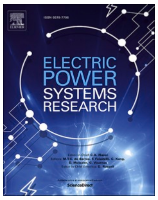
Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr



Machine Learning-Enhanced Benders Decomposition Approach for the Multi-Stage Stochastic Transmission Expansion Planning Problem

Stefan Borozan [a,*], Spyros Giannelos [a], Paola Falugi [b], Alexandre Moreira [c], Goran Strbac [a]

[a] Imperial College London, London, United Kingdom
[b] University of East London, London, United Kingdom
[c] Lawrence Berkeley National Laboratory, Berkeley, CA, USA

ARTICLE INFO

Keywords:
Benders decomposition
classification
expansion planning
machine learning
stochastic optimization
uncertainty

ABSTRACT

The necessary decarbonization efforts in energy sectors entail integrating flexible assets and increased levels of uncertainty for the planning and operation of power systems. To cope with this in a cost-effective manner, transmission expansion planning (TEP) models need to incorporate progressively more details to represent potential long-term system developments and the operation of power grids with intermittent renewable generation. However, the increased modeling complexities of TEP exercises can easily lead to computationally intractable optimization problems. Currently, most techniques that address computational intractability alter the original problem, thus neglecting critical modeling aspects or affecting the structure of the optimal solution. In this paper, we propose an alternative approach to significantly alleviate the computational burden of large-scale TEP problems. Our approach integrates machine learning (ML) with the well-established Benders decomposition to manage the problem size while preserving solution quality. The proposed ML-enhanced Multicut Benders Decomposition algorithm improves computational efficiency by identifying effective and ineffective optimality cuts via supervised learning techniques. We illustrate the benefits of the proposed methodology by solving multi-stage TEP problems of different sizes based on the IEEE24 and IEEE118 test systems, while also considering energy storage investment options..

1. Introduction

Power system planning is a large-scale problem that is crucial in facilitating the cost-effective transition to a net-zero power sector. However, it is also subject to fundamental changes brought on by decarbonization efforts, including growing levels of uncertainty and a greater number of associated technologies, necessitating novel planning frameworks and their urgent application.

A principal challenge is the multi-source and multi-dimensional uncertainty around future system developments. Neglecting uncertainty altogether, or even considering future scenarios deterministically as done in incumbent practices [1], could lead to flawed network expansion plans associated with increased societal costs [2]. Therefore, it has been established in the scientific literature that the application of a form of stochastic optimization method is essential in the presence of uncertainty [3,4], while [2] and [5] argue that multi-stage formulations are necessary to unlock the potential for strategic decision-making for cost-effective planning [6]. Further challenges arise with the importance of integrating new technologies and recent evidence demonstrating that smart investment options, such as energy storage and demand response, could enhance investment flexibility [6] within a multi-stage strategic-decision-making framework.

In this context, [2] and [5] propose alternative modeling approaches to the multi-stage stochastic transmission expansion planning (TEP) problem with energy storage as a non-network alternative, while the same is done for demand response, soft-open points, and coordinated voltage control in [7], and vehicle-to-grid charger placements are co-optimized with network reinforcements in [8]. The inclusion of detailed uncertainty representation, broad investment portfolios, and intertemporal operation of flexibility assets evolves this class of planning problems leading to more informed decisions. Nevertheless, the increase in modeling complexities results in a greater, and often prohibitively large, computational burden.

In general, stochastic TEP models are formulated as mixed-integer linear programming (MILP) problems, that are NP-hard and can be extremely difficult to solve, especially when associated with a large

[*] Corresponding author.
E-mail addresses: s.borozan@imperial.ac.uk (S. Borozan), p.falugi@uel.co.uk (P. Falugi), AMoreira@lbl.gov (A. Moreira), g.strbac@imperial.ac.uk (G. Strbac).

https://doi.org/10.1016/j.epsr.2024.110985
Received 5 April 2024; Received in revised form 23 July 2024; Accepted 12 August 2024
Available online 25 August 2024

number of candidate solutions over many operating points. Decomposition and parallelization techniques are often applied to handle computational difficulties [9]. Benders decomposition [10,11] has been highlighted as especially suited to address this class of problems because its structure allows for straightforward disaggregation of investment and operation constraints by taking the binary investment decisions as complicating variables [8]. Researchers have also attempted to enhance aspects of the performance of Benders decomposition [12]. For example, multicut formulations [13], which append one cut per subproblem (SP) in every iteration, are often used to improve convergence, while parallelization of the SPs can speed up solution times [12]. However, the critical issue in multicut Benders decomposition (MBD) is that all binary decision variables are aggregated in the master problem (MP), the size of which grows rapidly over iterations, becoming increasingly difficult to solve. It has been reported that more than 90% of the total solution time is dedicated to MPs [12] and could lead to intractability after a number of iterations. Nevertheless, only 3.13% of reviewed articles in [12] apply a method aimed at controlling the size of the MP, highlighting a research gap.

The interest in Machine Learning (ML) methods has been growing recently in power systems research, as evidenced in [14]. The previous reference reveals that ML has predominantly found applications in time-critical domains, which can lead to significant improvement in execution times. Nonetheless, the strengths of ML can be exploited to enhance various aspects of optimization-based frameworks via their integration, as argued in [15] and demonstrated with application to power systems in [16]. In line with the classification approach employed in [15], authors in [17] identify four categories of learning-assisted power system optimization algorithms: boundary parameter improvement, optimization option selection, surrogate model, and hybrid model. The review cites only two references with relevance to planning. One example is [18], in which the power flow calculation is replaced with a learning-based method in a hybrid model formulation. Other recent publications in this emerging field have focused on electric vehicle charging infrastructure planning in a predict-then-optimize formulation, such as [19], where charging demand is first predicted using a graph convolutional network to inform the optimal charger allocation problem. The relevance of these works notwithstanding, they do not directly address the computational bottleneck of combinatorial optimization problems, which could have a broader contribution to power system modeling.

Other promising techniques outlined in [15] are yet to be widely adopted in power system applications. For example, an ML model is proposed in [20] to identify sets of active constraints to obtain optimal solutions more efficiently. A similar approach based on learning active constraints is proposed in [21] for linear bilevel problems with application to generator strategic bidding, and [22] proposes to learn redundant constraints from previous instances of the unit commitment problem. However, such methods change the problem structure and are heuristic without guarantees of optimality or feasibility [21]. Leveraging the fact that not all supporting hyperplanes in iterative algorithms contribute equally to constraining the MP, novel approaches have emerged that incorporate classification techniques to identify useful cuts. Notably, [23] proposes a learning-based selection of cutting planes in integer programming methods, while [24] and [25] translate the idea to Benders decomposition of two-stage stochastic programs and of mixed-integer nonlinear programs for wireless communications, respectively, utilizing support vector machines to distinguish between effective and ineffective cuts. Nonetheless, to the authors' knowledge, this novel theory has not been extended to a multi-stage setting or applied to power systems problems to date.

Within this context, we propose a learning-assisted decomposition methodology to address computational intractability issues of large-scale stochastic TEP problems. More specifically, we build upon previous research on cuts classification to develop a Benders decomposition approach suitable for MILP problems with complete recourse that is able to manage the critical computational bottleneck – the MP size growth over iterations [3], thereby enabling the increase in modeling complexity that is necessary to tackle modern power system challenges. The proposed method is an ML-enhanced node-variable multicut Benders decomposition algorithm (ML-MBD) that can solve the multi-stage stochastic TEP problem, considering multiple investment alternatives with diverse techno-economic characteristics. We present an investigation into feature and target selection for this class of problems, as well as an appropriate sampling and training methodology. The proposed ML-MBD is then benchmarked against the classical MBD and evaluated against two alternative learning-assisted versions of the method. Finally, note that this work presents a novel implementation of ML in power systems that tackles common tractability issues in large stochastic problems without resorting to reducing problem size through approaches that could jeopardize solution quality. The contributions of this paper can be summarized as:

- Develops a novel computationally efficient ML-enhanced Benders decomposition method to solve the multi-stage stochastic TEP problem.
- Proposes a hybrid framework that exploits advantages of both ML and optimization for solving multi-stage stochastic optimization problems with complete recourse while preserving solution quality.
- Investigates and proposes suitable classification targets, features, and hyperparameters for this class of problems.
- Demonstrates the computational benefits and potential applications of the proposed method.

The following Section presents the mathematical formulations that form the basis for the developed ML-enhanced decomposition approach. Then, Section 3 presents the proposed method, including details on target selection, labeling, feature selection, and associated algorithms. The case studies that validate ML-MBD and demonstrate its benefits are presented in Section 4, while Section 5 provides a critical discussion on the developed method and its contributions. The conclusions are summarized in Section 6.

2. Mathematical formulations

2.1. Multi-stage stochastic transmission expansion planning

Long-term uncertainty in stochastic optimization problems with a discrete distribution can intuitively be represented using a scenario tree in which each node in Ω_m is a realization of the uncertain parameters and associated with a state probability (π_m). On the other hand, short-term operational variability is considered in TEP through representative time blocks in Ω_B that are typically not presented as part of the scenario tree but are directly associated with each m. The TEP under uncertainty problem can be compactly formulated as a node-variable multi-stage stochastic program with (1) – (6).

$$\psi = \min_{\mathrm{x}_m, \mathrm{y}_{m,b}} \sum_{m \in \Omega_m} \pi_m \left(f^I_m(\mathrm{x}_m) + \sum_{b \ \in \ \Omega_B} f^O_{m,b}\left(\mathrm{y}_{m,b}\right) \right) \tag{1}$$

subject to

$$h^I_m(\mathrm{x}_m) = 0 \tag{2}$$

$$g^I_m(\mathrm{x}_m) \leq 0 \tag{3}$$

$$h^O_m\left(\mathrm{x}_{a(m)}, \mathrm{y}_{m,b}\right) = 0 \tag{4}$$

$$g^O_m\left(\mathrm{x}_{a(m)}, \mathrm{y}_{m,b}\right) \leq 0 \tag{5}$$

$$\mathrm{x}_m \in \mathscr{X}_m, \ \mathrm{y}_{m,b} \geq 0 \tag{6}$$

The superscripts I and O relate to investment and system operation, respectively. The objective function minimizes the total expected cost across all considered realizations of uncertainty, comprising investment and system operation costs. The problem is MILP because investment decisions x_m are mixed integer and chosen from a portfolio of options, including network reinforcements and non-network alternatives, while decision variables $\mathrm{y}_{m,b}$ are continuous and related to system operation. In (4) and (5) it is observed that system operation constraints are subject to investment decisions in m and all its ancestor nodes, $a(m)$, considering any construction lead times.

A problem is said to have complete recourse when for any feasible x, there exist y such that the problem is feasible in all stages. The stochastic TEP is a complete recourse problem because for all feasible x_m, system operation is feasible in all m due to the presence of load curtailment as a slack decision variable ($y^{\sigma}_{m,b}$) in $f^{O}_{m,b}(\cdot)$, which is penalized with a large cost – the Value of Lost Load.

2.2. Multicut Benders Decomposition

Taking the investment decisions as complicating variables, (1) – (6) can be decomposed into an investment master problem ($\mathbb{P}^M$) and $|\Omega_s|$ system operation subproblems ($\mathbb{P}^S_s$), each corresponding to a unique (m, b) pair. Letting $c_m,\ d_m,\ A_m,\ b_m, F_{m,b}$ and $h_{m,b}$ be functions of uncertainty, (1) is rewritten as:

$$\psi = \min_{\mathrm{x}_m,\mathrm{y}_{m,b}} \sum_{m\in\Omega_M} \pi_m \left(c_m^T \mathrm{x}_m + \sum_{b\ in\ \Omega_B} d_m^T \mathrm{y}_{m,b} \right) \tag{7}$$

Introducing the continuous variable $\alpha_{m,b} \in \mathbb{R}^+,\ \forall m, b$ to approximate operational costs, we formulate the $\mathbb{P}^M$ in iteration k with (8) – (11).

$$\psi^{M(k)} = \min_{\mathrm{x}_m,\alpha_{m,b}} \sum_{m\in\Omega_M} \pi_m \left(c_m^T \mathrm{x}_m + \sum_{b\ in\ \Omega_B} \alpha_{m,b} \right) \tag{8}$$

subject to

$$A_m \mathrm{x}_m \le b_m, \quad \forall m \in \ \Omega_M \tag{9}$$

$$\alpha_{m,b} \ge \left(h_{m,b} - F_{m,b}\mathrm{x}_m \right)^T \upsilon_{m,b}, \quad \forall m \in \Omega_{\mathrm{M}},\ b \in \Omega_B,\ \upsilon_{m,b} \in \Omega^{(k-1)}_{\mathscr{V}(m,b)} \tag{10}$$

$$\mathrm{x}_m \in \mathscr{X}_m, \quad \forall m \in \Omega_m \tag{11}$$

The MP yields a candidate solution $(\overline{\mathrm{x}}_m, \overline{\alpha}_{m,b})$ in every iteration. The set $\Omega^{(k-1)}_{\mathscr{V}(m,b)}$ in (10) contains the identified extreme points of the feasible region of the $\mathbb{P}^S_s$ corresponding to (m, b) at the current solution and is associated with the optimality cuts. Benders' method additionally involves feasibility cuts associated with the identified extreme rays of $\mathbb{P}^S_s$, which are omitted here since all SPs are feasible for any MP solution. By fixing investment decisions to $\overline{\mathrm{x}}_m$, $\mathbb{P}^S_s$ are free from integer decision variables and defined for each (m, b) pair as:

$$\psi^{S(k)}_{m,b}(\overline{\mathrm{x}}_m) = \min_{\mathrm{y}_{m,b}} d_m^T \mathrm{y}_{m,b} \tag{12}$$

subject to

$$W_{m,b}\mathrm{y}_{m,b} \le h_{m,b} - F_{m,b}\overline{\mathrm{x}}_m \ \ : \ \overline{\lambda}_{m,b} \tag{13}$$

$$\mathrm{y}_{m,b} \ge 0 \tag{14}$$

The MP (8) – (11) is a relaxation of the original problem, the objective function of which is reconstructed by iteratively generating optimality cuts of the form (10) from the solution of (12) – (14) and appending them in (8) – (11) until convergence is achieved. By weak duality, $\overline{\psi}^M$ provides a lower bound (LB) to the original problem, while a valid upper bound (UB) is given by $\sum_{m\in\Omega_M} \pi_m \left(c_m^T \overline{\mathrm{x}}_m + \sum_{b\ in\ \Omega_B} \overline{\psi}^S_{m,b}(\overline{\mathrm{x}}_m) \right)$, where $\overline{\psi}^M$ and $\overline{\psi}^S_{m,b}$ are the optimal values of the MP and SPs at the current candidate solution, respectively.

Each appended cut increases computational burden but does not necessarily contribute notably towards convergence. The aim of the proposed ML-MBD method is to discard those cuts that provide no or little contribution.

2.3. Transmission expansion planning under uncertainty with energy storage investment options

We present the deterministic equivalent of the node-variable multistage stochastic TEP under uncertainty problem with energy storage units as investment options in its decomposed form, following the MBD approach. The investment MP in iteration k is described by (15) – (25), where $\mathbf{x}$ is a vector containing all its decision variables. The investment options considered are line reinforcements of all lines l and energy storage units at candidate buses h.

$$\psi^{M(k)} = \min_{\mathbf{x}} \sum_{m\in\Omega_{\mathrm{M}}} \pi_m \left[r^I_{\epsilon_m} \Psi^I_m(\mathrm{x}) + \sum_{b\in\Omega_{\mathrm{B}}} \alpha_{m,b} \right] \tag{15}$$

where

$$\Psi^I_m = \sum_{l\in\Omega_L} \left(\overline{\kappa}_l X_{m,l} + \ \widetilde{\kappa}_l F_{m,l} \right) \Lambda_l + \ \sum_{h\in\Omega_H} \overline{\kappa}_{m,h} Y_{m,h} \tag{16}$$

subject to

$$X_{m,l} \in \{0, 1\}, \quad \forall m \in \Omega_M, \forall l \in \Omega_L \tag{17}$$

$$F_{m,l} \ge 0, \quad \forall m \in \Omega_M, \forall l \in \Omega_L \tag{18}$$

$$Y_{m,h} \ge 0, \quad \forall m \in \Omega_M, \forall h \in \Omega_H \tag{19}$$

$$\sum_{m\in\Omega_{M(w)}} X_{m,l} \ \le 1, \quad \forall w \in \Omega_W, \forall l \in \Omega_L \tag{20}$$

$$X_{m,l}\overset{\vee}{F}_l \le F_{m,l} \le X_{m,l}\hat{F}_l, \quad \forall m \in \Omega_M, \forall l \in \Omega_L \tag{21}$$

$$\overset{\vee}{S}_h \le Y_{m,h} \le \hat{S}_h, \quad \forall m \in \Omega_M, \forall h \in \Omega_H \tag{22}$$

$$F^C_{m,l} = \sum_{a\in\Omega^{E_{\mathrm{L}}}_{A(m)}} F_{a,l}, \quad \forall m \in \Omega_M, \forall l \in \Omega_L \tag{23}$$

$$Y^C_{m,h} = \sum_{a\in\Omega^{E_S}_{A(m)}} Y_{a,h}, \quad \forall m \in \Omega_M, \forall h \in \Omega_H \tag{24}$$

$$\alpha_{m,b} \ge \overline{\psi}^{S,(\nu)}_{m,b} + \sum_{l\in\Omega_L} \left(F^C_{m,l} - \ \overline{F}^{C,(\nu)}_{m,l} \right) \overline{\lambda}^{F,(\nu)}_{m,b,l} + \sum_{h\in\Omega_H} \left(Y^C_{m,h} - \ \overline{Y}^{C,(\nu)}_{m,h} \right) \overline{\lambda}^{Y,(\nu)}_{m,b,h},$$

$$\forall m \in \Omega_M, \forall b \in \Omega_B, \nu = 1, \ldots k - 1 \tag{25}$$

The objective function (15) minimizes the expected total system cost, consisting of discounted investment and estimated annual operation costs, where system operation costs in each scenario tree node m and representative block b are approximated with variable $\alpha_{m,b}$. Investment costs are calculated according to (16). The mixed-integer nature of investment decision variables is enforced with (17) – (19). Constraints (20) limit network investments in a single line to one upgrade for the duration of the planning horizon, while (21) and (22) impose limits on the capacity expansion amounts for line reinforcements and energy storage, respectively. Then, (23) and (24) determine the cumulative capacity investments that are operational in node m, taking into account the commissioning delays for transmission lines E_L and energy storage E_S. Lastly, (25) represent all Benders cuts at the current iteration k.

The SPs in iteration k are defined for all $m \in \Omega_M$ and $b \in \Omega_B$ with (26) – (40), where $\mathbf{y}$ is a vector containing all decision variables. They represent independent problems that can be solved in parallel.

$$\psi^{S,(k)}_{m,b} = \min_{\mathbf{y}} \omega_b r^{O}_{\epsilon_m} \Psi^{O}_{m,b} \tag{26}$$

where

$$\Psi^{O}_{m,b} = \sum_{t\in\Omega^b_T} \tau \left[\sum_{g\in\Omega_G} \gamma^G_{m,g} p_{t,g} + \sum_{n\in\Omega_N} \left(\gamma^S_m \rho^-_{t,n} + \Gamma y^{\sigma}_{t,n} \right) \right] \tag{27}$$

subject to

$$\widetilde{F}^C_l = \overline{F}^{C,(k)}_{m,l} \ : \ \overline{\lambda}^{F,(k)}_{m,b,l}, \quad \forall l \in \Omega_L \tag{28}$$

$$\widetilde{Y}^C_h = \ \overline{Y}^{C,(k)}_{m,h} \ : \ \overline{\lambda}^{Y,(k)}_{m,b,h}, \quad \forall h \in \Omega_H \tag{29}$$

$$0 \leq p_{t,g} \leq \widehat{P}_{m,t,g}, \quad \forall t \in \Omega^b_T, \forall g \in \Omega_G \tag{30}$$

$$p_{t-1,g} - \tau\ R^d_g \leq p_{t,g} \leq p_{t-1,g} + \tau\ R^u_g, \quad \forall t \in \Omega^b_T \backslash \{t^1_b\}, \forall g \in \Omega_G \tag{31}$$

$$E_{t^e_b,n} = E_{t^0_b,n}, \quad \forall n \in \Omega_N \tag{32}$$

$$E_{m,t^1_b,n} = E_{m,t^0_b,n} + \tau \left(\eta^+_n \rho^+_{t^1_b,n} - \frac{\rho^-_{m,t^1_n,n}}{\eta^-_n} \right), \quad \forall n \in \Omega_N \tag{33}$$

$$E_{m,t,n} = E_{m,t-1,n} + \tau \left(\eta^+_n \rho^+_{t,n} - \frac{\rho^-_{t,n}}{\eta^-_n} \right), \quad \forall t \in \Omega^b_T \backslash \{t^1_b\}, \forall n \in \Omega_N \tag{34}$$

$$0 \leq E_{m,t,n} \leq \tau^S_h \left(S^0_n + \widetilde{Y}^C_h \right), \quad \forall t \in \Omega^b_T, \forall n \in \Omega_N \tag{35}$$

$$\rho^+_{t,n} \leq \left(S^0_n + \widetilde{Y}^C_h \right), \quad \forall t \in \Omega^b_T, \forall n \in \Omega_N \tag{36}$$

$$\rho^-_{t,n} \leq \left(S^0_n + \widetilde{Y}^C_h \right), \quad \forall t \in \Omega^b_T, \forall n \in \Omega_N \tag{37}$$

$$f_{t,l} = b_l \left(\delta_{t,i(l)} - \delta_{t,j(l)} \right), \quad \forall t \in \Omega^b_T, \forall l \in \Omega_L \tag{38}$$

$$\left| f_{t,l} \right| \leq F^0_l + \widetilde{F}^C_l, \quad \forall t \in \Omega^b_T, \forall l \in \Omega_L \tag{39}$$

$$\sum_{\forall g} G_{n,g} p_{t,g} + \sum_{\forall l} I_{n,l} f_{t,l} - \sum_{\forall n} \left(p^+_{t,n} - p^-_{t,n} \right) - D_{m,t,n} + y^{\sigma}_{t,n} = 0, \ \ \forall t \in \Omega^b_T, \forall n \in \Omega_N \tag{40}$$

The SP objective function (26) minimizes the discounted costs of generation, storage operation, and load curtailment, as defined in (27). Constraints (28) and (29) couple the operational subproblem with the current candidate investment decision, the Lagrange multipliers of which are used to construct Benders cuts that are to be appended to the MP in the following iteration. Generators' outputs are constrained with (30) and (31), while (32) – (37) model energy storage operation. Note that t^0_b, t^1_b and t^e_b denote the time period directly before the first hour of operation in block b, the first time period in b, and the final time period in b, respectively. Power flows are determined with (38) and constrained to the available capacity with (39), while (40) is the power balance equation.

The MBD solution algorithm is summarized in Fig. 1. The system operation problem is built up from below by appending Benders cuts in (25) over iterations until the MP objective function is equivalent to that of the original problem. The lower bound and upper bound are defined with (41) and (42), respectively, while (43) is the convergence criterion, where ϵ_{BD} is a value close to zero.

$$LB^{(k)} = \overline{\psi}^{M(k)} = \sum_{m\in\Omega_M} \pi_m \left[r^I_{\epsilon_m} \Psi^I_m \left(\overline{\mathrm{x}}^{(k)} \right) + \sum_{b\in\Omega_B} \overline{\alpha}^{(k)}_{m,b} \right] \tag{41}$$

$$UB^{(k)} = \sum_{m\in\Omega_M} \pi_m \left[r^I_{\epsilon_m} \Psi^I_m \left(\overline{\mathrm{x}}^{(k)} \right) + \sum_{b\in\Omega_B} \overline{\psi}^{S,(k)}_{m,b} \right] \tag{42}$$

$$\frac{UB^{(k)} - LB^{(k)}}{LB^{(k)}} < \epsilon_{BD} \tag{43}$$

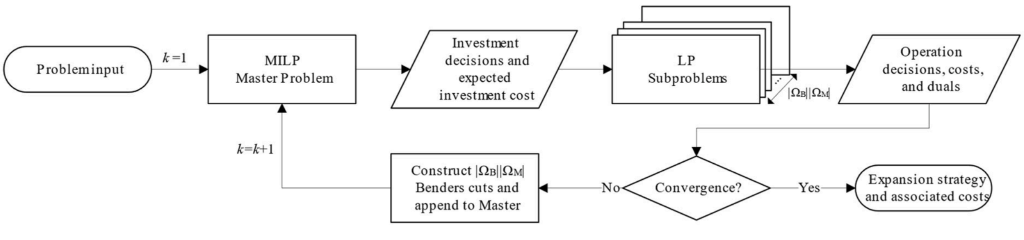
3. Learning-enhanced benders decomposition method

The proposed methodology leverages ML techniques to enhance the MBD algorithm, which is based on generating cutting planes to approximate future-cost functions. We extend the novel theory in cuts classification to a multi-stage multicut formulation and propose appropriate revisions where previous research falls short for the application to power system optimization problems with complete recourse. The ML-MBD method employs supervised learning techniques that take a set of cut characteristics as inputs to distinguish between effective and ineffective cuts based on user-defined criteria with varying strictness. It involves two main stages: 1) data sampling and classifiers training, and 2) problem execution.

3.1. Machine learning method

We consider the use of supervised learning techniques to train the classifiers in this research. Compared to other options, they have the advantage of being relatively well understood in the field of artificial intelligence, with a simpler implementation and faster convergence.

To maximize the performance of the proposed method, we explore and evaluate different ML techniques [26], as detailed in Subsection 3.3 Nonetheless, models based on Support Vector Machines (SVM), Decision Trees (DT), and Random Forests (RF) presented the best performance. As authors in [25] argue the importance of correctly classifying effective cuts, the precision-recall (PR) characteristics of the ML models, which measure the trade-off between the accuracy and completeness of positive predictions, have a direct impact on the method's performance. Models trained with RF incurred in better PR scores and are therefore used in the presented case studies.



Probleminput
k=1
MILP
Master Problem
Investment decisions and expected investment cost
LP
Subproblems
|ΩB||ΩM|
Operation decisions, costs, and duals
k=k+1
Construct |ΩB||ΩM| Benders cuts and append to Master
No
Convergence?
Yes
Expansion strategy and associated costs

Fig. 1. Workflow of the Multicut Benders Decomposition Approach.

3.2. Target selection and labeling

The classification target could be any variable that informs on the effectiveness of an individual cut in improving the approximation of the future-cost function within the Benders algorithm. We therefore refer to the target as a cut performance indicator (CPI). References [24] and [25] both propose the LB as CPI, which would be the natural choice given that it is the objective value of the MP and it increases monotonically. However, using the LB as CPI may lead to inefficient behavior of the UB and slower convergence as a result, an aspect that previous references do not investigate. Furthermore, the UB does not behave predictably and appending only the cuts that cause its value to decrease could be extremely valuable, thus worth exploring as a CPI. In the case studies, we identify shortcomings of both LB and UB as CPIs, and instead propose to use UB and LB sequentially as CPIs to mitigate limitations.

For the avoidance of doubt, we henceforth refer to the proposed method as ML-MBD-C, as it involves the combined use of both CPIs. We use ML-MBD-L and ML-MBD-U to indicate the use of a single CPI, namely the LB or the UB, respectively, which we apply for feature selection in Subsection 3.3 and to evaluate the performance of ML-MBD-C in Section 4.

Next, a cut performance metric (CPM) is introduced that provides information on the improvement of the CPI as a result of appending a cut. Since such a metric is continuous, a label transformation function is required to separate effective and ineffective cuts for the binary classification task. Equations (44) and (45) define the CPM with reference to each CPI before and after cut c has been added and evaluate it against a predefined threshold θ. Cuts that satisfy the inequalities are labeled 1, or labeled 0 otherwise.

$$CPM_c^{LB} = \frac{LB^{(after)} - LB^{(before)}}{LB^{(after)}} \geq \theta^{LB} \tag{44}$$

$$CPM_c^{UB} = \frac{UB^{(after)} - UB^{(before)}}{UB^{(after)}} \leq \theta^{UB} \tag{45}$$

Unlike the monotonically increasing LB, changes in UB can be both positive and negative, so θ^{UB} is a negative value since improvement implies a decrease in the UB. The threshold is an important factor in ML-MBD as it ultimately influences the conservativeness of the classifier. A stricter θ^{CPI} would result in fewer cuts appended to the MP, risking slower convergence, while an overly permissive θ^{CPI} could render the use of ML redundant. The choice of threshold is challenging for two main reasons. Firstly, because there is no established rule as to what constitutes an effective cut [27]. Secondly, because the rate of convergence of MBD has a tailing off effect such that the changes in the LB and UB diminish in latter iterations, causing distribution shift and implying that a single threshold might not be appropriate for all cuts. For these reasons, we use a set of thresholds with decreasing strictness, similar to the approach in [24]. We define N_ζ thresholds in $\Omega_\theta^{CPI} = \left\{\theta_1^{CPI}, \ldots, \theta_{N_\zeta}^{CPI}\right\}$, for each CPI, based on observations of CPM_c^{LB} and CPM_c^{UB} for each cut c in the respective training datasets. Thresholds in Ω_θ^{LB} are chosen uniformly with values between, but excluding, the largest and the smallest observed CPM_c^{LB}, and the same is done for Ω_θ^{UB} with values greater than the smallest observed CPM_c^{UB} up to, and including, zero. This approach to thresholds selection is adopted because it generalizes well to any TEP problem.

3.3. Feature selection

The aim of feature selection is to identify cut characteristics and complementary information that best inform the ML model on a cut's potential for CPI improvement. It is important to base this selection on both ML metrics and ML-MBD performance. We aim to define problem-independent features such that the method generalizes to a range of modified TEP problems, as discussed further in Subsection 4.2.3.

Here, we investigate ten potential features in combinations of two or more, with LB and UB as CPIs. We evaluate their correlation and feature importance (Gini Importance and Mean Decrease in Accuracy), as well as classification scores (receiver operating characteristic (ROC), balanced accuracy, and F-score) of models trained with SVM, DT, RF, logistic regression, and k-nearest neighbors [26]. The combinations with highest scores are selected to support ML-MBD-L and ML-MBD-U in solving two trial TEP problems, resulting in 72 test examples upon which the final feature selection is made based on the trade-off between the number of cuts and the number of iterations at convergence.

This is illustrated in Fig. 2 for all converging problems with an optimal cost within 1.5% of that obtained using MBD. It shows that despite a high classification score, a model based on an inadequate combination of features could prohibitively increase the required number of iterations or lead to larger MP sizes. Therefore, the aim is to select feature and target combinations resulting in an ML-MBD performance that lies in the lower left corner of the plot.

Table 1 summarizes the investigated features and the observations in terms of their efficacy. Note that quantities here are defined on the set Ω_s, where each s translates to a unique (m, b) in line with notation in Section 2. The symbol Δ denotes the difference between the values in iterations (k) and $(k-1)$.

The first feature is cut origin, namely the SP from which the cut originates. The second is the iteration in which the cut is generated, which might carry useful information given the tailing off effect of MBD. Then, Cut Violation is a feature proposed in [24] that is related to the feasible space of the MP that can be removed by appending the cut. It is inherently a useful measure; however, note that it is defined only at the current MP candidate solution $(\overline{x}, \overline{\alpha}_s)$ and resulting $\overline{\lambda}_s$, without any indication of how it translates to the rest of the feasible region and, by extension, to subsequent candidate solutions. Although this feature might contain useful information and is shown empirically as important, it does not carry full information and could be developed further in future work.

The objective function value of the SP in which the cut is generated and the corresponding dual variables are contained within Cut Violation. Nonetheless, our investigation reveals that they could act as complementary features that increase the performance of ML-MBD. A similar observation is made for the final two features – solution proximity and value of the slack variable. The slack variable is penalized with a high cost in the SP objective function and could be considered as a reflection of the quality of the MP candidate solution. Intuitively, it should carry important information about the cuts in a problem with complete recourse.

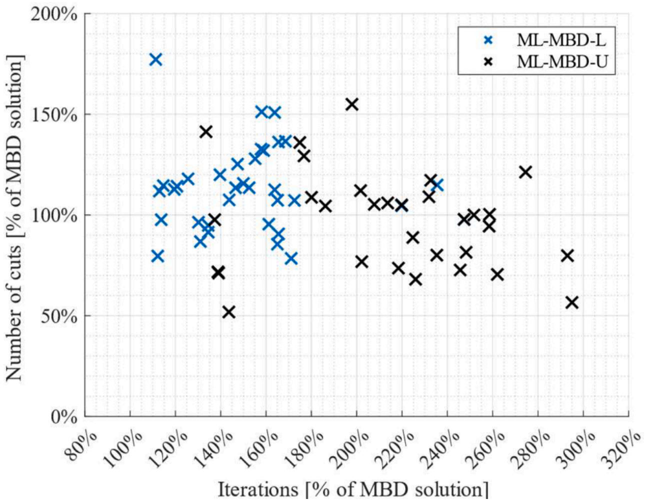


Fig. 2. Trade-off between number of cuts and number of iterations of the test models as a percentage of the non-ML MBD solution.

Table 1
Feature Selection Summary.

Feature		CPI = LB		CPI = UB	
		Observation	Use	Observation	**Use**
Cut origin:	$\mathbb{P}_s^S$	detrimental		irrelevant	
Iteration:	k	informative	✓	informative	✓
Cut Violation:	$(h_s - F_s\overline{\mathrm{x}})^T\overline{\lambda_s} - \overline{\alpha}_s$	informative	✓	informative	✓
SP objective:	$\overline{\psi}_s^S$	irrelevant		informative (supplementary)	✓
Change in SP objective:	$\Delta\overline{\psi}_s^S$	informative (supplementary)	✓	informative (supplementary)	
Dual variables:	$\overline{\lambda}_s$	irrelevant		irrelevant	
Change in dual variables:	$\Delta\overline{\lambda}_s$	informative (supplementary)	✓	irrelevant	
Solution proximity:	$\lvert\Delta\overline{\mathrm{x}}\rvert$	informative (supplementary)	✓*	informative (supplementary)	✓
Value of slack variable:	$\overline{y}_s^\sigma$	informative (supplementary)		informative (supplementary)	✓
Change in slack variable value:	$\Delta\overline{y}_s^\sigma$	informative (supplementary)	✓	informative (supplementary)	

Algorithm 1
Data sampling and classifiers training.

```
Step 0: Initialization
Set $N_\zeta$, $\epsilon_{ROC}^{min}$, $\epsilon_{PR}^{min}$, $\Omega_c^{(0)} = \emptyset$, $\Theta_i^{LB} \leftarrow \emptyset$, $\Theta_i^{UB} \leftarrow \emptyset$
Set $k = 0$, $UB^{(0)} = \infty$, $LB^{(0)} = 0$, $\epsilon_{BD}$, $N_S$, $j = 0$
Step 1: Sample using modified MBD
$j = j + 1$
While $k \leq j \times N_S$ and $\frac{UB^{(k)} - LB^{(k)}}{LB^{(k)}} > \epsilon_{BD}$
    $k = k + 1$
    For all $c$ in $\Omega_c^{(k-1)}$
        Solve $\mathbb{P}^{M(k)}$ with $c$ and all cuts in $\Omega_c^{(0)}$ ... $\Omega_c^{(k-2)}$, and obtain $\overline{\psi}_c^{M(k)}$
        If $k > 1$, compute $CPM_c^{LB} = \frac{\left|\overline{\psi}_c^{M(k)} - \overline{\psi}_c^{M(k-1)}\right|}{\overline{\psi}_c^{M(k)}}$
    Solve $\mathbb{P}^{M(k)}$ with all $c$ in $\Omega_c^{(0),\ldots,(k-1)}$ and obtain $\overline{\psi}^{M(k)}$ and $\overline{\mathrm{x}}^{(k)}$
    Solve $\mathbb{P}_s^{S(k)}$ $\forall s \in \Omega_s$, obtain $\overline{\psi}_s^{S(k)}$ and $\overline{\lambda}_s^{(k)}$, and save cuts in $\Omega_c^{(k)}$
    Extract features and save in $\Omega_{F(c)}^{LB}$ and $\Omega_{F(c)}^{UB}$, respectively, $\forall c \in \Omega_c^{(k)}$
    Compute $UB^{(k)}$ and $LB^{(k)}$
    If $k > 1$, compute $CPM_c^{UB} = \frac{\left|UB^{(k)} - UB^{(k-1)}\right|}{UB^{(k)}}$ $\forall c \in \Omega_c^{(k-1)}$
Step 2: Labeling
Determine $\Omega_\theta^{LB}$ based on all observed $CPM_c^{LB}$ values
For $i = 1, .., N_\zeta$ and $c = 1, .., |\Omega_c|$
    If $CPM_c^{LB} \geq \theta_i^{LB}$, $\Theta_i^{LB} \leftarrow \{\Omega_{F(c)}^{LB}, 1\}$
    Else      $\Theta_i^{LB} \leftarrow \{\Omega_{F(c)}^{LB}, 0\}$
Determine $\Omega_\theta^{UB}$ based on all $CPM_c^{UB}$ values
For $i = 1, .., N_\zeta$ and $c = 1, .., |\Omega_c|$
    If $CPM_c^{UB} \leq \theta_i^{UB}$, $\Theta_i^{UB} \leftarrow \{\Omega_{F(c)}^{UB}, 1\}$
    Else      $\Theta_i^{UB} \leftarrow \{\Omega_{F(c)}^{UB}, 0\}$
Step 3: Training
Perform under-sampling in $\Theta_i^{LB}$ and $\Theta_i^{UB}$ $\forall i \in \{1, .., N_\zeta\}$
Train $\zeta_1$, $\zeta_{median}$ and $\zeta_{N_\zeta}$ and obtain $\epsilon_{ROC}$ and $\epsilon_{PR}$ for each
If $\epsilon_{ROC} < \epsilon_{ROC}^{min}$ or $\epsilon_{PR} < \epsilon_{PR}^{min}$, for $\zeta_1$, $\zeta_{median}$ or $\zeta_{N_\zeta}$, go to Step 1
Else Finish training remaining models in $\Omega_\zeta$
END
```

There are a few observations on the final feature selection marked with (✓) in the table. Firstly, different features are preferred depending on the CPI. For instance, ML-MBD-L performance is improved if supporting features are formulated as a change in value from the previous iteration, which is not the case for ML-MBD-U. Similarly, solution proximity is used with both CPIs, but the binary decision variables are excluded from the ML-MBD-L features (✓*). Secondly, our investigation finds that the two-feature approach in [24] leads to underperformance, both in terms of ML metrics and number of cuts at convergence, and that the additional features proposed in [25] are not applicable to our decomposition technique. Lastly, unlike in both previous papers, which focus on other problems, we find that cut origin is not an effective predictor for a cut's effectiveness in our multi-stage formulation, and in fact, it is detrimental to the performance of ML-MBD-L.

3.4. Sampling and training procedures

The proposed offline sampling and training procedure is summarized in Algorithm 1 and consists of two parts – solving a TEP problem with a modified MBD algorithm to extract cut information and subsequently training the ML models. The sampling problem should be of a similar structure as the one in which the ML-MBD will be applied, for example involving the same network and a simplified uncertainty representation.

In the first part, the MDB is modified so as to reveal unique information on the effectiveness of each cut individually. This is achieved by re-solving the MP with only one newly appended cut and measuring the improvement in the LB as a direct consequence of that cut alone. Then, all cuts are appended to progress the algorithm and continue to SPs execution. The improvement in UB is measured at the end of the iteration as an aggregated contribution of all cuts appended in that iteration. To obtain individualized UB improvement information, similar to LB, $|\Omega_S|^2$ additional SP instances would need to be solved, which prohibitively increases the sampling time for problems with large scenario trees and the benefit of this information does not justify the increased time demand for smaller problems. Finally, the features are extracted at the end of each iteration.

Step 1 is run for N_S iterations, which is significantly fewer than the number of iterations required to achieve convergence. Then, Ω_θ^{CPI} are determined for both LB and UB, and cut samples are labelled accordingly. With that, N_ζ datasets are created for each CPI (Θ_i^{LB} and Θ_i^{UB} in Algorithm 1) with $|\Omega_c|$ rows and $|\Omega_F^{CPI}| + 1$ columns. They contain the same number of rows (samples) and identical first $|\Omega_F^{CPI}|$ columns (features), but the labels contained in the final column differ due to the varying strictness of the thresholds. Finally, an under-sampling procedure is performed for each dataset to balance the classes before training.

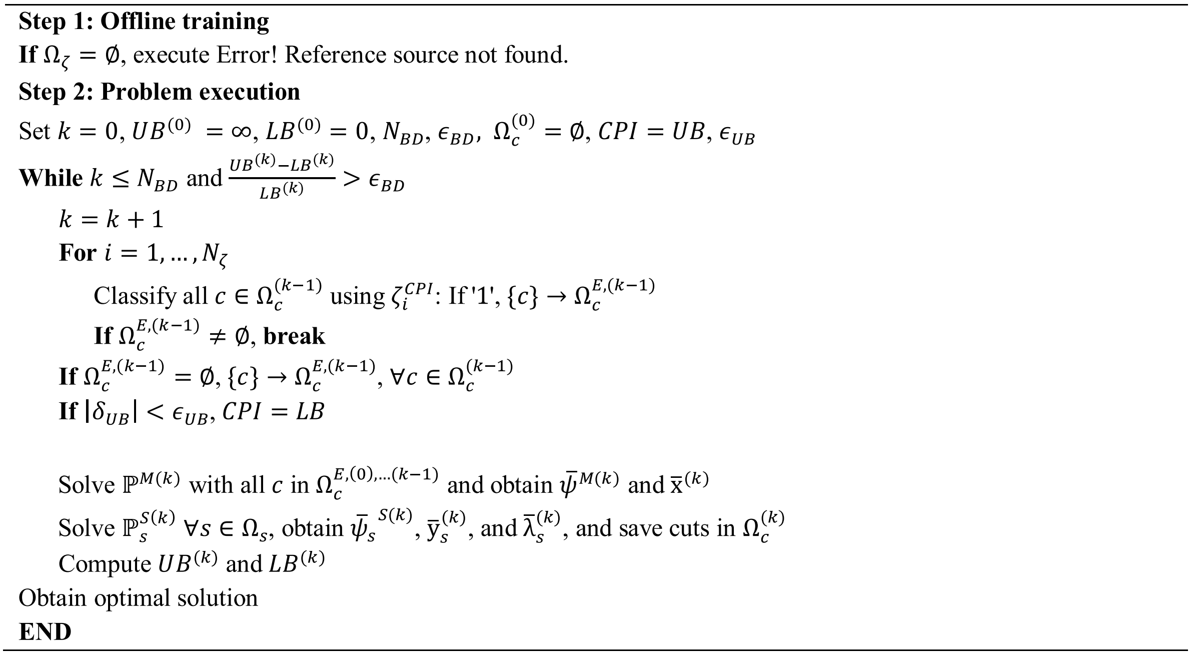
Classifiers $\zeta_1^{CPI}, \zeta_{med}^{CPI}$ and $\zeta_{N_\zeta}^{CPI}$ are trained on $\Theta_1^{CPI}, \Theta_{med}^{CPI}$ and $\Theta_{N_\zeta}^{CPI}$, respectively, where Θ_{med}^{CPI} is the dataset corresponding to the median threshold in Ω_θ^{CPI}. They are evaluated using their ROC and PR curves, such that if the areas under the ROC and PR curves are below predefined minimum levels (ϵ_{ROC}^{min}, ϵ_{PR}^{min}), the algorithm returns to Step 1 where the modified MBD is resumed for another N_S iterations. The termination criterion is based on the performance of three ML models because each is trained with a different number of samples due to the class imbalance of the respective dataset and subsequent under-sampling, which could result in different performances of the models. The remaining classifiers in Ω_ζ are trained at the end.

3.5. Solution algorithm

The proposed method is summarized in Algorithm 2. Step 1 is performed only if suitable classifiers do not already exist from a previously concluded training procedure, as supported by the generalization properties evidenced in Subsection 4.2.3.

Step 2 describes the ML-supported solution of a TEP problem, as illustrated in Fig. 3. Cuts are classified using $\Omega_\zeta^{CPI} = \left[\zeta_1^{CPI}, \dots, \zeta_{N_\zeta}^{CPI}\right]$, starting with the strictest model ζ_1^{CPI} until at least one cut is identified as effective. Should no cut be deemed effective by any classifier in Ω_ζ^{CPI}, then the algorithm proceeds as traditional MBD in that iteration. Stricter classifiers may become redundant considering the distribution shift as the Benders algorithm progresses and could therefore be discarded. However, we find that although CPI improvements decrease on average

Algorithm 2
ML-MBD-C.



```
Step 1: Offline training
If $\Omega_\zeta = \emptyset$, execute Error! Reference source not found.
Step 2: Problem execution
Set $k = 0, UB^{(0)} = \infty, LB^{(0)} = 0, N_{BD}, \epsilon_{BD}, \Omega_c^{(0)} = \emptyset, CPI = UB, \epsilon_{UB}$
While $k \le N_{BD}$ and $\frac{UB^{(k)}-LB^{(k)}}{LB^{(k)}} > \epsilon_{BD}$
    $k = k + 1$
    For $i = 1, \dots, N_\zeta$
        Classify all $c \in \Omega_c^{(k-1)}$ using $\zeta_i^{CPI}$: If '1', $\{c\} \to \Omega_c^{E,(k-1)}$
        If $\Omega_c^{E,(k-1)} \neq \emptyset$, break
    If $\Omega_c^{E,(k-1)} = \emptyset, \{c\} \to \Omega_c^{E,(k-1)}, \forall c \in \Omega_c^{(k-1)}$
    If $|\delta_{UB}| < \epsilon_{UB}, CPI = LB$

    Solve $\mathbb{P}^{M(k)}$ with all $c$ in $\Omega_c^{E,(0),\dots,(k-1)}$ and obtain $\bar{\psi}^{M(k)}$ and $\bar{x}^{(k)}$
    Solve $\mathbb{P}_s^{S(k)}$ $\forall s \in \Omega_s$, obtain $\bar{\psi}_s^{S(k)}$, $\bar{y}_s^{(k)}$, and $\bar{\lambda}_s^{(k)}$, and save cuts in $\Omega_c^{(k)}$
    Compute $UB^{(k)}$ and $LB^{(k)}$
Obtain optimal solution
END
```

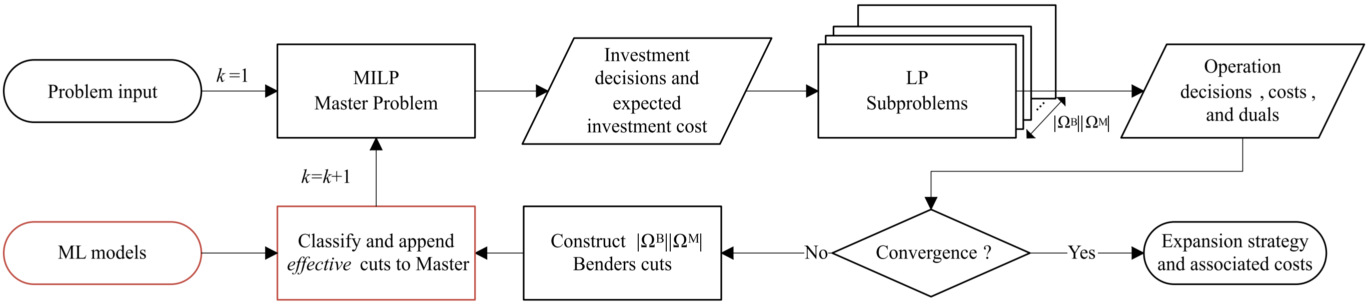

Problem input
k=1
MILP
Master Problem
Investment decisions and expected investment cost
LP
Subproblems
|ΩB||ΩM|
Operation decisions , costs , and duals
k=k+1
ML models
Classify and append effective cuts to Master
Construct |ΩB||ΩM| Benders cuts
No
Convergence ?
Yes
Expansion strategy and associated costs

Fig. 3. Workflow of the ML-MBD-C solution algorithm.

over iterations, they do not decrease monotonically and often a stricter classifier can still be utilized after it has failed to label any cuts as useful in a previous iteration, potentially yielding greater benefits compared to the approach in [24]. For the proposed ML-MBD-C, the CPI is updated when the moving average of CPM^{UB} in the previous 10 iterations (δ_{UB}) falls below a pre-determined value ϵ_{UB}. For the two testing versions of the method, this action can be ignored in ML-MBD-U, while ML-MBD-L only requires adequate CPI initialization.

In summary, the proposed ML-MBD framework for solving TEP under uncertainty problems consists of the steps illustrated in the flowchart in Fig. 4, and are as follows:

1) Start: The process begins with initializing the problem parameters and variables.
2) Problem Formulation/ Problem Inputs Update: The multi-stage stochastic TEP problem is defined. In a first execution, this includes specifying the objective function, decision variables, constraints, and the scenario tree and other input parameters. In subsequent model executions, the input parameters are updated correspondingly.
3) Decomposition: The problem is decomposed into a MP and SPs. The MP includes the investment decisions, while the SPs handle the operational aspects for each scenario tree node and representative block.
4) Machine Learning Integration: Data on the cuts generated in the initial iterations is sampled and used to train ML classifiers, as defined with Algorithm 1. These classifiers aim to distinguish between effective and ineffective cuts based on their impact on the convergence process, and are designed to generalize to modified problems such that this step can be omitted in subsequent model executions.
5) Iterative Solution Process: The solution process is described with Algorithm 2 and illustrated in Fig. 3. It involves building up the MP with effective cuts, solving the SPs based on the candidate MP solution in the current iteration, generating new cuts, and classifying them. This process repeats until the convergence criteria are met.
6) End: The results of the TEP problem are outputted and evaluated.

3.6. Computational effort

ML integration adds a certain overhead to the full solution process. The bulk of it is in fact in Step 1 of Algorithm 1 and it depends on the problem size and number of iterations required for generating an adequate dataset. In the studies presented here, we do not impose any expectations or restrictions on the total training time and $N_S = 20$ is used throughout. In all cases, Algorithm 1 terminated after two or three repetitions of Step 1 for a high standard for ϵ_{ROC}^{min} and ϵ_{PR}^{min} of 0.92. In terms of training in Step 3, the ML techniques employed require a relatively low number of samples to produce well-performing models and training time is insignificant. In our investigations, the training time of a single classifier typically ranged between 0.5 and 3 seconds, with only a few instances taking between 20 and 30 seconds, depending on the number of samples and the ML technique. Nevertheless, the sampling and training process does not add to the computational efforts of the problem execution itself because it is performed offline and only once with application to different problems, enabled by the generalization properties of the method. The classification step in Algorithm 2 added at most 0.6 seconds, including the time required for memory communication between scripts, which is negligible considering the total time of a single MBD iteration.

4. Case studies

We apply ML-MBD to the problem of co-expansion of the transmission system with battery energy storage (BES) as a non-network investment alternative, as presented in Subsection 2.3. The proposed 'ML-MBD-C' is benchmarked against the classical MBD and evaluated with respect to 'ML-MBD-L' and 'ML-MBD-U' that use a single CPI as done in previous research [24,25]. The computational benefits of the proposed method are analyzed with respect to the number of appended cuts, which is directly related to the memory requirements of the problem, and MP solution times. The optimization problems are implemented in FICO XPress 8.13 and the ML scripts in Python 3.9. Results are obtained on a Dual Xeon computer with 512 GB RAM. All cases are solved with $N_\zeta = 10$, $\epsilon_{BD} = 0.01$, and $N_{BD} = 1{,}000$.

4.1. Description

The case studies are based on the IEEE 24-Bus Reliability Test System (IEEE24) and the IEEE 118-Bus System (IEEE118), the topologies of which can be found in [28]. Line capacities in both networks are modified to allow unconstrained operation in the initial system state, as explained in [5], while the IEEE24 case additionally includes solar and

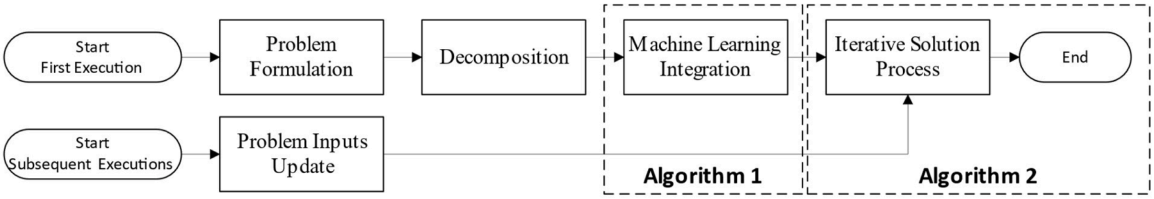


Fig. 4. Flowchart of the ML-Enhanced Multicut Benders Decomposition Framework.

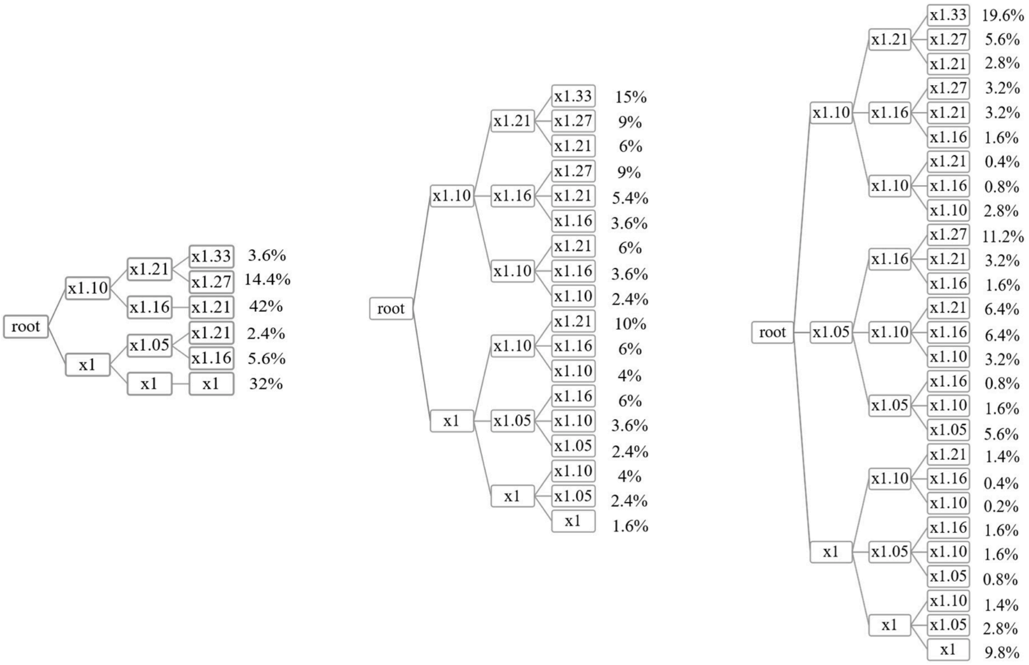


Fig. 5. Scenario trees describing the three test cases.

wind generation as done in [8]. Generator capacities are scaled up to ensure system adequacy throughout the planning horizon.

Three test cases of different sizes are developed subject to uncertainty in the level and rate of peak demand increase, as presented with the scenario trees in Fig. 5. All cases involve four decision-making stages over a 40-year horizon, but the number of scenarios and scenario tree nodes is different. The first tree shows 6 scenarios over 13 nodes, the second 18 scenarios and 27 nodes, and the final case considers 27 scenarios over 40 nodes. Each node represents an investment decision-making point and is associated with a unique system operation problem. Short-term variability is accounted for with four representative blocks. Demand profiles are obtained from IEEE24 and IEEE118 data and renewable generation time-series from [8]. Since each (m, b) pair represents a separate SP, the first case involves $|\Omega_S| = 52$, the second $|\Omega_S| = 108$, and the third $|\Omega_S| = 160$, and the same number of Benders cuts per iteration.

Line reinforcements are subject to an annualized fixed cost of $121,600 per km, annualized variable cost of $76 per MW and km, and a commissioning delay of one stage. BES units are assumed to have a discharging duration of 2 hours and 90% efficiency. They are subject to an annualized investment cost of $102,000 per MW and are available in the same stage in which the investment decision is made. Case studies on the IEEE24 involve the possibility to invest in BES in all buses, while 13 candidate buses are selected in IEEE118 in keeping with [5].

Table 2
Results of the case studies.

Network	Scenario tree	Method	Iterations	Total cuts	Optimality gap [%]	Total solution time [h]	Best LB [£ million]	Best UB [£ million]
IEEE24	**13 nodes**	MBD	88	4,524	0.89	1.81	6,285	6,317
		ML-MBD-L	127	4,885	0.91	2.64	6,261	6,314
		ML-MBD-U	287	3,080	0.36	3.31	6,311	6,371
		ML-MBD-C	116	4,398	0.86	2.16	6,268	6,314
	27 nodes	MBD	60	6,372	0.58	9.57	21,964	22,175
		ML-MBD-L	75	6,534	0.84	10.89	21,821	22,128
		ML-MBD-U	164	2,110	0.91	13.01	21,715	22,227
		ML-MBD-C	65	4,765	0.86	9.14	21,929	22,121
	40 nodes	MBD	58	9,120	0.83	11.35	38,144	39,038
		ML-MBD-L	67	7,569	0.77	10.52	38,180	38,504
		ML-MBD-U	1,000	4,880	15.04	23.08	34,131	39,532
		ML-MBD-C	63	5,970	0.81	9.78	38,146	38,516
IEEE118	**13 nodes**	MBD	156	8,060	0.50	7.22	58,451	58,969
		ML-MBD-L	175	6,434	0.84	6.23	57,542	58,460
		ML-MBD-U	224	4,179	0.95	6.78	57,746	58,668
		ML-MBD-C	179	6,276	0.72	6.35	57,538	58,476
	27 nodes	MBD	250	26,892	0.96	35.44	56,353	56,679
		ML-MBD-L	305	22,189	0.78	33.99	56,022	56,499
		ML-MBD-U	812	13,319	0.93	59.00	55,891	56,403
		ML-MBD-C	278	21,261	0.88	32.22	56,350	56,632
	40 nodes	MBD	175	27,840	2.35	240.00	55,503	55,964
		ML-MBD-L	232	26,542	0.85	165.15	55,519	55,946
		ML-MBD-U	1,000	9,267	13.66	222.63	52,243	56,798
		ML-MBD-C	226	25,702	0.96	159.90	55,503	55,957

4.2. Results

4.2.1. Validation and scalability

Table 2 summarizes the results at termination of the solution algorithm. As the contributions of this research concern the solution approach, we focus on convergence and obtained costs to validate the method, while the exact investment strategies are not of consequence. We stress, however, that all convergent ML-MBD cases result in identical expansion decisions as with MBD. In terms of cost, the results table shows that all expected total system costs obtained using a version of the ML-MBD method fall within 0.5% of the corresponding MBD reference values, except on two occasions with ML-MBD-U when costs differ by 0.86% and 1.34%. This signals that ML-MBD converges to the correct TEP solution, as further evidenced by the values for best UB in Fig. 6.

In terms of convergence, we emphasize that MBD failed to find a solution for the IEEE118 40-node problem as the algorithm stopped progressing after 7 days and was terminated after 10 days. In contrast, the proposed ML-MBD-C reaches convergence, highlighting its primary contribution – the ability to solve larger stochastic optimization problems while preserving solution quality. Fig. 7 demonstrates the rates of MP growth using the four considered decomposition methods. MBD adds 160 constraints in every iteration, regardless of their effectiveness, and so fails to progress past iteration 175. Conversely, all ML-MBD versions manage the size growth by discarding cuts classified as ineffective, leading ML-MBD-C and ML-MBD-L to convergence within 226 and 232 iterations, respectively. Finally, the figure reveals that the problem grows most conservatively with ML-MBD-U, although in this case it reaches the 1,000-iteration limit before convergence.

To highlight the computational benefit of ML-MBD with respect to problem size across all considered problems, Fig. 8 shows the percentage reduction in the total number of appended cuts compared to the MBD solution as a reference. ML-MBD-C consistently converges to a solution that is marginally better than that of MBD, as supported by Fig. 6, while also appending between 3% and 35% fewer cuts. Note that for the largest problem, the benefit is much more significant than the figure suggests because, as previously noted, MBD fails to converge. Moreover, the results show certain flaws in both testing versions of ML-MBD. Although the problem grows the least with ML-MBD-U, appending between 32% and 67% fewer cuts than MBD, convergence within the limit is not always reached. ML-MBD-L, on the other hand, underperforms in the smaller-sized problems. This validates the proposed approach involving the use of both UB and LB as CPIs.

4.2.2. Computational performance

The MP is NP-hard and the worst-case complexity increases exponentially with the problem size. However, there are no proven rules regarding the average computational complexity of NP-hard problems and solution times in practice depend on additional factors, such as CPU usage. Therefore, the computational performance of the proposed method must be evaluated empirically and we analyze the results only for direct comparison between the ML-assisted and classical MBD on

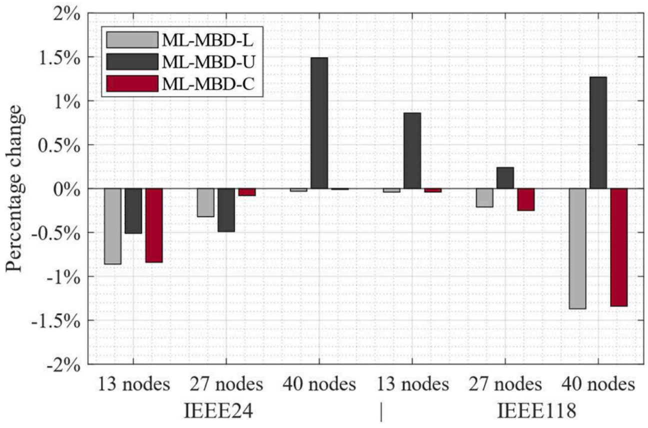


Fig. 6. Percentage change between the best upper bound obtained using ML-MBD methods and the reference values obtained using MBD.

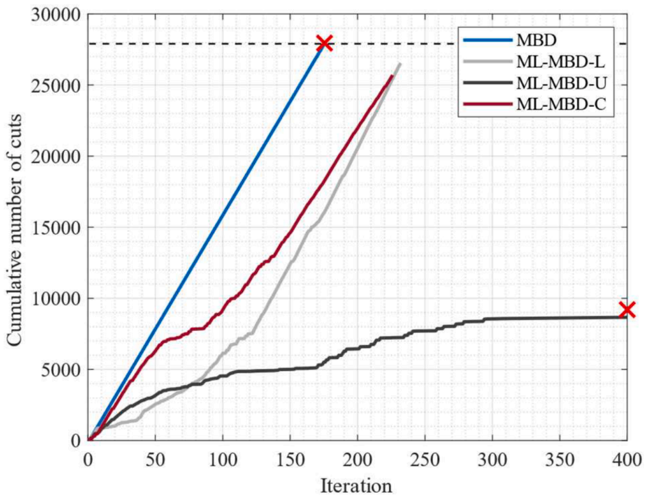


Fig. 7. Cumulative number of appended cuts over iterations for the four MBD methods in the IEEE118 27 nodes case.

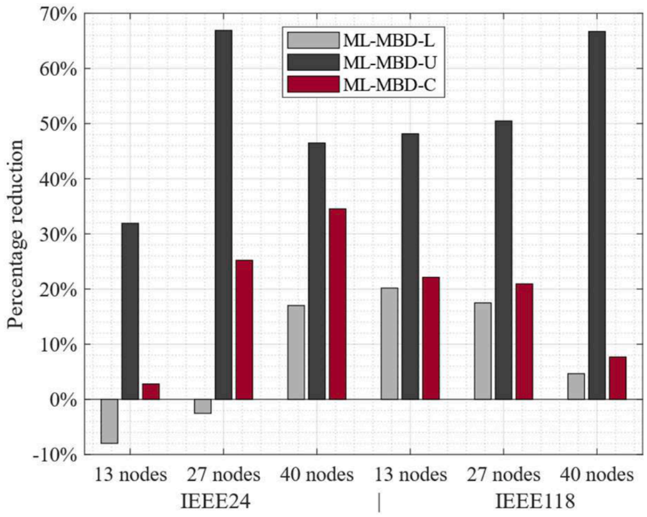


Fig. 8. Percentage reduction in the total number of cuts at convergence using ML-MBD methods compared to MBD.

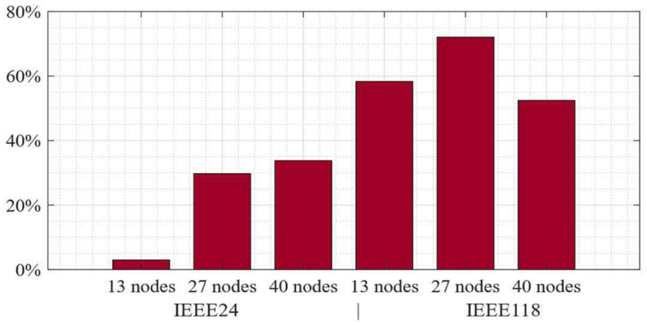


Fig. 9. MP solution time improvement with ML-MBD-C normalized for the number of iterations and as a percentage of the MBD reference values.

separate test cases. Notwithstanding, the results show a clear trend that less time is spent solving MP instances with the proposed method. Fig. 9 demonstrates the reductions in average MP solution time with ML-MBD-C. The computational time savings increase with the size of the TEP problem, up to 72.1%. Note again that the final case does not reflect the full benefit because MBD leads to an intractable MP in iteration 176. The percentage reduction in MP solution time up to this iteration is 52.4%.

Although ML-MBD-C requires fewer cuts to find a solution, it completes more iterations compared to MBD, which could lead to an increase in total solution time in some cases. Note, however, that the impact of the number of iterations on total solution time could be largely reduced with parallel execution of SPs, which we avoid in order to identify direct effects of the application of the proposed method. Nevertheless, the considerable MP solution time savings with ML-MBD-C in the larger problems compensate for the time demands of additional iterations and lead to reductions in total solution times of up to 33.4%, as observed in the last case in Table 2. This fact highlights the pertinence of the proposed method to large-scale problems, especially those that challenge tractability limits.

4.2.3. Generalizability

Lastly, we test the generalizability properties of the method on modified problems with respect to aspects that might occur in implementation, such as changes to the assumptions on long-term uncertain parameters, short-term operational variability, scenario tree transition probabilities, or the size and shape of the scenario tree. Specifically, we take ML-MBD-C trained on the IEEE118 problem with a 13-node scenario tree and apply it directly, without re-training, to the following problems:

- IEEE118 and 13-node scenario tree with different transition probabilities
- IEEE118 and 13-node scenario tree with different uncertainty assumptions on long-term demand increase
- IEEE118 and 13-node scenario tree with different short-term variability time-series (representative blocks)
- IEEE118 and 27-node scenario tree.

In all cases, ML-MBD-C converges to an equivalent solution to the one obtained with MBD, but with reduced computational burden as highlighted in Fig. 10, which shows the significant MP solution time savings obtained. Fig. 11 shows the rate of MP size growth over iterations for the four cases and demonstrates that the proposed method is able to manage the problem size despite the ML models being trained on a different problem. ML-MBD-C achieves a decrease in the total number of constraints in the MP of the final iteration of 18.9%, 27.0%, 15.5% and 1.2%, for the four cases respectively. The results demonstrate that the proposed method generalizes exceptionally well to modified problems without repeated execution of Algorithm 1.

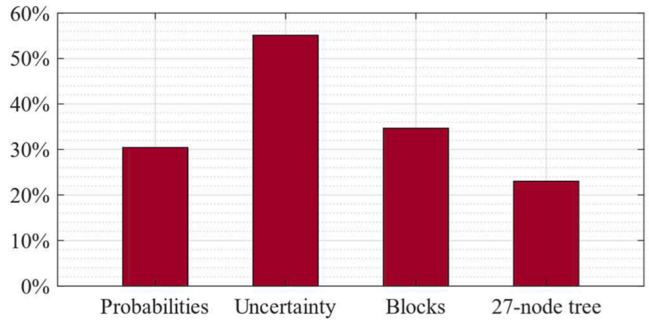


Fig. 10. MP solution time improvements with ML-MBD-C normalized for the number of iterations and as a percentage of the MBD reference values.

5. Discussion

Modern TEP exercises are witnessing a trend of increasing complexity, leading to computationally intractable optimization problems. As such, modeling real-world details that allow for more informed decision-making is limited by computational scalability, which is one of the principal challenges in power system problems [3]. Techniques like scenario reduction or recent learning-based solutions have been applied to mitigate tractability issues, but such approaches alter the problem structure and may lead to different or suboptimal solutions. In contrast, the proposed method preserves the original problem and leads to a solution of the same quality. Most notably, we demonstrate its application to solve a highly complex TEP problem that is intractable with a state-of-the-art decomposition method. Moreover, the results in Section 4 prove that the application of ML-MBD could be highly beneficial even for instances that can be solved with the classical decomposition.

The proposed ML-MBD-C outperforms MBD in all cases as it is consistently able to find a tighter UB, observed in Fig. 6, with the help of fewer Benders cuts, shown in Fig. 8, thus implying a notable reduction in memory requirements and leading to the great computation time savings highlighted in Fig. 9. The requirement of additional iterations could be considered a drawback of the proposed method; however, as evidenced in Table 2, the computational impact of extra iterations is compensated for, ultimately leading to total solution time savings that increase with problem size.

Notwithstanding the low computational overhead of the ML-MBD, the sampling procedure involves solving a number of iterations of MBD. This fact is justified for two reasons: Firstly, ML-MBD could enable the convergence of certain problems that are otherwise intractable; and secondly, TEP studies typically involve solving the problem multiple times with modified inputs. In Subsection 4.2.3, we show that the method generalizes well to modified problems and this property would allow the user to solve all subsequent studies without the need to re-sample and re-train ML models. Furthermore, the problem-independent feature selection could result in the applicability of ML-MBD to problems with a similar structure that require repeated execution, such as stochastic unit commitment, which could benefit with faster MP computation and the opportunity to increase modeling complexity.

In this research, we leverage supervised learning techniques because of their straightforward implementation and short training times, which is crucial to minimize the burden of the procedure described in Subsection 3.4. An alternative would be to employ unsupervised learning algorithms, such as neural networks that have an advantage in processing high-dimensional complex data and could potentially be more efficient in evaluating the effectiveness of individual cuts [24]. Nevertheless, they have certain limitations, such as larger dataset requirements, difficult implementation, longer training time, and poor transparency. Therefore, neural networks would not be suitable for the TEP problem, but their application could be explored in combination with other power system problems.

The proposed approach combines Benders' Decomposition with ML techniques to improve the computational efficiency and scalability of long-term multi-stage transmission problems under uncertainty. It can also potentially improve the problem's flexibility and robustness, leveraging the framework's capability to allow the user to solve problems with modified parameters without the need to re-sample and retrain ML models.

Existing dynamic TEP problems under uncertainty include stochastic and robust formulations depending on the description of the uncertainty and the computational challenges encountered in their solution. Robust formulations involve solving a series of nested optimization problems to ensure feasible solutions under all possible realizations within the uncertainty sets. Typically, the robust formulation involves min-max structures, making them more complex and challenging to solve than their stochastic counterparts [32], especially considering the large

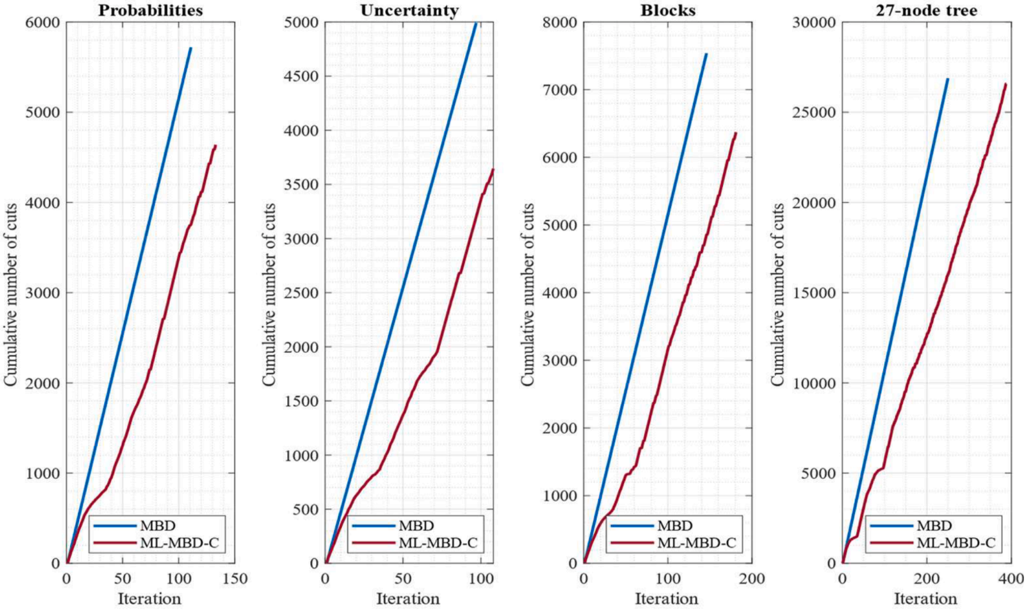


Fig. 11. Comparison between MBD and ML-MBD-C in terms of cumulative number of appended cuts over iterations for the four modified problems.

uncertainty affecting the long-term TEP. Moreover, robust formulations can be overly conservative, leading to higher costs. In [32], the authors present comprehensive computational results showing that robust formulations are more complex and require more computational resources than stochastic formulations. Scenario-based approaches have been adopted to solve robust formulations of long-term multi-stage transmission planning problems to reduce computational complexity and improve scalability compared to traditional robust optimization methods [35]. Hence, the proposed framework could also support the solution of robust formulations, allowing the efficient identification of representative scenarios and constraints rather than considering all possible uncertainties. The integration of ML allows for the adaptive handling of uncertainties and offers more robust solutions than purely heuristic or deterministic models. The proposed method is scalable to larger and more complex problems, which is a significant advantage over some traditional heuristic methods [31] that might struggle with large-scale TEP problems and can enhance recent state-of-the-art Dynamic Robust Transmission Expansion Planning formulations [30] leveraging Adaptive Robust Optimization frameworks to handle uncertainties.

The most used algorithmic approaches to solve large robust and stochastic optimization problems are i) Decomposition techniques, ii) Column-and-Constraint Generation (CCG), and iii) Metaheuristics. The CCG generates scenarios and constraints iteratively to refine the solution space. It balances solution quality and computational effort but can be slow for very large problems [33]. Instead, metaheuristic techniques like Genetic Algorithms, Simulated Annealing, and Particle Swarm Optimization can be used for larger problems [34] where exact methods are infeasible. They provide good solutions in reasonable time frames but may not guarantee optimality, and they do not return any information on the quality of the provided solution. Moreover, their performance depends heavily on parameter tuning and algorithm design.

This paper aims to push forward the computational limits of decomposition techniques in general. Decomposition techniques based on Benders decomposition are the most efficient for large-scale problems [36], as they can exploit the problem structure. The Nested Benders decomposition in [5] demonstrated good computational performance solving long-term multi-stage problems for the IEEE 118-bus system by trading off the solution optimality. Instead, the technique proposed in this paper demonstrated the ability to achieve optimality while preserving acceptable computational times. A limitation of the proposed method is that it works to filter ineffective cuts and, as such, convergence cannot be achieved in fewer iterations than with MBD. Methods aimed at accelerating convergence via advanced cut generation exist, for instance as in [29]. Applying the ideas developed in this work to such techniques could unlock further computational benefits.

6. Conclusion

Modern TEP frameworks must involve a high degree of modeling complexity that enables more informed decision-making to facilitate the transition to a net-zero power sector. However, state-of-the-art models are reaching computational limits, even with the application of advanced solution techniques. Motivated by the main computational bottleneck of MBD and the potential of learning-assisted optimization, this paper proposed an ML-enhanced Benders decomposition approach to solve large-scale stochastic TEP problems that manages the increase in MP size over iterations. The importance of the contribution has been corroborated by the attained results, which, most notably, demonstrated that ML-MBD is able to reach convergence when applied to a problem that is otherwise unsolvable with the well-established MBD approach, thus pushing the tractability limits of modern TEP problems. Significant benefits were observed in comparably smaller problems as well, particularly in reducing computational time and memory requirements. We also emphasized the advantage of the proposed method in preserving solution quality, in contrast to other approaches that alter the original problem structure.

The proposed framework is the first to use ML to classify or evaluate the quality of cuts in a multi-stage setting or when applied to power systems. Its generality can benefit researchers who are exploring the integration of ML with decomposition methods to solve large-scale optimization problems, while the generalizability properties highlight the value and applicability of ML-MBD in practice.

In future work, we aim to apply the proposed decomposition method to other power system problems and formulations while enhancing the convergence rate of ML-MBD through advanced cut generation techniques. We also plan to investigate further the scalability of the ML-MBD approach on larger and more complex expansion planning problems to validate its computational efficiency and robustness. In addition, we plan to integrate real-time data and adaptive learning techniques to boost the algorithm's dynamic decision-making capabilities. Using real-time operational data could improve the predictive accuracy of the ML models. Finally, we will work on refining the cut classification process and overall algorithm performance by leveraging advanced ML techniques, such as deep learning and reinforcement learning, while considering the trade-offs between computational complexity and the benefits of these advanced techniques. The envisaged future directions aim to enhance further the effectiveness and applicability of the proposed ML-enhanced Multi-Cut Benders Decomposition approach, making it a more versatile and powerful tool for solving large-scale optimization problems considering multidimensional uncertainty in power systems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data sources have been referenced in the manuscript.

Acknowledgements

The authors would like to acknowledge the support from the Engineering and Physical Sciences Research Council (EPSRC), Grant numbers EP/R045518/1 and EP/W034204/1.

Appendix: Nomenclature

Parameters	
CPI	Cut Performance Indicator
N_{BD}	Maximum number of iterations in the Benders algorithm
N_S	Number of sampling iterations
N_ζ	Number of classifiers ζ
ϵ_{BD}	Convergence tolerance of the Benders algorithm
ϵ_{PR}	Area under the Precision-Recall curve
ϵ_{ROC}	Area under the Receiver Operating Characteristic curve
Θ_i^{CPI}	Training dataset for the corresponding CPI, $i = 1, .., N_\zeta$
$\mathbb{P}^M$	Master problem
$\mathbb{P}_s^S$	Subproblem s
G	Bus-to-generator incidence matrix
I	Bus-to-line incidence matrix
Λ	Length of transmission line
b	Susceptance of transmission line
i,j	Start and end bus of transmission line
D	Electricity demand in MW
F^0, S^0	Initial transmission and storage capacity
$\hat{F}, \overset{\vee}{F}$	Maximum and minimum network capacity upgrade
$\hat{S}, \ \overset{\vee}{S}$	Maximum and minimum storage capacity upgrade
η^+, η^-	Charging and discharging efficiency of storage units
$\widehat{P}$	Maximum generator output
R^u, R^d	Ramp up and ramp down capability of generators in MW/h
$\overline{\kappa}, \ \widetilde{\kappa}$	Fixed and variable investment costs
γ^G, γ^S	Operation cost for generators and storage units
Γ	Value of Lost Load
τ	Duration of time period t in hours
τ^S	Discharge duration of energy storage units
ϵ	Planning stage
π	State probability
ω	Weighting factor of a representative block
r_{ϵ_m}	Cumulative discount factor in the planning stage to which node m belongs (ϵ_m)
$\mathrm{E_o}$	Construction time (delay) of investment option o expressed in number of stages
Decision variables and solutions	
X	Binary decision variable for network investments
F	Continuous decision variable for network capacity investments
Y	Continuous decision variable for storage investments
F^C, Y^C	State variables representing aggregate investments
$\widetilde{F}^C, \widetilde{Y}^C$	Auxiliary decision variables
α	Decision variables to approximate the objective value of the subproblems in the current iteration
λ^F, λ^Y	Lagrangian multipliers (dual variables)
p	Generator output
ρ^+, ρ^-	Storage charge and discharge
E	Storage state of charge
y^σ	Slack decision variable representing load curtailment
f	Power flow
δ	Bus voltage angle
x, y	Vectors containing all decision variables of $\mathbb{P}^M$ and of $\mathbb{P}_s^S$
$\overline{\mathrm{x}}, \overline{\mathrm{y}}$	Vectors containing the current candidate solutions of $\mathbb{P}^M$ and of $\mathbb{P}_s^S$
$\overline{\alpha}_s$	$\mathbb{P}_s^S$ objective approximation at the current $\mathbb{P}^M$ solution
$\overline{\lambda}_s$	Vector containing dual variables obtained from $\mathbb{P}_s^S$
$\overline{\psi}^M$	Current objective function value of $\mathbb{P}^M$
$\overline{\psi}_s^S$	Current objective function value of $\mathbb{P}_s^S$

(*continued on next page*)

(*continued*)

Sets and indices	
Ω_B	Set of all short-term representative blocks, indexed b
Ω_C	Set of all Benders cuts, indexed c
Ω_C^E	Set of all effective Benders cuts generated in iteration k
$\Omega_{F(c)}^{CPI}$	Set of values of c for all features corresponding to CPI
Ω_G	Set of all generation units, indexed g
Ω_H	Set of all candidate storage units, indexed h
Ω_L	Set of all transmission lines, indexed l
Ω_M	Set of all scenario tree nodes, indexed m
$\Omega_{M(w)}$	Set of all scenario tree nodes belonging to scenario w
Ω_N	Set of all network buses, indexed n
Ω_S	Set of all subproblems, indexed s
$\Omega_{A(m)}^{E_o}$	Set of all nodes that are ancestors of m, from stage 1 until stage $\epsilon_m - E_o$
Ω_T^b	Set of all time periods in block b, indexed t
Ω_W	Set of all scenarios, indexed w
Ω_θ^{CPI}	Set of all labeling thresholds for the corresponding CPI, indexed θ
Ω_ζ^{CPI}	Set of all classifiers of the corresponding CPI, indexed ζ

References

[1] CIGRE, “C1 technical brochure: Optimal power system planning under growing uncertainty,” 2020.

[2] I. Konstantelos, G. Strbac, Valuation of Flexible Transmission Investment Options Under Uncertainty, IEEE Transactions on Power Systems 30 (2) (2015), https://doi.org/10.1109/TPWRS.2014.2363364.

[3] L.A. Roald, D. Pozo, A. Papavasiliou, D.K. Molzahn, J. Kazempour, A. Conejo, Power systems optimization under uncertainty: A review of methods and applications, Electric Power Systems Research 214 (2023), https://doi.org/10.1016/j.epsr.2022.108725.

[4] A.J. Conejo, et al., Investment in Electricity Generation and Transmission: Decision Making Under Uncertainty, Springer International Publishing, Cham, Switzerland, 2016.

[5] P. Falugi, I. Konstantelos, G. Strbac, Planning With Multiple Transmission and Storage Investment Options Under Uncertainty: A Nested Decomposition Approach, IEEE Transactions on Power Systems 33 (4) (2018) 3559–3572, https://doi.org/10.1109/TPWRS.2017.2774367.

[6] A.K. Dixit, R. Pindyck, Investment Under Uncertainty, Princeton University Press, Princeton; Chichester, 1994.

[7] I. Konstantelos, S. Giannelos, G. Strbac, Strategic Valuation of Smart Grid Technology Options in Distribution Networks, IEEE Transactions on Power Systems 32 (2017), https://doi.org/10.1109/TPWRS.2016.2587999.

[8] S. Borozan, S. Giannelos, G. Strbac, Strategic Network Expansion Planning with Electric Vehicle Smart Charging Concepts as Investment Options, Advances in Applied Energy 5 (2022), https://doi.org/10.1016/j.adapen.2021.100077.

[9] A.J. Conejo, E. Castillo, R. Minguez, R. Garcia-Bertrand, Decomposition Techniques in Mathematical Programming: Engineering and Science Applications, Springer Science & Business Media, 2006.

[10] J. Benders, Partitioning methods for solving mixed variables programming problems, Numerische Mathematik 4 (1962).

[11] R.M. Van Slyke, R Wets, L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming, SIAM Journal on Applied Mathematics 17 (4) (1969).

[12] R. Rahmaniani, T.G. Crainic, M. Gendreau, W. Rei, The Benders decomposition algorithm: A literature review, European Journal of Operational Research 259 (2017), https://doi.org/10.1016/j.ejor.2016.12.005.

[13] J.R. Birge, F.V. Louveaux, A multicut algorithm for two-stage stochastic linear programs, Eur. J. Oper. Res. 34 (1988) 384–392.

[14] M.S. Ibrahim, W. Dong, Q. Yang, Machine learning driven smart electric power systems: Current trends and new perspectives, Applied Energy 272 (2020), https://doi.org/10.1016/j.apenergy.2020.115237.

[15] Y. Bengio, A. Lodi, A. Prouvost, Machine learning for combinatorial optimization: A methodological tour d’horizon, European Journal of Operational Research 290 (2021), https://doi.org/10.1016/j.ejor.2020.07.063.

[16] S. Pineda, J.M. Morales, Is learning for the unit commitment problem a low-hanging fruit? Electric Power Systems Research 207 (2022) https://doi.org/10.1016/j.epsr.2022.107851.

[17] G. Ruan, et al., Review of learning-assisted power system optimization, CSEE Journal of Power and Energy Systems 7 (2021), https://doi.org/10.17775/CSEEJPES.2020.03070.

[18] Y. Tao, J. Qiu, S. Lai, A Supervised-Learning Assisted Computation Method for Power System Planning, IEEE Transactions on Artificial Intelligence 3 (5) (2022), https://doi.org/10.1109/TAI.2021.3133821.

[19] C. Li, Z. Dong, G. Chen, B. Zhou, J. Zhang, X. Yu, Data-Driven Planning of Electric Vehicle Charging Infrastructure: A Case Study of Sydney, Australia, IEEE Transactions on Smart Grid 12 (4) (2021), https://doi.org/10.1109/TSG.2021.3054763.

[20] S. Misra, L. Roald, Y. Ng, Learning for Constrained Optimization: Identifying Optimal Active Constraint Sets, INFORMS Journal on Computing 34 (2021), https://doi.org/10.1287/ijoc.2020.1037.

[21] E. Prat, S. Chatzivasileiadis, Learning Active Constraints to Efficiently Solve Linear Bilevel Problems: Application to the Generator Strategic Bidding Problem, IEEE Transactions on Power Systems 38 (3) (2022), https://doi.org/10.1109/TPWRS.2022.3188432.

[22] A.S. Xavier, F. Qiu, S. Ahmed, Learning to Solve Large-Scale Security-Constrained Unit Commitment Problems, INFORMS Journal on Computing 33 (2021), https://doi.org/10.1287/ijoc.2020.1037.

[23] Y. Tang, S. Agrawal, Y. Faenza, Reinforcement Learning for Integer Programming: Learning to Cut, in: Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 9367–9376.

[24] H. Jia, S. Shen, Benders Cut Classification via Support Vector Machines for Solving Two-Stage Stochastic Programs, INFORMS Journal on Optimization 3 (3) (2021), https://doi.org/10.1287/ijoo.2019.0050.

[25] M. Lee, N. Ma, G. Yu, H. Dai, Accelerating Generalized Benders Decomposition for Wireless Resource Allocation, IEEE Transactions on Wireless Communications 20 (2) (2021), https://doi.org/10.1109/TWC.2020.3031920.

[26] A.C. Müller, S. Guido, Introduction to machine learning with Python: a guide for data scientists, 1st ed., O’Reilly Media, Sebastopol, CA, 2017.

[27] A. Ruszczyński, Decomposition Methods, Handbooks in Operations Research and Management Science 10 (2003) 141–211, https://doi.org/10.1016/S0927-0507(03)10003-5.

[28] S. Peyghami, P. Davari, M. Fotuhi-Firuzabad, F. Blaabjerg, Standard Test Systems for Modern Power System Analysis: An Overview, IEEE Industrial Electronics Magazine 13 (4) (2019), https://doi.org/10.1109/MIE.2019.2942376.

[29] R. Brandenberg, P. Stursberg, Refined cut selection for benders decomposition: applied to network capacity expansion problems, Math Meth Oper Res 94 (2021), https://doi.org/10.1007/s00186-021-00756-8.

[30] R. García-Bertrand, R. Mínguez, Dynamic Robust Transmission Expansion Planning, IEEE Transactions on Power Systems 32 (4) (July 2017) 2618–2628, https://doi.org/10.1109/TPWRS.2016.2629266.

[31] Gholam-Reza Kamyab, Mahmood Fotuhi-Firuzabad, Masoud Rashidinejad, A PSO based approach for multi-stage transmission expansion planning in electricity markets, International Journal of Electrical Power & Energy Systems 54 (2014) 91–100. ISSN 0142-0615.

[32] C. Ruiz, A.J. Conejo, Robust Transmission Expansion Planning, European Journal of Operational Research”, 242 (2) (2015) 390–401, https://doi.org/10.1016/j.ejor.2014.10.011, 2015.

[33] S. Sen, J.L. Higle, Decomposition Algorithms for Large-Scale Two-Stage Stochastic Mixed-Integer Programming: A Computational Study, Mathematical Programming 104 (1) (2005) 1–20, https://doi.org/10.1007/s10107-005-0583-0.

[34] M. R. Alvarez, C. Rahmann, R. Palma-Behnke and P. A. Estévez, "A novel meta-heuristic model for the multi-year transmission network expansion planning", International Journal of Electrical Power & Energy Systems, vol. 107, 523-537, 10.1016/j.ijepes.2018.12.022.

[35] S. Jin, S.M. Ryan, Scenario-Based Multi-Stage Robust Optimization for Transmission Expansion Planning, IEEE Transactions on Power Systems 26 (4) (2001) 2271–2280, https://doi.org/10.1109/TPWRS.2011.2145384.

[36] E. Rahmanian, S.J. Wright, Benders Decomposition for Large-Scale Mixed-Integer Linear Programming, INFORMS Journal on Computing 30 (3) (2018) 432–447, https://doi.org/10.1287/ijoc.2017.0785.