Accelerating L-shaped Two-stage Stochastic SCUC with Learning Integrated Benders Decomposition

Fouad Hasan, *Student Member, IEEE*, Amin Kargarian, *Senior Member, IEEE*,

***Abstract*—Benders decomposition is widely used to solve large mixed-integer problems. This paper takes advantage of machine learning and proposes a variant of Benders decomposition to tackle two-stage stochastic security-constrained unit commitment (SCUC). The problem is decomposed into a master problem and subproblems corresponding to individual load scenarios. The primary objective is to mitigate computational expenses and memory consumption associated with Benders decomposition by generating tighter cuts and reducing the master problem's dimensions. A regressor reads load profile scenarios and predicts objective function proxy values for the subproblems, enabling the creation of tighter cuts for the master problem. The numerical difference between cut values and proxy variable values serves as the basis for identifying useful cuts. Analytical cut-filtering and classification-assisted cut-filtering approaches are discussed and compared. Useful cuts contain the necessary information to form the feasible region and are iteratively added to the master problem, whereas non-useful cuts are discarded, thus reducing the computational burden at each Benders iteration. Simulation studies conducted across various test systems demonstrate the efficacy of the proposed learning-enhanced Benders decomposition in solving two-stage SCUC problems, showcasing superior performance compared to conventional multi-cut Benders decomposition and offering numerical advantages over cut classifier-based Benders approaches.**

***Index Terms*—Stochastic unit commitment, Benders decomposition, useful cuts, machine learning.**

NOMENCLATURE

Indices, Sets, and Parameters:

δ	Cut filtering criterion.
Δ_g	Ramping limit for post contingency re-dispatch of unit g .
ϵ	Duality gap tolerance limit.
$\eta\{\cdot\}$	Random parameter.
ϕ	List of filtered cuts.
π_ω	Probability of stochastic scenario ω.
$\psi\{\cdot\}$	Numerical value of a cut.
b	Accumulated cuts.
$D_{t,\omega}$	Nodal power demand at time t under scenarios ω.
$f\{\cdot\}$	Generation cost function.
g, t, s, w, c	Indices for generators, time horizon, sample demand scenarios, subsample of each demand scenario, and contingency.
J_{MP}	Master problem objective function.
J_{SP}	Subproblem objective function.
k	Benders iteration index.
RU_g, RD_g	Ramp up and ramp down limits of unit g.
SF_c	Shift factor matrix under contingency c.
SU_g, SD_g	Startup and shutdown costs of unit g.
T_g^{on}, T_g^{off}	Minimum ON and OFF time limits of unit g.
UT_g, DT_g	Initial Up and down time of of unit g.

Variables:

α_ω	Proxy of subproblem ω objective function.
$\lambda_{g,t,\omega}$	Dual variables.
$\underline{\nu}, \overline{\nu}$	Auxiliary continuous variable.
$p_{g,t,\omega,c}$	Power generated by unit g at time t under demand scenarios ω and contingency c.
$p_{g,t,\omega}$	Power generated by unit g at time t under demand scenarios ω and no contingency condition.
$P_{t,\omega,c}^{inj}$	Net nodal power injection matrix.
$u_{g,t}$	ON/OFF status of generating unit g.
$y_{g,t}, z_{g,t}$	Startup and shutdown binary variable indicators.

This work was supported by National Science Foundation under Grant ECCS-1944752.

F. Hasan and A. Kargarian are with the Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803 USA (e-mail: fhasan1@lsu.edu, kargarian@lsu.edu).

I. INTRODUCTION

A. Background

SECURITY-constrained unit commitment (SCUC) is a complex optimization problem solved daily to determine the optimal schedule of generating units [1]. SCUC is a mixed-integer program (MIP) that involves a mix of continuous and integer decision variables. The challenge with mixed integer problems is that the inclusion of integer variables often makes the problem NP-hard. SCUC becomes even more complex under input parameter (e.g., demand) uncertainty [2]–[4]. SCUC is often formulated as a stochastic MIP problem, known as an L-shaped optimization problem [5]. An L-shaped problem is formulated as a two-stage stochastic optimization that involves here-and-now and wait-and-see decision variables. It is a two-stage optimization problem with a single integer recourse decision in the second stage. This type of problem is commonly encountered in decision-making under uncertainty, where a decision-maker may face different scenarios or outcomes with associated probabilities. Due to the block structure, the L-shaped two-stage stochastic program is an effective method for SCUC [6].

L-shaped optimization problems can be solved using specialized techniques such as Benders decomposition. Although Benders decomposition solves L-shaped problems, it suffers from exponential worst-case computational complexity. This is due, in part, to the large number of successively added cuts over iterations, which causes the size of Benders' master problem to grow excessively. Moreover, the master problem takes over 90% of the time required to implement Benders decomposition [7]. Also, as noted in [8], not every extreme point in the feasible region of subproblems contributes equally to limiting the optimal solution to the master problem. This implies that a significant number of Benders cuts may not be tight enough at the final optimal solution. Consequently, these non-useful Benders cuts can make solving large-scale integer programs challenging.

Various mathematical and heuristic approaches have been presented in the literature to enhance Benders decomposition performance. This study aims to accelerate the convergence of Benders decomposition by taking advantage of machine learning and cut filtering techniques.

B. Literature Review

It has been more than 50 years since the development of the Benders decomposition algorithm by J. Bender (1962) [9]. The algorithm is designed to address complicating variables that, when temporarily fixed, simplify the problem significantly. It distributes the computational load between a master problem and a subproblem (SP). Benders decomposition has proven successful in various fields, including planning and scheduling, healthcare, transportation, telecommunications, energy and resource management, and chemical process design [6]. The primary application of Benders decomposition is initially focused on solving MIP problems. Once integer variables are fixed, the problem is converted into a continuous linear program, which can develop cuts using standard duality theory. Many enhancements have been made to extend Benders' applicability to a wider range of problems. As a result, Benders decomposition has been widely used to solve linear, nonlinear, integer, stochastic, multi-stage, and bilevel optimization problems [10].

The traditional implementation of Benders decomposition can be computationally expensive, time-consuming, and memory-intensive, with problems such as poor feasibility and optimality cuts, ineffective early iterations, and the zigzagging behavior of primal solutions [7]. Researchers have explored various strategies to speed up the convergence and reduce the number of iterations and time spent on each iteration. The master problem is usually solved using branch-and-bound, with the simplex approach used to solve the subproblem. However, a significant number of cuts generated do not contribute to convergence, leading to memory occupation [8]. To address this problem, improvement criteria are proposed to ensure that new and useful cuts are included in the master problem [11]. Additionally, researchers have observed several orders of improvement when using constraint programming to solve the master problem [12]. Column generation has been introduced to handle specific structures more effectively and achieve tighter constraints at the root node of the branch-and-bound tree [13], [14]. For large subproblems, decomposition, parallelization, and column generation have been used to reduce overall solution time. By adding valid inequalities to the master problem, one can significantly reduce the number of generated cuts and solution time [15]. Moreover, clustering subproblems can decrease the number of iterations [16].

Recently, researchers have been exploring the use of machine learning (ML) to alleviate the computational complexity of complex problems [17]–[19]. [20], [21] review the learning-assisted algorithm for power systems optimization problems. A survey of the application of ML to optimal power flow is provided in [22].

Learning-based algorithms can be categorized into three types: end-to-end learning (with label), unsupervised/reinforcement learning (no ground truth), and algorithm-specific hybrid learning, which leverages the specific structure of the target problem to accelerate the optimization process. We focus on ML applications to Benders decomposition, particularly on cut classifications. The literature in this field is limited. A support vector machine is used in [23] to construct a cut classifier that identifies valuable cuts in each Benders iteration, thus reducing the size of the master problem and shortening the solution time. Another strategy is to use the Lagrange multipliers of the Benders subproblem to aggregate the optimality cuts [24]. In [25], a cut classifier is used, and five features are designed to characterize the generated cuts [26].

C. Motivation

Reducing the computational burden of two-stage stochastic SCUC models enables system operators to adopt them for real-time and near-real-time decision-making. This is increasingly vital for stochastic scheduling in environments with high renewable energy penetration and significant uncertainties. Solving two-stage stochastic SCUC with Benders decomposition is still computationally challenging and needs further investigation to pave the road for the adaptation by systems operators. This problem has not been extensively studied, and there is a lack of ML-based Benders approaches to accelerate L-shaped two-stage stochastic SCUC. Generating a labeled dataset for the classifier requires significant computational costs, particularly in the context of multi-cut Benders decomposition. The SCUC problem with intertemporal constraints typically involves a high number of Benders iterations, resulting in a multiplicatively larger number of cuts. Determining the Lagrange multiplier for each cut requires re-solving the master problem after each Benders iteration [24], which makes the cut labeling process computationally intensive. Similarly, identifying the positive increment in the lower bound of the master problem requires re-solving the master problem for each individual cut [25], achieved by cumulatively adding them one after another following the compilation of all cuts upon convergence. This highlights the need for streamlined, learning-enhanced methods to expedite the Benders decomposition in solving L-shaped SCUC models.

D. Contribution

We present a combined machine learning and cut filtering approach to enhance Benders decomposition for tackling L-shaped two-stage stochastic SCUC challenges. This approach generates tighter cuts and drops unuseful cuts from the Benders master problem to reduce its computational complexity. Specifically, a regressor is trained to predict sub-problem objective proxy values, enabling the creation of a more constrained feasible region for the master problem. Following each Benders iteration, a criterion based on numerical distance is used to identify useful cuts. Subsequently, a truncated master problem is constructed by incorporating these useful cuts and discarding non-useful ones. By combining this regressor and cut filtering technique, the computational burden and memory requirements of Benders decomposition are significantly reduced. Moreover, the inherent iterative nature of Benders decomposition ensures the feasibility and optimality of the obtained results by regenerating any discarded potentially useful cut. The proposed accelerated Benders approach is evaluated across various test systems.

E. Paper Organization

The rest of the paper is structured as follows. The problem formulation is given in Section II. The proposed R-Benders approach is presented in Section III. The existing C-Benders and enhanced CR-Benders are discussed in Section IV. The numerical simulations are discussed in Section V, and concluding remarks are provided in Section VI.

II. Two-stage Stochastic SCUC

This study addresses a two-stage stochastic model for security-constrained unit commitment, where power demand is considered an uncertain variable and network security criteria are incorporated. The resulting formulation is structured as a MIP problem, which we solve using Benders decomposition.

A. Problem Formulation

The two-stage SCUC problem is formulated in (1), which includes two sets of variables pertaining to first-stage or here-and-now decisions and second-stage or wait-and-see decisions [27]. On/off status of generating units are first-stage decision variables that are made before the realization of uncertainty. Generation dispatches are second-stage decision variables that are made after the realization of actual power demand. The first term of (1a) is the first-stage startup and shutdown costs, and the second is the second-stage generation dispatch costs. The first-stage scenario-independent unit commitment constraints (1b)–(1i) model generators' on/off status and minimum up and down time. The second-stage scenario-dependent operational constraints, including generating unit production and ramp limitations and power flow equations, are formulated in (1j)–(1t). $N-1$ security constraints are included in the model with index c. Constraints (1j)-(1n) are formulated for every uncertainty scenario ω under normal conditions, i.e., no contingency, and (1o)-(1t) are formulated for every ω and every contingency condition c.

$$\min \sum_t \sum_g (SU_g y_{g,t} + SD_g z_{g,t}) + \sum_\omega \pi_\omega \sum_t \sum_g f(p_{g,t,\omega}) \tag{1a}$$

s.t.

$$y_{g,t} - z_{g,t} = u_{g,t} - u_{g,(t-1)} \quad \forall g, \forall t \tag{1b}$$

$$y_{g,t} + z_{g,t} \leq 1 \quad \forall g, \forall t \tag{1c}$$

$$\sum_{t=1}^{UT_g} (1 - u_{g,t}) = 0 \quad \forall g \tag{1d}$$

$$\sum_{\tau=t}^{t+T_g^{on}-1} u_{g,\tau} \geq T_g^{on} y_{g,t} \quad \forall g, t = UT_g + 1, UT_g + 2, ...T - T_g^{on} + 1 \tag{1e}$$

$$\sum_{\tau=t}^{T} (u_{g,\tau} - y_{g,t}) \geq 0 \quad \forall g, t = T - T_g^{on} + 2, ...T \tag{1f}$$

$$\sum_{t=1}^{DT_g} u_{g,t} = 0 \quad \forall g \tag{1g}$$

$$\sum_{\tau=t}^{t+T_g^{off}-1} (1 - u_{g,\tau}) \geq T_g^{off} z_{g,t} \quad \forall g, t = DT_g + 1, DT_g + 2, ...T - T_g^{off} + 1 \tag{1h}$$

$$\sum_{\tau=t}^{T} (1 - u_{g,\tau} - z_{g,t}) \geq 0 \quad \forall g, t = T - T_g^{off} + 2, ...T \tag{1i}$$

$$\Bigg\{$$

$$p_g^{min} u_{g,t} \leq p_{g,t,\omega} \leq p_g^{max} u_{g,t} \quad \forall g, \forall t \tag{1j}$$

$$p_{g,t,\omega} - p_{g,(t-1),\omega} \leq RU_g(1 - y_{g,t}) + P_g^{min} y_{g,t} \quad \forall g, \forall t \tag{1k}$$

$$p_{g,(t-1),\omega} - p_{g,t,\omega} \leq RD_g(1 - z_{g,t}) + P_g^{min} z_{g,t} \quad \forall g, \forall t \tag{1l}$$

$$\sum_g p_{g,t,\omega} = D_{t,\omega} \quad \forall t \tag{1m}$$

$$-PL^{max} \leq SFP_{t,\omega}^{inj} \leq PL^{max} \quad \forall t \tag{1n}$$

$$p_g^{min} u_{g,t} \leq p_{g,t,\omega,c} \leq p_g^{max} u_{g,t} \quad \forall g, \forall t \tag{1o}$$

$$p_{g,t,\omega,c} - p_{g,(t-1),\omega,c} \leq RU_g(1 - y_{g,t}) + P_g^{min} y_{g,t} \quad \forall g, \forall t \tag{1p}$$

$$p_{g,(t-1),\omega,c} - p_{g,t,\omega,c} \leq RD_g(1 - z_{g,t}) + P_g^{min} z_{g,t} \quad \forall g, \forall t \tag{1q}$$

$$\sum_{g} p_{g,t,\omega,c} = D_{t,\omega} \quad \forall t \tag{1r}$$

$$-PL^{max} \leq SF_c P^{inj}_{t,\omega,c} \leq PL^{max} \quad \forall t \tag{1s}$$

$$|p_{g,t,\omega} - p_{g,t,\omega,c}| \leq \Delta_g \quad \forall g, \forall t \tag{1t}$$

$$\Big\} \ \forall \omega, \forall c$$

$$x \in \{p_{g,t,\omega}, p_{g,t,\omega,c}, u_{g,t}, y_{g,t}, z_{g,t}, \alpha_\omega\}$$

B. Benders Decomposition

The computational complexity of mixed-integer program (1) increases with the increasing size of the network, number of demand scenarios, and number of contingencies. Benders decomposition is suitable for solving MIP problems with a block structure over contingencies and uncertainty scenarios [6]. We use a multi-cut variant of Benders decomposition that converges faster than the classical single-cut approach [24], [28]. The multi-cut technique decomposes a problem into multiple subproblems and generates multiple cuts in each iteration. This multi-cut generation improves convergence performance [29]. Problem (1) is decomposed into a master problem and Ω (the number of uncertainty scenarios) subproblems. The master problem (MP) at iteration k is formulated in (2a)-(2c).

$$\min_x \sum_t \sum_g (SU_g y_{g,t} + SD_g z_{g,t}) + \sum_\omega \alpha_\omega \tag{2a}$$

s.t.

$$(1b) - (1i) \tag{2b}$$

$$\cup_{i=1}^{i=k-2} h^i_\beta (u_{g,t}, \alpha_\omega) \tag{2c}$$

$$h^{k-1}_\beta (u_{g,t}, \alpha_\omega) : \alpha_\omega \geq J^{k-1}_{SP,\omega} + \sum_t \sum_g \lambda^{k-1}_{g,t,\omega} (u_{g,t} - u^{k-1}_{g,t}) \ \ \forall \omega \tag{2d}$$

$$\alpha_\omega \geq \alpha^{min}_\omega \quad \forall \omega \tag{2e}$$

$$x \in \{u_{g,t}, y_{g,t}, z_{g,t}, \alpha_\omega\}$$

The objective function (2a) consists of startup and shutdown costs and a term as a proxy for the subproblems' objective function. Equation (2c) is the accumulation of all cuts generated up to iteration $k-2$, (2d) denotes Benders cuts generated at iteration $k-1$, and (2e) defines the bound on proxy variables where α^{min}_ω is set as a large negative constant.

Benders subproblem ω (SP_ω) is formulated in (3a)-(3f). Given the unit on/off statuses predetermined by the master problem, a subproblem could encounter infeasibility due to insufficient generation capacity to meet the load requirements. To circumvent this issue, we use an 'always-feasible' subproblem model [30]. This model ensures that, regardless of the constraints set by the master problem, the subproblems remain feasible. This model incorporates non-negative slack variables $\underline{\nu}$ and $\overline{\nu}$ to mitigate infeasibility in SP_ω. Auxiliary variables $\underline{\nu}$ and $\overline{\nu}$ relax generation capacity constraints (3d) and (3e). h_g are large positive constants imposing a high penalty on the auxiliary variables.

$$\Big\{ \min_x \sum_t \sum_g \big(\pi_\omega f(p_{g,t,\omega}) + h_g(\underline{\nu}_{g,t,\omega} + \overline{\nu}_{g,t,\omega})\big) \tag{3a}$$

s.t.

$$u_{g,t} = u^*_{g,t} : \lambda_{g,t,\omega} \tag{3b}$$

$$\Big\{$$

$$(1k) - (1n) \& (1p) - (1t) \tag{3c}$$

$$(p^{min}_g - \underline{\nu}_{g,t,\omega}) u_{g,t} \leq p_{g,t,\omega} \leq (p^{max}_g + \overline{\nu}_{g,t,\omega}) u_{g,t} \quad \forall g, \forall t \tag{3d}$$

$$(p^{min}_g - \underline{\nu}_{g,t,\omega}) u_{g,t} \leq p_{g,t,\omega,c} \leq (p^{max}_g + \overline{\nu}_{g,t,\omega}) u_{g,t} \quad \forall g, \forall t \tag{3e}$$

$$\underline{\nu}_{g,t,\omega} \geq 0, \overline{\nu}_{g,t,\omega} \geq 0 \tag{3f}$$

$$\Big\} \ \forall c \ \Big\} \ \forall \omega$$

$$x \in \{p_{g,t,\omega}, p_{g,t,\omega,c}, \underline{\nu}_{g,t,\omega}, \overline{\nu}_{g,t,\omega}\}$$

The objective function (3a) consists of generation costs and constraint violation penalty modeled by $h_g(\underline{\nu}_{g,t,\omega} + \overline{\nu}_{g,t,\omega})$. Constraint (3b) sets the first stage decisions, i.e., generator unit status, as fixed values received from the master problem. Inequalities (3f) set bound on auxiliary variables. The MP and SPs are solved iteratively until convergence tolerance ε is smaller than a predetermined threshold.

$$\epsilon = |\frac{UB - LB}{LB}| \tag{4}$$

where

$$LB = J_{MP}, \quad UB = J_{SP} + J_{MP} - \sum_\omega \alpha_\omega$$

The two-stage stochastic SCUC with multi-cut Benders is summarized in Algorithm I.

Algorithm 1 Two-stage stochastic SCUC with multi-cut Benders

Initialize convergence tolerance ϵ, set iteration index $k = 0$, and set α^{min}_ω
while $\epsilon > tolerance$ **do**
  $k \leftarrow k + 1$
  if $k > 1$ **then**
    Form new cut, $\alpha_\omega \geq J^{k-1}_{SP,\omega} + \sum_t \sum_g \lambda^{k-1}_{g,t,\omega} (u_{g,t} - u^{k-1}_{g,t}) \forall \omega$
  end if
  Solve master problem (2) to obtain $u^k_{g,t}$
  for $n = 1 : \omega$ **do**
    Solve SP_ω to obtain $J_{SP,\omega}$ and $\lambda_{g,t,\omega}$
  end for
  Calculate convergence tolerance, $\epsilon = |\frac{UB-LB}{LB}|$
end while

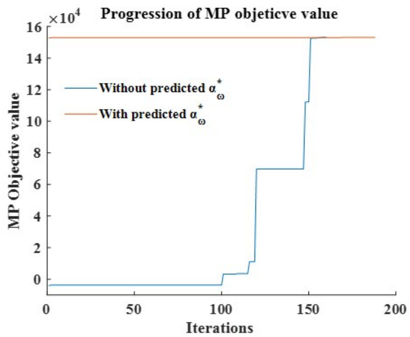

Fig. 1. Benders convergence using α^*_ω $\forall$ ω in inequality (2b).

III. Proposed R-Benders

While the multi-cut Benders method outperforms single-cut Benders in terms of convergence performance, its memory usage and computational overhead still need enhancement. We present a learning-aided approach to reduce the computational costs of multi-cut Benders decomposition for solving two-stage stochastic SCUC. We mainly focus on the master problem, which has a higher computational cost than subproblems.

The proposed approach uses a combination of a regression learner to predict a proxy variable for each subproblem ω and an analytical cut filtering approach to drop non-useful cuts from the master problem at each iteration. The regressor forms a tighter bound for the master problem at the root node starting from $k = 1$. We predict the optimal subproblem proxy variables α^*_ω, i.e., α_ω upon the convergence of Algorithm I. Knowing α^*_ω, or even a good approximation, makes inequalities (2c), (2d), and (2e) tighter and the lower bound LB closer to the optimal subproblem cost. The analytical cut filtering approach identifies and drops non-useful cuts during each subsequent Benders iteration to reduce the size of the master problem. Non-useful cuts do not affect the master problem feasible region.

A. SP Objective Proxy Prediction

In the multi-cut Benders formulation (2), α_ω is a variable and α^{min}_ω is a bound where $\alpha_\omega \geq \alpha^{min}_\omega$ constitutes a master problem constraint. The value of α_ω increases after each iteration and reaches its optimal value α^*_ω upon convergence. The value of α^{min}_ω can be selected by analyzing the physical and economic aspects of the SCUC problem. Setting a suitable α^{min}_ω reduces the search space and enhances Benders convergence performance. An ideal case is to set α^{min}_ω corresponding to each subproblem ω as the optimal value of α_ω instead of using a large negative value. Fig. 1 shows the advantage of using optimal α^*_ω. The master problem objective reaches its optimal value in the very initial iterations. We use machine learning to predict α^*_ω for each subproblem ω by reading power demand before implementing multi-cut Benders Algorithm I.

B. Training Dataset Preparation

Power demand scenarios are input to two-stage SCUC. The optimal values of proxy variables α^s_ω and thus the formation of useful Benders cuts are correlated with the power demand profile of each subsample ω. Thus, we use demand information as the input to a supervised learning model whose output is α^{s*}_ω. One can incorporate other problem-dependent features such as generation cost functions (which are assumed to be non-variable in this paper) along with the demand profile.

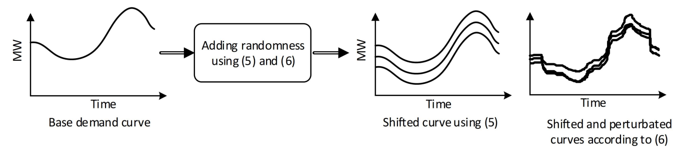

Fig. 2. Demand scenario generation.

To account for potential system operation scenarios in the training phase, we generate a set of daily load profile samples D^s according to (5). To follow the concept of demand uncertainty modeling in the two-stage SCUC problem, each sample D^s is used to generate ω subsample D^s_ω generated by (6).

$$D^s = D_{base}[\eta^L_s + \eta_s(\eta^U_s - \eta^L_s)] \quad \forall s \tag{5}$$

$$D^s_\omega = D^s[\eta^L_\omega + \eta_\omega(\eta^U_\omega - \eta^L_\omega)] \quad \forall \omega, \forall s \tag{6}$$

where $D^s \in \mathbb{R}^{1\times t}$ and $D^s_\omega \in \mathbb{R}^{\omega\times t}$. $\eta_s\{\cdot\}$ and $\eta_\omega\{\cdot\}$ follow a uniform distribution between 0 and 1. Indices for samples and subsamples are denoted by s and ω. Two stages of randomness are carried out to consider a range of realistic operational conditions. The load point is shifted using the random parameter $\eta_s\{\cdot\}$ to model its daily and seasonal volatility within the range $[\eta^U_s,\ \eta^L_s]$. η^U_s and η^L_s can be determined by historical data and load growth projection or to the point when any further increase or decrease renders unit commitment infeasible. The uncertainty in subsamples is modeled using the random parameter $\eta_\omega\{\cdot\}$ within a specified range $[\eta^U_\omega, \eta^L_\omega]$. Subsamples model the hourly load randomness by multiplying D^s with random parameters and creating D^s_ω. Fig. 2 illustrates the load scenario generation.

For a given system, Algorithm I is executed for each demand profile sample D^s. Optimal values of proxy variables upon Benders convergence are labeled as α^{s*} and stored. Expressions (7) and (8) are, respectively, the input and target of the supervised learning model.

$$Input: \mathbf{D} = [D^s_1, D^s_2, D^s_3, ..., D^s_\omega]^T \quad \forall s \tag{7}$$

$$Target: \boldsymbol{\alpha} = [\alpha^{s*}_1, \alpha^{s*}_2, \alpha^{s*}_3, ..., \alpha^{s*}_\omega]^T \quad \forall s \tag{8}$$

C. Supervised Learning Strategy

We use neural networks (NN), an efficient tool to capture the complexity and nonlinearity of a function by utilizing various activation functions. The regressor uses a fully connected NN with mini-batch gradient descent and Rectified Linear Units (ReLU) activation functions for hidden layers. The loss

TABLE I
HYPERPARAMETERS OF NN REGRESSOR

Hidden layer=1, Batch size=300 500, Activation = ReLU, Loss function = MSE, Optimizer = Adam

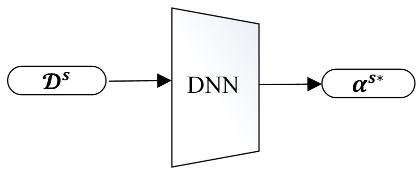

Fig. 3. NN regressor reads demand profiles and predicts proxy variables α.

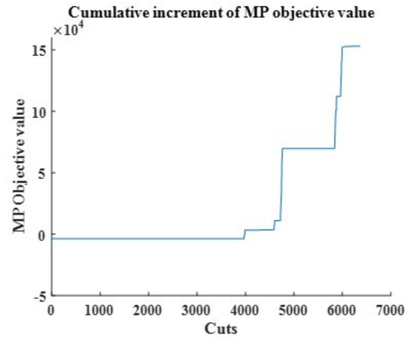

Fig. 4. Benders lower bound improvement with cuts added cumulatively; a case of IEEE 118-bus system.

function, mean squared error (MSE), provides a quantitative measure of prediction accuracy and how well the model's predictions align with the actual observed values.

$$MSE = \frac{\sum_n (\alpha_n - \hat{\alpha_n})^2}{n} \tag{9}$$

where α_n and $\hat{\alpha_n}$ are, respectively, the ground truth and predicted value of the subproblem proxy, and n is the number of samples. We use Adam optimizer to train the learner and determine optimal weights. Various batch sizes, epochs, and layer counts are tested to determine the best architecture. A single hidden layer is selected to have a minimalistic model. Table I shows the architecture and hyperparameters used in this study. Although more sophisticated structures may improve the results, we obtained promising outcomes with this minimalistic architecture. Fig. 3 illustrates a conceptual schematic of the learner model.

D. Data Scaling

Normalization plays a crucial role in mitigating biases arising from differing scales among features. By ensuring consistent feature scales, normalization stabilizes gradient descent steps, allowing for the use of higher learning rates, which in turn leads to faster convergence. The data samples are normalized using equation (10).

$$d_{normalized} = \frac{d - d_{min}}{d_{max} - d_{min}} \tag{10}$$

where d is the target value of demand in the training dataset and d_{min} and d_{max} are minimum and maximum values of the demand used to normalize the data.

E. Useful Cut Identification

The master problem still retains all cuts generated after carrying out each Benders iteration. As a result, the master problem size grows at every iteration, calling for a considerable computational memory requirement. In most cases, a subset of Benders cuts generated at each iteration contains the necessary information to build the feasible search space of the master problem. Currently, a lack of practical and systematic approach exists for classifying useful and non-useful cuts for large-scale problems [10]. Several features are suggested in [23]–[25], such as cut violation, cut depth, cut order, cut producing scenario, and Lagrange multiplier to identify approximately useful cuts.

Our idea of useful cut identification is based on Fig. 4. which shows the incremental progression of master problem objective value with cumulatively added cuts. It can be observed that not all cuts contribute equally to the improvement of the lower bound and optimality gap. Several cuts provide a positive increase in the lower bound. Such cuts can be classified as useful cuts. Other cuts do not contribute to the lower bound improvement and can be classified as non-useful. Our experimental observation shows that the numerical difference between cut values $\psi(u_{g,t}^k)$ and proxy values α_ω^k can be used for useful cut identification. If an inequality constraint is satisfied as equality upon solving optimization, it is typically referred to as a binding or strictly useful constraint. Thus, if α_ω^k - $\psi(u_{g,t}^k)$ = 0, the cut is useful.

However, some cuts that are not exactly satisfied as equality may contain the necessary information to form the master problem feasible region. Such cuts should also be considered useful. As an inequality constraint approaches equality, its importance in forming a feasible region is expected to increase. Consider a demand profile sample D^s. To capture all cuts that contain necessary information at each iteration k, we label cuts as "useful" if the difference between α_ω^k and $\psi(u_{g,t}^k)$ is less than a threshold δ, which is selected through experimental observation. We have tested various cases and observed that δ within the range $[0, 100]$ works well for the studied cases. Once all cuts are generated at iteration $k-1$, we add them to MP and solve the updated MP at iteration k to obtain $u_{g,t}^k$ and α_ω^k. We check (11) to identify useful cuts generated at iteration $k-1$. Cuts that satisfy (11) contain necessary information and are useful. We repeat this process at each iteration k to detect useful cuts generated at iteration $k-1$. Non-useful cuts are dropped from MP before moving to iteration $k+1$.

$$h_\beta^{k-1}(u_{g,t}, \alpha_\omega)^\phi : |\alpha_\omega^k - \psi(u_{g,t}^k)| \leq \delta \tag{11}$$

where $\psi(u_{g,t}^k)$ is the numerical value of the right-hand side of

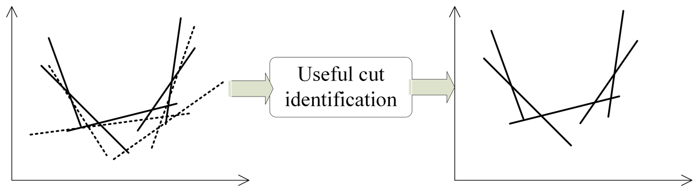

Fig. 5. Useful cut identification.

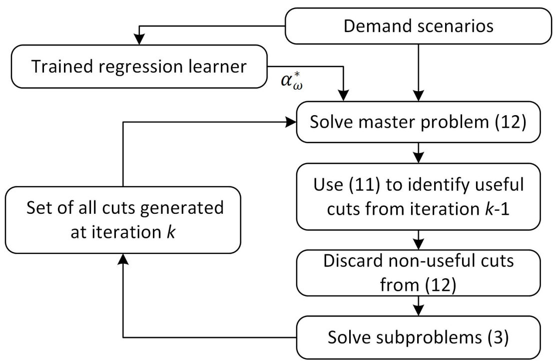

Fig. 6. R-Benders flowchart.

(2d).

$$\psi(u^k_{g,t}) = J^{k-1}_{SP,\omega} + \sum_t \sum_g \lambda^{k-1}_{g,t,\omega}(u_{g,t} - u^{k-1}_{g,t})$$

Fig. (5) illustrates the cut filtering concept, with dotted lines representing non-useful cuts and solid lines denoting useful cuts that must be retained in the master problem. Even if a cut is labeled useful in iteration k, it is possible that another cut, found to be useful in a subsequent iteration $k+n$, might supersede the earlier cut. Therefore, at every iteration k, the usefulness of all cuts accumulated from the first iteration up to the current one should be reassessed to ensure their continued relevance.

F. Avoiding Information Loss

Selecting a small δ results in labeling fewer cuts as useful. A large δ yields more cuts in the master problem, and thus less computational cost reduction is gained. Depending on the value of hyperparameter δ, we have observed cases with no cut labeled as useful at a few iterations. Retaining at least one cut from each iteration is crucial; otherwise, the proposed R-Benders may not converge due to loss of information from that particular iteration. To prevent this, we keep several cuts generated from high-load scenarios at each iteration, most of which may already be on the useful cut list. The number of retained cuts is determined through experimentation.

G. Reduced MP Formulation

The R-Benders master problem at iteration k is as follows:

$$\min_x \sum_t \sum_g (SU_g y_{g,t} + SD_g z_{g,t}) + \sum_\omega \alpha_\omega \tag{12a}$$

s.t.

$$(1b) - (1i) \tag{12b}$$

$$\cup_{i=1}^{i=k-2} h^i_\beta(u_{g,t}, \alpha_\omega)^\psi \tag{12c}$$

$$h^{k-1}_\beta(u_{g,t}, \alpha_\omega) : \alpha_\omega \geq J^{k-1}_{SP,\omega} + \sum_t \sum_g \lambda^{k-1}_{g,t,\omega}(u_{g,t} - u^{k-1}_{g,t}) \;\; \forall\omega \tag{12d}$$

$$\alpha_\omega \geq \alpha_\eta \alpha^*_\omega \;\; \forall\omega \tag{12e}$$

$$x \in \{u_{g,t}, y_{g,t}, z_{g,t}, \alpha_\omega\}$$

where (12c) denotes the list of all useful cuts detected until iteration $k-2$. (12d) includes all cuts generated at iteration $k-1$. The value of α_ω successively progresses to reach the optimal value. To maintain the quality of the lower bound solution, we should ensure $\alpha_\omega \geq \alpha^*_\omega$. Given the possibility of machine learning error, the regressor predictions α^*_ω are reduced by a factor α_η and $\alpha_\omega \geq \alpha^*_\omega$ is replaced with $\alpha_\omega \geq \alpha_\eta \alpha^*_\omega$. The reduction factor α_η is determined by analyzing the regressor prediction error.

The R-Benders approach is summarized in Algorithm II and Fig. 5. The only learner embedded in R-Benders is an NN regressor to predict α^*_ω. One can train a classification learner, as in the C-Bender approach presented in Section IV, to identify useful cuts. However, the R-Benders approach does not use any classifier, instead opting to analytically filter cuts with a one-iteration delay. The proposed analytical cut filtering approach is versatile and can identify and eliminate non-useful cuts from the master problem, irrespective of varying factors in the SCUC model such as demand, the number of scenarios, and generation cost, as well as the number of cuts generated at each iteration.

Algorithm 2 Proposed R-Benders decomposition

```
Initialize convergence tolerance ε, set iteration index k = 0.
Feed demand scenarios to the trained regressor to predict α*_ω.
while ε > tolerance do
    k ← k + 1
    if k = 1 then
        Solve master problem, (12a)-(12b), (12e).
    end if
    if k > 1 then
        Form new cut, α_ω ≥ J^{k-1}_{SP,ω} + Σ_t Σ_g λ^{k-1}_{g,t,ω}(u_{g,t} − u^{k-1}_{g,t}) ∀ω
        Solve master problem (12) to obtain u^k_{g,t} and α^k_ω.
        Identify the useful cuts from iteration k − 1 using (11).
        Store the useful cuts, (12c).
    end if
    for n = 1 : ω do
        Solve subproblem (3) to obtain J^k_{SP,ω} and λ^k_{g,t,ω}.
    end for
    Calculate tolerance value, ε = |(UB−LB)/LB|.
end while
```

H. Subproblem Acceleration

Two non-binding transmission line constraint removal strategies can be used to accelerate subproblem solutions, particularly for large SCUC problems. In the first learning-aided strategy, two classifiers can be trained to identify non-binding branch constraints and generation ramp constraints at each Benders iteration and remove them from the model [31]. Another more straightforward but less efficient strategy is to solve a relaxed subproblem without branch constraints. The relaxed subproblem solution and shift factors are used to check branch constraints. The violated constraints are added to the relaxed subproblem, which is repeated until all branch constraints are satisfied [32]. These two strategies reduce computation time and memory usage. We have used the second strategy.

IV. Learning Classification-based Benders

In this section, we discuss a recently developed method that leverages machine learning to classify Benders cuts after each iteration [25]. We then draw comparisons between this method and our proposed R-Benders approach. Key features that define a cut's utility, as identified in existing research, are discussed, along with the computational costs involved in cut labeling. Furthermore, we highlight how our approach incorporates regression-based proxy prediction into the C-Benders framework, improving the efficacy of the two-stage SCUC solution process.

A. C-Benders

We conducted performance evaluations of the cut classifier approach, during which we created a dataset specifically designed for training the classifier. Upon generating the cut coefficients from the subproblem, a trained supervised classifier is deployed to detect useful cuts prior to resolving the master problem at each iteration k. The classifier's input comprises features extracted from cuts generated at iteration k, while its output is binary cut labels (i.e., useful (1) or non-useful(0)). Only the useful cuts are retained, while the remainder are excluded from the set of cuts forming the master problem. A neural network is trained with F-score as the loss function, and the model architecture and hyperparameters are outlined in Table I.

In order to prepare a training dataset for the classifier, all cuts must undergo labeling as either useful or non-useful. This necessitates the extraction of cut features. According to existing literature [25], a Benders cut qualifies as useful if it yields a positive increment in the lower bound (i.e., cut violation as discussed earlier) of the master problem. To accomplish this, we execute Benders decomposition for each demand sample and compile all cuts from all iterations upon convergence. Subsequently, we re-solve the master problem to ascertain the lower bound improvement attributable to each cut. We augment the initial master problem with one of the stored cuts and measure its impact on the lower bound improvement.

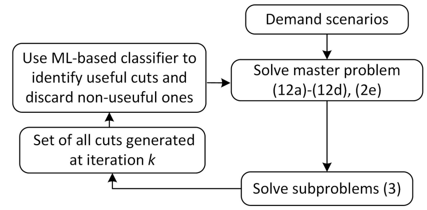

Fig. 7. C-Benders flowchart.

This process is iterated, with cuts being cumulatively added one after another, to label all cuts based on their contribution. An alternative approach might be initially forming the master problem with all cuts and subsequently removing cuts one by one to observe their effect on the lower bound.

Generating a labeled dataset for C-Benders is computationally expensive, particularly for multi-cut Benders decomposition while the proposed R-Benders approach does not require such dataset generation. The number of Benders iterations is relatively high for the SCUC problem with intertemporal constraints, leading to a multiplicatively larger number of cuts that should be individually checked. The number of cuts generated at each iteration depends on the number of demand scenarios. This is another factor that increases the computational cost of cut labeling. For instance, consider solving the 118-bus system with a 4-hour commitment horizon and a load profile sample with 40 stochastic scenarios. This case takes 289 iterations to converge with a less than 1% gap. The total number of cuts generated for each sample is 40×289 =11,560. To determine the label of each cut, we must solve the master problem 11,560 times. However, C-Benders provides comparatively more accurate cut labels than R-Benders, which estimates labels using (11). Fig. 7 shows the C-Benders flowchart. The master problem is formulated in (12a)-(12d), and (2e). Useful cuts in (12c) are identified using the classifier. The training process is offline. Although the overall accuracy of the C-Benders is slightly better than the proposed R-Benders approach, the computational overhead of C-Benders associated with cut labeling is higher compared to R-benders proxy prediction.

Another study [24] uses the same cut classification principle but adopts a slightly different approach to cut labeling. It leverages the dual value, specifically the Lagrange multiplier of constraints from consecutive iterations, to distinguish between useful and non-useful cuts. If the Lagrange multiplier associated with a Benders cut is non-zero, it is categorized as useful; otherwise, it is classified as non-useful and discarded. However, determining the Lagrange multiplier necessitates re-solving the master problem after each Benders iteration, rendering the cut labeling process computationally intensive. In contrast, the proposed R-Benders algorithm only requires solving algebraic operations instead of resolving the optimization problem to determine the Lagrange multiplier, resulting in significantly reduced computational overhead.

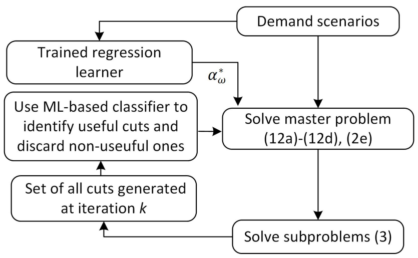

Fig. 8. CR-Benders flowchart.

TABLE II
PARAMETER RANGE FOR DEMAND DATASET GENERATION

System	Load	
	$[\eta^L_s, \eta^U_s]$	$[\eta^L_\omega, \eta^U_\omega]$
24-bus	70%-130%	95%-105%
118-bus	70%-130%	95%-105%
1354-bus	70%-110%	95%-105%

TABLE III
AVERAGE NUMBER OF BENDERS ITERATIONS, MP SOLVE TIME, AND USEFUL CUTS FOR DIFFERENT NUMBERS OF DEMAND SCENARIOS

# demand scenarios	10	20	30	40
# iter	53	51	55	49
# cuts	532	1029	1643	1951
MP time (sec)	62.5	63	75	39

TABLE IV
VARIATION IN NUMBER USEFUL CUTS AND ITERATIONS UNDER VARYING GENERATION COST FUNCTION

Cost variation range	70% $\sim$ 130%
# iterations	43 $\sim$ 73
# cuts	437 $\sim$ 1397

B. CR-Benders

We have incorporated certain features of the R-Benders method into the C-Benders approach, creating a hybrid technique we call CR-Benders. Illustrated in Fig. 8, CR-Benders uses a regressor to predict the initial values of subproblem proxy variables α^*_ω and uses a trained classifier to filter cuts before the master problem is solved in each iteration k. This combined strategy outperforms the traditional C-Benders approach in terms of efficiency.

V. NUMERICAL SIMULATIONS

We evaluate the effectiveness of the proposed R-Benders approach and demonstrate its performance comparison with C-Benders on multiple test systems. For the implementation of Benders decomposition, we use the YALMIP toolbox and CPLEX solver [33]. Neural network models are constructed using PyTorch. All simulations are carried out on a computer equipped with an Intel(R) Xeon(R) CPU at 2.10 GHz and 512 GB of RAM.

A. Test Systems and Data Preparation

Three test systems with various scheduling horizons are used, including the IEEE 24-bus, 118-bus, and 1354-bus systems [34]. The 24-bus system includes ten generators and 34 transmission lines. The 118-bus system consists of 54 generators and 186 lines. The 1354-bus system has 270 generators and 1991 branches. Each case includes 40 equiprobable load profile scenarios. Table II shows the range of load perturbation with respect to base case demand values to incorporate daily and hourly load uncertainty. The convergence tolerance is set to 1% for all cases.

B. Useful Cut Statistics

Table V shows the average number of total cuts and useful cuts for different systems where h stands for horizons. To determine the usefulness of a cut, we have calculated the contribution of each cut to improving the objective value of the master problem. A cut is useful if it results in a non-zero increase in the objective value. The percentage of the number of useful cuts to total cuts reduces as the size of the system and the number of scheduling horizons increase. For instance, for the IEEE 118-bus system with three time periods, only 13% of cuts are labeled as useful, and the rest of 87% are non-useful. Filtering non-useful cuts significantly reduces the computational burden of the master problem.

While we have considered the demand profile with 40 load profile scenarios, one can use a different number of scenarios and also another varying factor such as the generation cost function. To evaluate the impact of the number of scenarios on cut generation, we have selected ten stochastic samples, each comprising 40 scenarios over a 12-hour horizon of the IEEE 24-bus system. From these, we have extracted subsamples with 10, 20, and 30 scenarios evenly spaced between the highest and lowest total load. Subsequently, we have executed the proposed accelerated Benders decomposition for each subset and the complete set of 40 scenarios. Table III shows that the number of iterations, master problem solve time, and the number of useful cuts vary depending on the number of scenarios. As expected, increasing the number of scenarios increases the total number of cuts and, thus, the number of useful cuts determined by the proposed cut filtering approach.

One can also consider varying generation cost profiles when forming cuts. For the 24-bus system, we generated multiple distinct generation cost scenarios, where each generator's cost fluctuated randomly between 70% and 130% of the base value while keeping demand and network topology constant. Table IV shows that generation cost impacts on Benders cuts formation and thus the usefulness of a cut. Although these factors affect the number of Bender cuts and iterations, the proposed analytical cut filtering approach can identify and eliminate non-useful cuts from the master problem, irrespective of the number of cuts generated at each iteration.

TABLE V
COMPARISON OF TOTAL CUTS AND USEFUL CUTS

Test case	Average number of total cuts	Average number of useful cuts	% of useful cuts
24-bus,1h	224	103	46%
24-bus,12h	1872	220	12%
118-bus,1h	308	109	35%
118-bus,3h	2280	288	13%
1354-bus,2h	5520	795	15%

TABLE VI
AVERAGE SOLVER TIME (AND IMPROVEMENT PERCENTAGE) COMPARISON IN SECONDS

System	Benders	R-Benders	C-Benders	CR-Benders
24-bus,1h	2.1	1.4 (33%)	0.73 (65%)	0.66 (69%)
24-bus,12h	37.7	32.9 (13%)	9.7 (74%)	9.4 (75%)
118-bus,1h	2.8	2.6 (7%)	2.5 (11%)	2.5 (11%)
118-bus,3h	236	30 (87%)	57 (76%)	24 (90%)
1354-bus,2h	110	74 (33%)	65 (41%)	62 (44%)

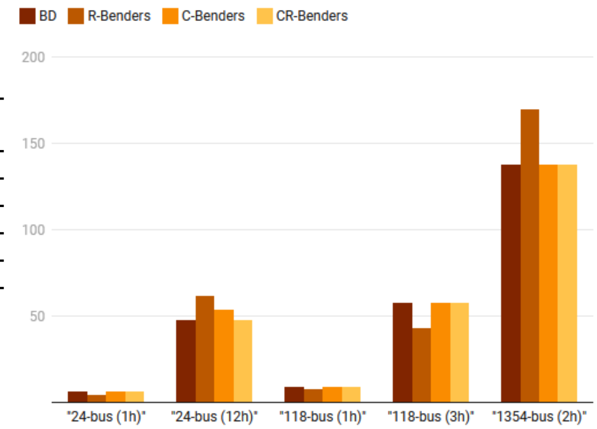

Fig. 9. Average number of iterations.

TABLE VII
COSTGAP INDEX (%)

System	R-Benders	C-Benders	CR-Benders
24-bus,1h	0	0.13	0.07
24-bus,12h	0.05	0.8	0.03
118-bus,1h	0.6	0.01	0.01
118-bus,3h	0	0.02	0.02
1354-bus,2h	0.04	0.05	0

C. Master Problem Runtime and Solution Quality

Table VI reports the average solver time of the master problem and the percentage of time improvement obtained by the proposed R-Benders approach and two other classifier-based approaches, i.e., C-Benders and CR-Benders, compared to the conventional multi-cut Benders. During each iteration of R-Benders, C-Benders, and CR-Benders, we tackle a significantly smaller problem, resulting in considerable time savings. The time saving becomes more significant as the size of the optimization model increases. For instance, R-Benders reduces the master problem solution time from 236 seconds to 30 for the 118-bus system, an 87% improvement. The time-saving of CR-Benders is better than R-Benders and C-Benders.

The average number of Benders iterations is shown in Fig. 9. The three proposed approaches, particularly C-Benders and CR-Benders, take roughly the same number of iterations as conventional Benders decomposition. A comparison of the solution time and the number of iterations shows that R-Benders, C-Benders, and CR-Benders can capture the necessary information to build a computationally less expensive master problem.

We use a CostGap index to assess the quality of the solution. CostGap measures the difference between the objective values generated by the proposed approach f^p and the conventional Benders f^{BD}.

$$CostGap\% = \frac{|f^P - f^{BD}|}{f^{BD}} \times 100 \tag{13}$$

Table VII presents the average CostGap index for all cases. The average cost gap for all test cases is negligible. For instance, the gap is less than 0.02% for the 118-bus system with a 3-hour horizon. We have observed several cases for which the proposed approaches obtain even a better solution than the conventional Benders. This is due to a warm start with stronger cuts in the first iteration.

D. Memory Usage

The memory usage of the proposed R-Benders approach is compared with that of conventional Benders as well as the classifier-based C-Benders and CR-Benders. The memory occupied to form the constraint set from Benders cuts is illustrated in Fig. 10. The memory requirement is reduced significantly by filtering out non-useful cuts from the model. For example, Fig. 10.d shows that for the 118-bus system with a 3h horizon, the memory usage of CR-Benders is 87.5% less than that of the conventional Benders decomposition. Fig. 11 demonstrates the number of cuts added to the master problem

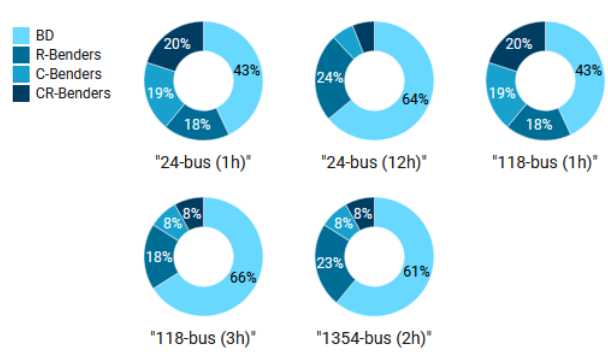

Fig. 10. Memory usage comparison. A) 24-bus (1h), b) 24-bus (12h), c) 118-bus (1h), d) 118-bus (3h), and e) 1354-bus (2h).

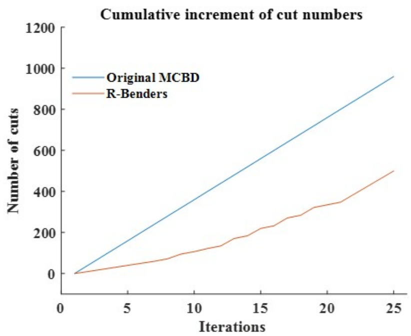

Fig. 11. Cumulative increment of the number of cuts for IEEE 118-bus 3h case.

per iteration. The slope of the increment for the conventional Benders is steeper than that of RC-Benders.

E. Non-convergent case analysis

In specific situations, the conventional Benders decomposition technique struggles to achieve convergence within the allocated timeframe, even in the presence of an optimal solution. Through our experiments, we have demonstrated that the utilization of a predicted subproblem proxy can effectively address these scenarios and enhance the duality gap. We focused on two particular scenarios involving a 118-bus system and a 3-hour horizon, where the original multi-cut Benders approach failed to converge within 400 iterations. By employing the R-benders method, we observed an average convergence within 59 iterations for these same scenarios. It is evident that the incorporation of the proxy value yields superior performance.

VI. Conclusion

The proposed algorithm aims to reduce the computational costs of two-stage SCUC by taking advantage of machine learning. Given the iterative nature of the learning-aided Benders algorithm, potentially discarded useful cuts will be regenerated in subsequent iterations, and the proposed algorithm will provide high-quality solutions. Conventional Benders decomposition suffers from slow convergence and might not reach an optimal point within a specified time. Not all cuts generated through Benders iterations contain useful information to form the feasible region of the master problem. Removing non-useful cuts and forming stronger cuts enhance Benders decomposition performance. The proposed approach uses a regressor to initialize the subproblem objective proxy variables, along with cuts being filtered based on the comparison between the numerical values of cuts and proxy variables obtained in a specific iteration. Furthermore, we introduced a scheme where the regressor for subproblem objective proxy value, along with a cut classifier, can be seamlessly integrated within the Benders decomposition algorithm. The proposed approaches are applicable to many MIP problems.

Simulation results on various test systems show CR-Benders outperform R-Benders and C-Benders in terms of solution time and memory saving. On average, CR-Benders leads to 58% time saving and 77% memory saving, which is higher than the other two approaches. Although C-Benders and CR-Benders might save more time than R-Benders, they need computationally expensive cut labeling for classifier training, which in turn demands excessive offline overhead compared to R-Benders.

References

[1] A. J. Conejo, L. Baringo, A. J. Conejo, and L. Baringo, "Unit commitment and economic dispatch," *Power System Operations*, pp. 197–232, 2018.

[2] T. Yang, D. Wu, Y. Sun, and J. Lian, "Minimum-time consensus-based approach for power system applications," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1318–1328, 2016.

[3] Y. Du, Y. Li, C. Duan, H. B. Gooi, and L. Jiang, "Adjustable uncertainty set constrained unit commitment with operation risk reduced through demand response," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1154–1165, 2021.

[4] W. Zhang, C. Shao, B. Hu, J. Zhou, M. Cao, K. Xie, P. Siano, and W. Li, "Proactive security-constrained unit commitment against typhoon disasters: An approximate dynamic programming approach," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 5, pp. 7076–7087, 2023.

[5] C. C. Carøe and J. Tind, "L-shaped decomposition of two-stage stochastic programs with integer recourse," *Mathematical Programming*, vol. 83, pp. 451–464, 1998.

[6] A. J. Conejo, E. Castillo, R. Mínguez, and R. García-Bertrand, "Decomposition in linear programming: complicating variables," *Decomposition Techniques in Mathematical Programming: Engineering and Science Applications*, pp. 107–139, 2006.

[7] 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.

[8] M. Minoux, *Mathematical programming: theory and algorithms*. John Wiley & Sons, 1986.

[9] J. BnnoBRs, "Partitioning procedures for solving mixed-variables programming problems '," *Numerische mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

[10] 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.

[11] K. Holmberg, "On the convergence of cross decomposition," *Mathematical Programming*, vol. 47, pp. 269–296, 1990.

[12] A. I. Corréa, A. Langevin, and L.-M. Rousseau, "Scheduling and routing of automated guided vehicles: A hybrid approach," *Computers & operations research*, vol. 34, no. 6, pp. 1688–1707, 2007.

[13] J.-F. Cordeau, G. Stojković, F. Soumis, and J. Desrosiers, "Benders decomposition for simultaneous aircraft routing and crew scheduling," *Transportation science*, vol. 35, no. 4, pp. 375–388, 2001.

[14] M. I. Restrepo Ruiz, *Grammar-Based Decomposition Methods for Multi-Activity Tour Scheduling*. PhD thesis, École Polytechnique de Montréal, 2015.

[15] J. Naoum-Sawaya and S. Elhedhli, "A nested benders decomposition approach for telecommunication network planning," *Naval Research Logistics (NRL)*, vol. 57, no. 6, pp. 519–539, 2010.

[16] K. Brandes, *Implementierung und analyse verschiedener strategien zur aggregation und disaggregation von multi-cuts im benders dekompositionsverfahren*. PhD thesis, Master's thesis, Universität Paderborn, North Rhine-Westphalia, Germany, 2011.

[17] A. Paulus, M. Rolínek, V. Musil, B. Amos, and G. Martius, "Comboptnet: Fit the right np-hard problem by learning integer programming constraints," in *International Conference on Machine Learning*, pp. 8443–8453, PMLR, 2021.

[18] M. Mohammadi and A. Mohammadi, "Empowering distributed solutions in renewable energy systems and grid optimization," *arXiv preprint arXiv:2310.15468*, 2023.

[19] P. Van Hentenryck, "Machine learning for optimal power flows," *Tutorials in Operations Research: Emerging Optimization Methods and Modeling Techniques with Applications*, pp. 62–82, 2021.

[20] G. Ruan, H. Zhong, G. Zhang, Y. He, X. Wang, and T. Pu, "Review of learning-assisted power system optimization," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 221–231, 2020.

[21] A. Perera, P. Wickramasinghe, V. M. Nik, and J.-L. Scartezzini, "Machine learning methods to assist energy system optimization," *Applied energy*, vol. 243, pp. 191–205, 2019.

[22] F. Hasan, A. Kargarian, and A. Mohammadi, "A survey on applications of machine learning for optimal power flow," in *2020 IEEE Texas Power and Energy Conference (TPEC)*, pp. 1–6, IEEE, 2020.

[23] 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.

[24] B. Vandenbussche, S. Delikaraoglou, I. Blanco, and G. Hug, "Data-driven adaptive benders decomposition for the stochastic unit commitment problem," *arXiv preprint arXiv:1912.01039*, 2019.

[25] M. Lee, N. Ma, G. Yu, and H. Dai, "Accelerating generalized benders decomposition for wireless resource allocation," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1233–1247, 2020.

[26] A. Ruszczyński, "Decomposition methods," *Handbooks in operations research and management science*, vol. 10, pp. 141–211, 2003.

[27] A. J. Conejo and L. Baringo, *Power system operations*, vol. 11. Springer, 2018.

[28] J. R. Birge and F. V. Louveaux, "A multicut algorithm for two-stage stochastic linear programs," *European Journal of Operational Research*, vol. 34, no. 3, pp. 384–392, 1988.

[29] L. Su, L. Tang, and I. E. Grossmann, "Computational strategies for improved minlp algorithms," *Computers & Chemical Engineering*, vol. 75, pp. 40–48, 2015.

[30] A. Nasri, S. J. Kazempour, A. J. Conejo, and M. Ghandhari, "Network-constrained ac unit commitment under uncertainty: A benders' decomposition approach," *IEEE transactions on power systems*, vol. 31, no. 1, pp. 412–422, 2015.

[31] F. Hasan and A. Kargarian, "Topology-aware learning assisted branch and ramp constraints screening for dynamic economic dispatch," *IEEE Transactions on Power Systems*, vol. 37, no. 5, pp. 3495–3505, 2022.

[32] D. A. Tejada-Arango, P. Sánchez-Martın, and A. Ramos, "Security constrained unit commitment using line outage distribution factors," *IEEE Transactions on power systems*, vol. 33, no. 1, pp. 329–337, 2017.

[33] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, pp. 284–289, IEEE, 2004.

[34] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, *et al.*, "The power grid library for benchmarking ac optimal power flow algorithms," *arXiv preprint arXiv:1908.02788*, 2019.

Fouad Hasan (Student Member, IEEE) received his B.Sc. degree in electrical engineering from the Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, in 2015 and his Ph.D. degree in electrical and computer engineering from Louisiana State University, Baton Rouge, USA, in 2023. His research interests include power systems optimization, electricity market, and machine learning.

Amin Kargarian (Senior Member, IEEE) is currently an Associate Professor with the Electrical and Computer Engineering Department at Louisiana State University, Baton Rouge, LA, USA. His research interests include optimization, machine learning, quantum computing, and their applications to power systems.