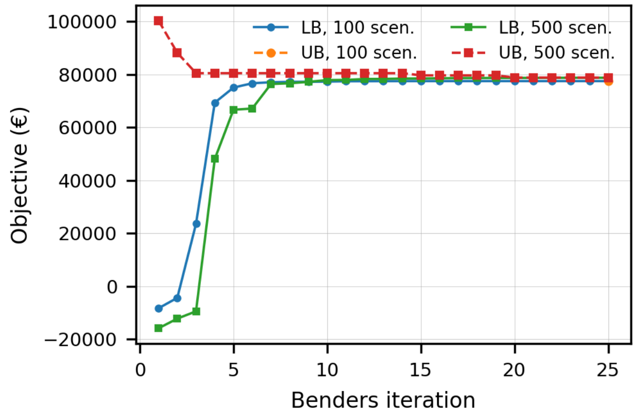
The learning component is evaluated at full sweeps, where all decomposed scenarios are solved and the true CVaR-tail set is known. Tail recall measures the fraction of true tail scenarios captured by the adaptive batch, while risk coverage measures the fraction of the total CVaR-weighted residual captured by the selected scenarios.

Fig. 2a shows that the S100 selector becomes reliable after retraining and reaches full tail recall and full risk coverage from iteration 15 onward. The behavior of S500 case is more difficult to capture in the initial training phase and the LR-based selector quality degrades at iteration 20, where both recall and risk coverage drop. However, the accuracy improves drastically after iteration 20 due to model retraining and temporary batch expansion.

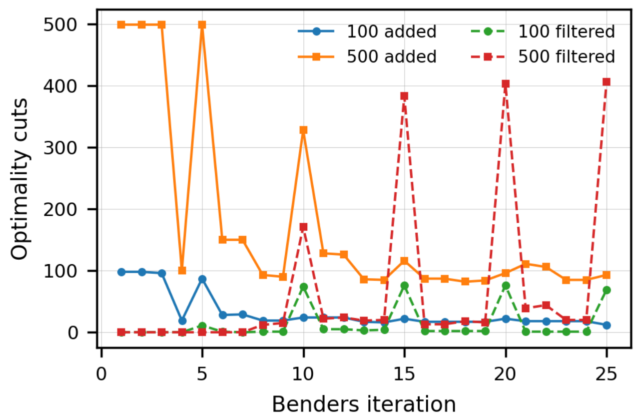
Fig. 2b shows the corresponding batch-size management normalized by the empirical CVaR-tail size, K_k/m_α. The nominal value is approximately 2, since the adaptive batch is chosen as about twice the CVaR-tail size. Temporary increases to 3 correspond to expanding the batch from 100 to 150 in the S500 case, i.e., from 20.0% to 30% of the decomposed scenario set. Once the selector stabilizes, the batch size returns to the nominal level. This confirms that learning is used conservatively in the sense that it reduces subproblem evaluations during stable regimes, while full sweeps detect selector failure, trigger retraining, and recover missed information.

V. Conclusions and Outlook

This paper presents an accelerated online learning-based adaptive hybrid Benders approach for risk-averse optimal sizing of BtM BESS in energy communities. The main objective of this work is to demonstrate the capability of a lightweight ML model, such as logistic regression, to learn SP significance in the CVaR tail from selected features and use this to predict their significance in a risk-based problem formulation. We illustrate that retraining the model during regime changes

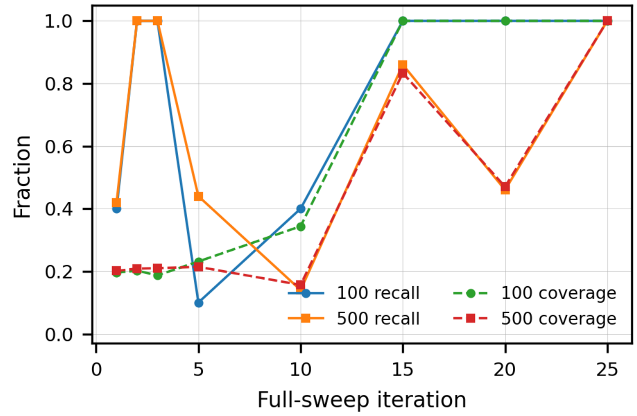


(a) Certified lower and upper bounds.

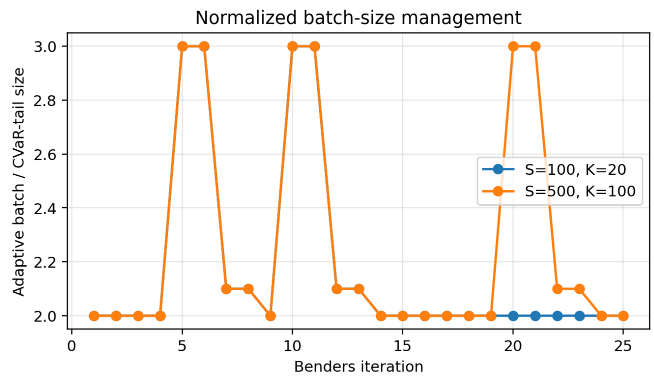


(b) Added and filtered optimality cuts.

Fig. 1: Convergence and cut-management behavior for the 100- and 500-scenario cases. Upper bounds are available only at full sweeps, where all decomposed scenarios are evaluated. Cut filtering reduces master growth by discarding optimality cuts that are not immediately CVaR-relevant.



(a) Tail recall and risk coverage.



(b) Adaptive batch normalized by CVaR-tail size.

Fig. 2: Learning diagnostics and batch-size control. Tail recall measures the fraction of true CVaR-tail scenarios captured by the adaptive policy, while risk coverage measures the fraction of tail-weighted residual captured by the selected scenarios. The normalized batch size K_k/m_α shows how the algorithm expands the adaptive batch after validation failures and shrinks it back toward the nominal level.

improves its selection capabilities and that it scales well across scenarios despite the model's small size. Current limitations of the work, such as the LP formulation of SPs and RMP, can be addressed using different BD architectures, such as logic-based BD or column-and-constraint generation, within a robust optimization framework. A byproduct of the LP formulation of RMP is that the algorithm is implemented iteratively rather than via callbacks, which is only possible for MILP or MIP RMP. It would be interesting to explore implementing the proposed approach for MILP RMP using callbacks.

Appendix A
Scenario Generation

The uncertainty set consists of joint PV-load trajectories. Historical PV and load data are decomposed into a deterministic calendar baseline and a residual process. The PV baseline is computed by month and hour, while the load baseline is computed by month, day type, and hour. The residual vector is then modeled using a vector autoregressive process (VAR) with kernel density estimator (KDE) based sampled innovations

$$\epsilon_t = c + \sum_{\ell=1}^{p} A_\ell \epsilon_{t-\ell} + u_t, \qquad u_t \sim \widehat{f}_u^{\mathrm{KDE}}. \tag{21}$$

Separate VAR–KDE models are fitted by calendar bucket. Simulated residual paths are added back to the deterministic baseline and clipped to enforce physical limits, including nonnegative load, nonnegative PV, and zero PV during zero-baseline periods. Details are omitted due to space limitations, but the developed algorithm is expected to perform well in a similar setting with any other scenario-generation approach and is independent of this part of the work.

Appendix B
Modeling Details for the Optimal Sizing Problem

For the considered BtM BESS OSP in an EC, the first-stage feasible set $\mathcal{X}$ used in (3) is defined as

$$0 \le E_i \le \overline{E}_i, \qquad 0 \le P_i \le \overline{P}_i, \qquad \forall i \in \mathcal{I}, \tag{22a}$$

$$\underline{h} P_i \le E_i \le \overline{h} P_i, \qquad \forall i \in \mathcal{I}, \tag{22b}$$

$$0 \le \bar{g}_i^{\mathrm{sh}} \le \overline{G}_i^{\mathrm{sh}}, \qquad 0 \le \bar{g}_i^{\mathrm{ret}} \le \overline{G}_i^{\mathrm{ret}}, \qquad \forall i \in \mathcal{I}. \tag{22c}$$

For fixed first-stage decision x and scenario s, the recourse function $Q_s(x)$ is the optimal value of the following linear dispatch problem. Let $\mathcal{T}^- = \{1, \ldots, |\mathcal{T}| - 1\}$.

$$\begin{aligned} Q_s(x) = \min_{y_s \ge 0} \quad & \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \Big[\lambda_t^{\mathrm{buy}} p_{its}^{\mathrm{g,l}} + \tau_t^{\mathrm{sh}} \left(p_{its}^{\mathrm{sh,l}} + p_{its}^{\mathrm{sh,bat}} \right) \\ & + c^{\mathrm{deg}} \left(p_{its}^{\mathrm{pv,bat}} + p_{its}^{\mathrm{sh,bat}} + p_{its}^{\mathrm{bat,l}} + p_{its}^{\mathrm{bat,sh}} \right) \\ & + c^{\mathrm{sl}} p_{its}^{\mathrm{sl}} - \lambda^{\mathrm{exp}} p_{its}^{\mathrm{pv,g}} \Big] \Delta t \end{aligned}$$

subject to

$$p_{its}^{\mathrm{pv,bat}} + p_{its}^{\mathrm{pv,l}} + p_{its}^{\mathrm{pv,g}} + p_{its}^{\mathrm{pv,sh}} + p_{its}^{\mathrm{curt}} = r_{its}, \qquad \forall i, t, \tag{23}$$

$$p_{its}^{\mathrm{bat,l}} + p_{its}^{\mathrm{pv,l}} + p_{its}^{\mathrm{sh,l}} + p_{its}^{\mathrm{g,l}} + p_{its}^{\mathrm{sl}} = d_{its}, \qquad \forall i, t, \tag{24}$$

$$\begin{aligned} e_{i,t+1,s} = & e_{its} + \eta^{\mathrm{ch}} \Delta t \left(p_{its}^{\mathrm{pv,bat}} + p_{its}^{\mathrm{sh,bat}} \right) \\ & - \frac{\Delta t}{\eta^{\mathrm{dis}}} \left(p_{its}^{\mathrm{bat,l}} + p_{its}^{\mathrm{bat,sh}} \right), \qquad \forall i, \ t \in \mathcal{T}^-, \end{aligned} \tag{25}$$

$$e_{i1s} = \rho^0 E_i, \qquad e_{i|\mathcal{T}|s} = e_{i1s}, \qquad \forall i, \tag{26}$$

$$\rho^{\min} E_i \le e_{its} \le \rho^{\max} E_i, \qquad \forall i, t, \tag{27}$$

$$p_{its}^{\mathrm{pv,bat}} + p_{its}^{\mathrm{sh,bat}} \le P_i, \qquad \forall i, t, \tag{28}$$

$$p_{its}^{\mathrm{bat,l}} + p_{its}^{\mathrm{bat,sh}} \le P_i, \qquad \forall i, t, \tag{29}$$

$$\sum_{i \in \mathcal{I}} \left(p_{its}^{\mathrm{pv,sh}} + p_{its}^{\mathrm{bat,sh}} \right) = \sum_{i \in \mathcal{I}} \left(p_{its}^{\mathrm{sh,l}} + p_{its}^{\mathrm{sh,bat}} \right), \qquad \forall t, \tag{30}$$

$$p_{its}^{\mathrm{sh,l}} + p_{its}^{\mathrm{sh,bat}} \le \bar{g}_i^{\mathrm{sh}}, \qquad \forall i, t, \tag{31}$$

$$p_{its}^{\mathrm{pv,sh}} + p_{its}^{\mathrm{bat,sh}} \le \bar{g}_i^{\mathrm{sh}}, \qquad \forall i, t, \tag{32}$$

$$p_{its}^{\mathrm{g,l}} \le \bar{g}_i^{\mathrm{ret}}, \qquad \forall i, t. \tag{33}$$

Constraints (23) and (24) allocate PV generation and serve site demand, (25)–(29) model BESS operation and couple recourse decisions to the first-stage BESS size, (30) enforces balance of the shared-energy pool, while (31)–(33) enforce the subscribed-capacity decisions.

References

[1] R. M. Patel, J. Dumouchelle, E. Khalil, and M. Bodur, "Neur2sp: Neural two-stage stochastic programming," *Advances in neural information processing systems*, vol. 35, pp. 23 992–24 005, 2022.

[2] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

[3] H. Jia and S. Shen, "Benders cut classification via support vector machines for solving two-stage stochastic programs," *INFORMS Journal on Optimization*, vol. 3, no. 3, pp. 278–297, 2021.

[4] H. Cai and X. Yu, "Learning to cut: Reinforcement learning for benders decomposition," *arXiv preprint arXiv:2605.06516*, 2026.

[5] C. Ramírez-Pico, I. Ljubić, and E. Moreno, "Benders adaptive-cuts method for two-stage stochastic programs," *Transportation Science*, vol. 57, no. 5, pp. 1252–1275, 2023.

[6] C. Guan, E. M. E. Raqabi, M. Tanneau, and P. Van Hentenryck, "The proxy benders decomposition," *arXiv preprint arXiv:2606.07403*, 2026.

[7] X. Blanchot, F. Clautiaux, B. Detienne, A. Froger, and M. Ruiz, "The benders by batch algorithm: design and stabilization of an enhanced algorithm to solve multicut benders reformulation of two-stage stochastic programs," *European Journal of Operational Research*, vol. 309, no. 1, pp. 202–216, 2023.

[8] T. Donkiewicz, "Adaptive subproblem selection in benders decomposition for survivable network design problems," *arXiv preprint arXiv:2604.09031*, 2026.

[9] T. L. Magnanti and R. T. Wong, "Accelerating benders decomposition: Algorithmic enhancement and model selection criteria," *Operations Research*, vol. 29, no. 3, pp. 464–484, 1981.

[10] Y. Liu and F. Oliveira, "Icnn-enhanced 2sp: Leveraging input convex neural networks for solving two-stage stochastic programming," *arXiv preprint arXiv:2505.05261*, 2025.

[11] D. Bertsimas, R. Cory-Wright, J. Pauphilet, and P. Petridis, "A stochastic benders decomposition scheme for large-scale stochastic network design," *INFORMS Journal on Computing*, vol. 37, no. 5, pp. 1163–1181, 2025.

[12] J. Pauphilet and Y. Wu, "The surprising performance of random partial benders decomposition," *Optimization Online*, vol. 32181, 2025.

[13] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "The benders decomposition algorithm: A literature review," *European Journal of Operational Research*, vol. 259, no. 3, pp. 801–817, 2017.

[14] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 2, no. 3, pp. 21–41, 2000.